ETHIRIS®

VIDEO MANAGEMENT SOFTWARE

Admin - Configuration for Ethiris

KENTIMA

Automation and Security Products

Production and layout: Kentima AB
Version: 11.2
First edition: March 2020
Printing: Kentima AB
Trademarks: Ethiris and **WideQuick** are registered trademarks. All other marks belong to their respective owners.

Kentima AB

Postal address:
PO Box 174
SE-245 22 STAFFANSTORP
Sweden

Street address:
Kastanjevägen 4
SE-245 44 STAFFANSTORP
Sweden

Email address:
info@kentima.com

Internet:
www.kentima.com

Contents

KENTIMA
Automation and Security Products

1 Introduction 1:1

1 Introduction

1.1 Introduction

This manual is designed to give users a detailed description of Ethiris Admin.

There are a total of 6 various manuals for Ethiris. Other than this one, there is also *Installing Ethiris*, *Getting Started with Ethiris*, *Ethiris Client User´s Guide, Integration with Ethiris* & *Getting started with Ethiris Mobile*

This part of the manual suite describes the details of how to configure an Ethiris system using *Ethiris Admin*, assuming that Ethiris is already installed on the computer.

For in-depth information on Ethiris Client, please see the *Ethiris Client User´s Guide* manual.

For information on how to get started with Ethiris, please see the *Getting Started with Ethiris* manual.

Note that depending on the current configuration and the license level in your system, some of the settings described in this manual may not be accessible.

1.1.1 Use

The primary purpose of Ethiris is camera surveillance, which is performed in two ways. One way is manually to monitor live video from different cameras. The other way involves recording video from connected cameras. Recording can take place continuously from one or more cameras or in the form of video sequences when a recording condition is met. The recorded video can be played back afterward using sophisticated timelines and a VCR-like interface.

1.2 General Description

Ethiris is a surveillance system that uses network cameras and analog cameras, together with video encoders from different suppliers.

The product is divided into several program parts, where *Ethiris Server* and *Ethiris Client* are the most important ones. The server part manages all cameras and stores video on the hard disk. The client part displays live video and recorded video sequences.

There is also a program called *Ethiris Admin*, which is used for configuring all Ethiris modules in the system. In *Ethiris Admin,* you configure the *Ethiris Servers* by, e.g., defining which cameras are connected to each *Ethiris Server* when video shall be recorded, what frame rate and resolution to use, etc. You also define the view layout in the various *Ethiris Clients* in the system using *Ethiris Admin*.

In theory, an unlimited number of cameras could be connected to each Ethiris server and be displayed in the desired number of Ethiris clients simultaneously. In practice, however, bandwidth and monitor resolution set limits for the appropriate number of cameras connected.

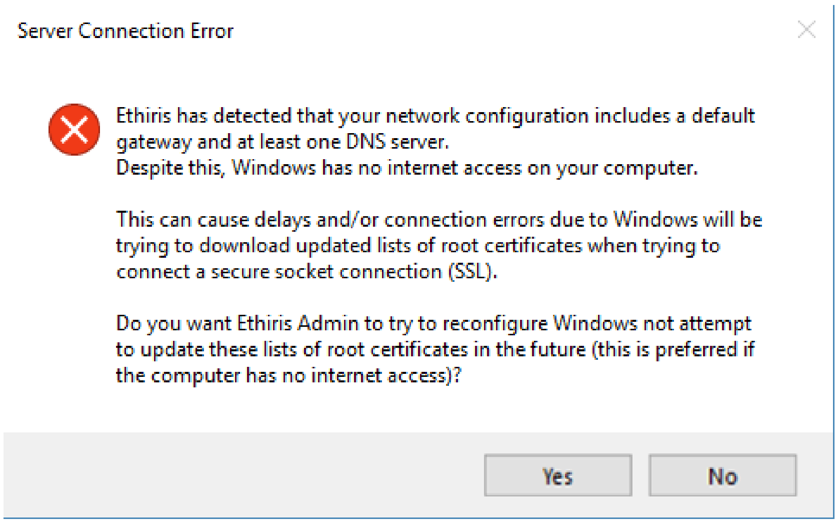
There are different license levels for Ethiris, which permit different numbers of cameras to be connected. To meet the need for a large number of cameras, Ethiris is designed with a focus on scalability. Scalability means that it is possible to divide your system into several Ethiris servers and thus distribute the load over several computers.

1.3 Startup/Connection delays

An issue when connecting to *Ethiris Server* has been noted. The issue shows itself in two ways. When starting *Ethiris Admin*, there would be a long startup delay, around 10 seconds. And when *Ethiris Admin* was started, and the server(s) where loaded in the treeview, this would fail on the first try. And subsequent attempts would succeed.

The issue occurs most often due to the system being configured with both a default gateway and one or more DNS servers in the network settings, while there is no Internet connection available. Hence Windows times out when trying to retrieve updated SSL certificate information.

From Ethiris 11.2, there is added detection and correction procedure for this issue. Upon detection of the issue, the following popup is displayed on the startup of *Ethiris Admin*.


Server Connection Error

Ethiris has detected that your network configuration includes a default gateway and at least one DNS server.
Despite this, Windows has no internet access on your computer.

This can cause delays and/or connection errors due to Windows will be trying to download updated lists of root certificates when trying to connect a secure socket connection (SSL).

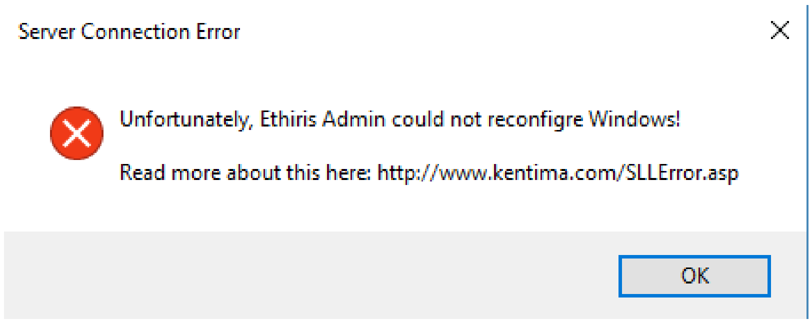
Do you want Ethiris Admin to try to reconfigure Windows not attempt to update these lists of root certificates in the future (this is preferred if the computer has no internet access)?

Figure 1.1 Ethiris Admin detected a connection issue.

Pressing Yes, an attempt to fix the issue begins, pressing No, simply closes the dialog.

After pressing Yes, a UAC prompt, User Account Control, from Windows might popup, depending on system configuration. Pressing No, the attempt to correct the issue is halted.

And the following dialog is displayed.

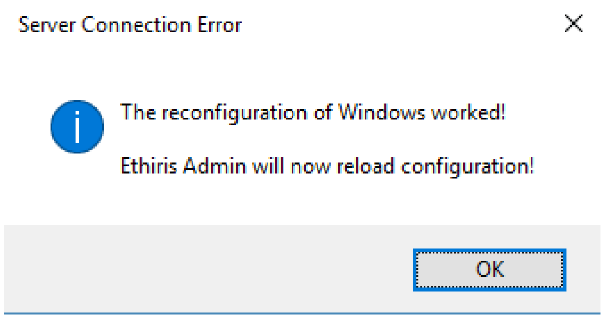

Server Connection Error

Unfortunately, Ethiris Admin could not reconfigre Windows!

Read more about this here: http://www.kentima.com/SLLError.asp

Figure 1.2 Ethiris Admin failed to correct the issue.

Pressing Yes on the UAC prompt, and if needed, inputting credential for higher privileges, the correction is attempted, and if successful, the following dialog is shown.


Server Connection Error

The reconfiguration of Windows worked!

Ethiris Admin will now reload configuration!

Figure 1.3 Ethiris Admin succeeded in correcting the issue.

This correction can also be manually applied to the server and clients affected by this delay.

Read more here: http://www.kentima.com/SSLError.asp

1.4 Ethiris Components

Ethiris Server	Ethiris Server is the core of an Ethiris system. It handles all communication with the cameras and recording of video on a hard disk. Ethiris Server runs as a service under the operating system and is automatically started as soon as the computer is started. An Ethiris system can be comprised of one or several Ethiris Servers.
Ethiris Client	Ethiris Client is used for viewing live video and recorded video. An Ethiris Client can connect to one or several Ethiris Servers for access to cameras. An Ethiris system can contain one or several Ethiris Clients.
Ethiris Admin	Ethiris Admin is used for configuring the different parts in an Ethiris system. Ethiris Server and Ethiris Client are configured with this common tool. From any computer in the system, Ethiris Admin can be run and used for configuring all Ethiris components on-site.
Ethiris Mobile	*Ethiris Mobile* is an app for connecting to your Ethiris systems via your cell. With Ethiris Mobile you can watch live video from various cameras in your system. You can also see all the alarms.
Ethiris Server OPC Server	Ethiris Server OPC Server is a separate Ethiris component that is used for letting other systems gain access to the information in Ethiris Server. Any other system with an OPC client can connect to one or several Ethiris Servers and read/write to all signals in Ethiris Server. E g starting recording of a camera or control a PTZ camera.
Ethiris ActiveX	Ethiris ActiveX is a component used for viewing live video from a camera connected to an Ethiris Server. This component can be used in any system that can handle standard ActiveX components.
Ethiris Viewer	Ethiris Viewer is a separate program used for viewing exported video from an Ethiris system.
Ethiris Signature Validator	*Ethiris Signature Validator* is a separate software that is used to validate the authenticity of exported jpg images and video sequences, so-called *watermark*.

2 Ethiris Admin

2.1 Admin windows

2.1.1 Overview

In this section, we take a closer look at the user interface in Ethiris Admin and its various windows, toolbars, menus, and panels.

The *main window* is made up of several smaller parts. Some of these, like the main menu and toolbar, are fixed in size and position, while others, like the various *panels*, can be moved around, docked, resized, and even closed.

There are two types of windows in Ethiris Admin; *Document windows* and *Tool windows*.

Document windows (a.k.a. *panels*) are used for presenting various types of information such as properties for a camera. In Ethiris Admin, no document windows are open by default. Document windows are opened when double-clicking the corresponding node in the *Ethiris Explorer* treeview.

Tool windows are used to supply the user with various tools, such as a treeview with the current project. In Ethiris Admin, there is only one tool window; *Ethiris Explorer*. Ethiris Explorer is a treeview that presents the currently open project with all its components, such as Ethiris Servers and Ethiris Clients.

If you close all windows that can be closed in Ethiris Admin, there is a menu bar and toolbar at the top and a stripe at the bottom containing the Kentima logotype. The remaining space is left for the various windows. See *Figure 2.1*.

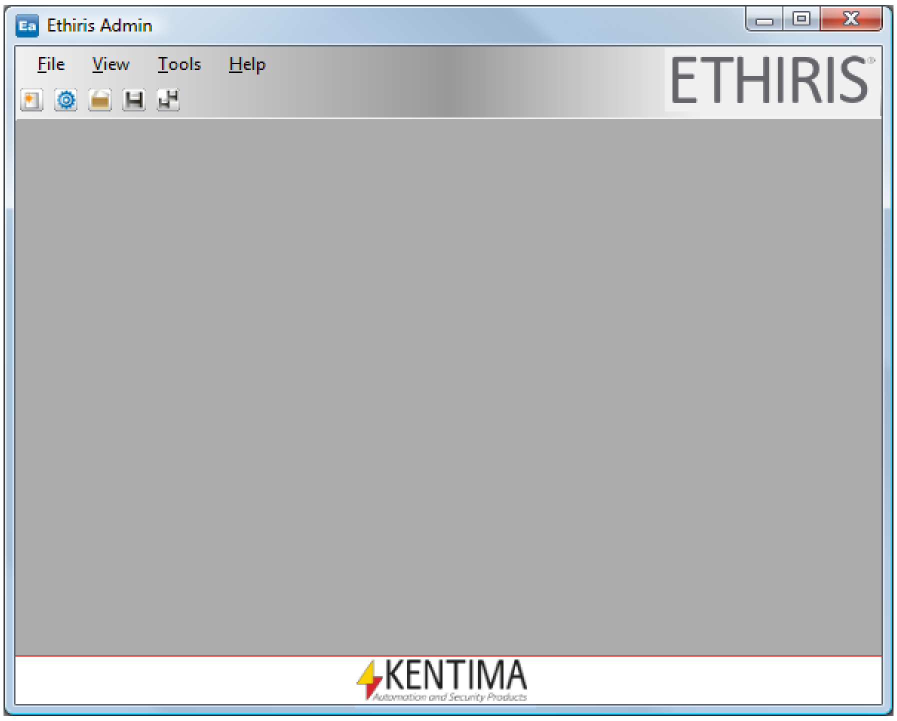


Figure 2.1 The Ethiris Admin with all windows closed.

By default, when you start Ethiris Admin, the *Ethiris Explorer* tool window is docked to the left side of the main frame. See *Figure 2.2*.

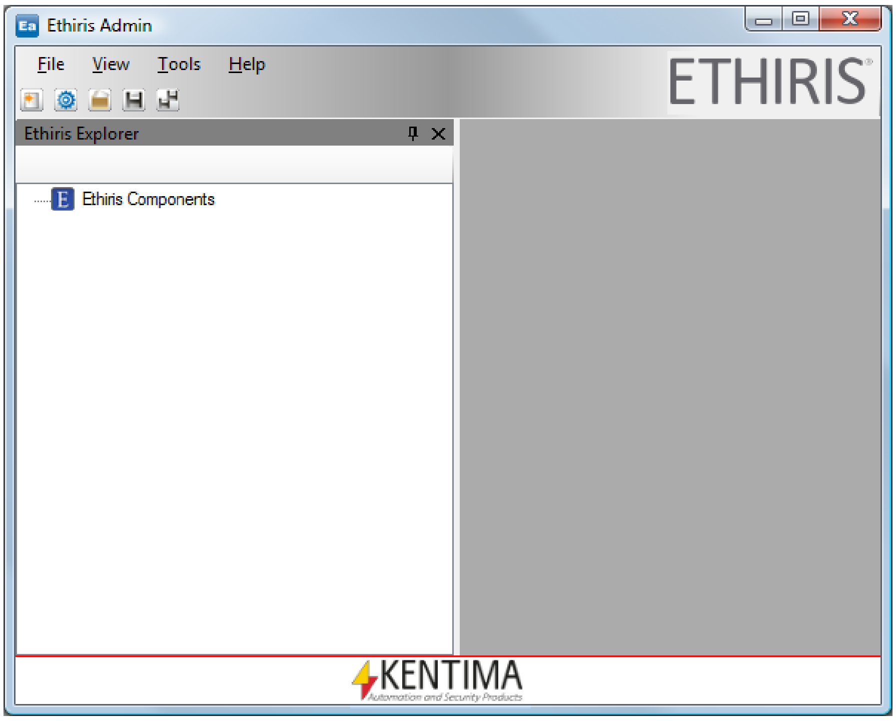


Figure 2.2 The Ethiris Admin with just Ethiris Explorer tool window open.

The *Ethiris Explorer* tool window, by default, is docked to the left edge and is pinned. When a window is docked to the side of the main frame, it automatically becomes a tool window. A tool window has two states, pinned and unpinned.

When a window is pinned, the pin icon is in a vertical position, see *Figure 2.3*. In this state, the window is visible all the time. The width of the window can be changed by moving the right edge of the tool window.

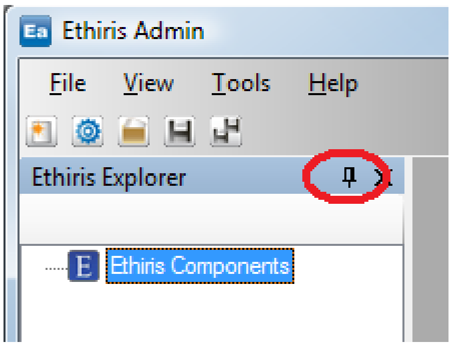


Figure 2.3 A pinned tool window.

To unpin the window, click the vertical pin icon. When a tool window is unpinned and loses focus, it automatically slides in against the edge it is docked to, leaving visible a tag displaying the title of the tool window. See *Figure 2.4*.

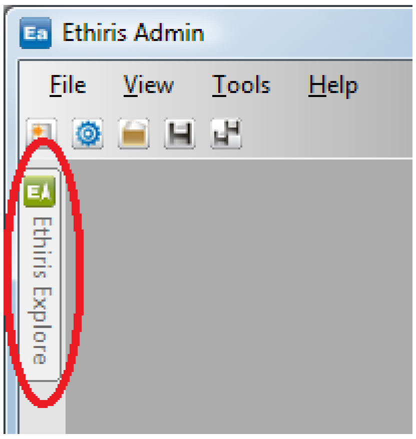


Figure 2.4 An unpinned tool window visible only as a tab.

To make the tool window visible again, move the mouse pointer over the tab, and the window slides out. In this state, the tool window has not yet received focus; see *Figure 2.5*. Meaning that if you move the mouse pointer outside the window, it slides back to the edge again.

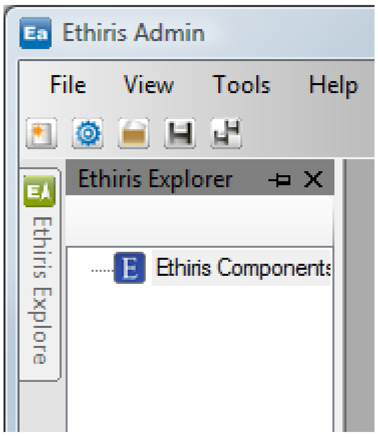


Figure 2.5 An unpinned tool window made visible again, but without focus.

To make the window stay out, click the window title area. Notice in *Figure 2.5* that the title area is light blue. This means that it has not got focus. A tool window with focus has a blue color, see *Figure 2.6*. As long as a tool window has focus, it doesn't slide back to the edge.

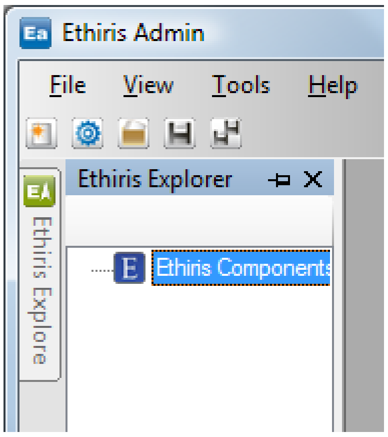


Figure 2.6 An unpinned tool window with focus (blue title bar).

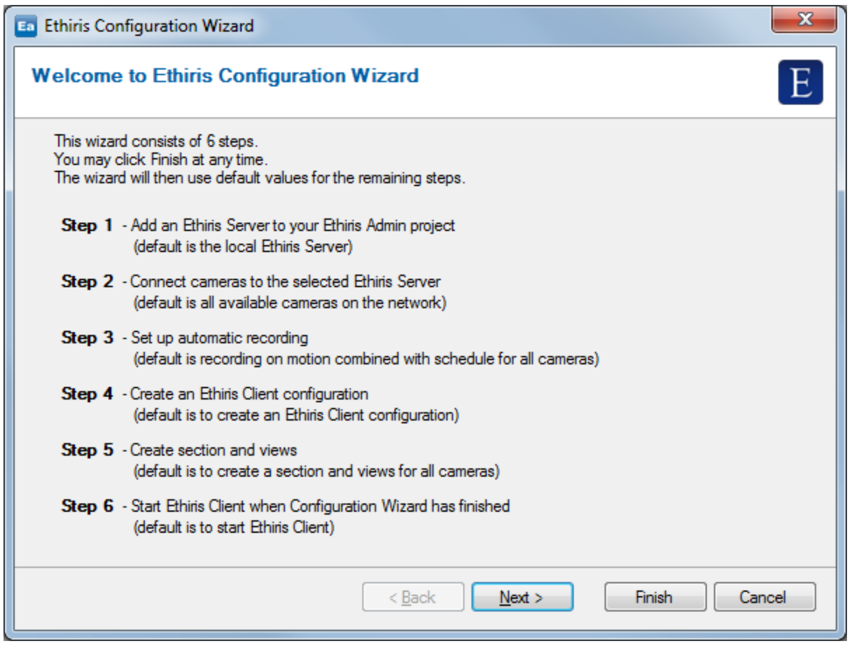
To pin the window again, click the horizontal pin icon.

The whole idea with unpinning the tool windows is that the remaining space in the main frame is increased.

Ethiris configuration wizard

Every time you start Ethiris Admin with an empty configuration, the Ethiris configuration wizard is automatically started.

The purpose is to help the user get started with her Ethiris system.


Ethiris Configuration Wizard

Welcome to Ethiris Configuration Wizard

This wizard consists of 6 steps.
You may click Finish at any time.
The wizard will then use default values for the remaining steps.

Step 1 - Add an Ethiris Server to your Ethiris Admin project
(default is the local Ethiris Server)

Step 2 - Connect cameras to the selected Ethiris Server
(default is all available cameras on the network)

Step 3 - Set up automatic recording
(default is recording on motion combined with schedule for all cameras)

Step 4 - Create an Ethiris Client configuration
(default is to create an Ethiris Client configuration)

Step 5 - Create section and views
(default is to create a section and views for all cameras)

Step 6 - Start Ethiris Client when Configuration Wizard has finished
(default is to start Ethiris Client)

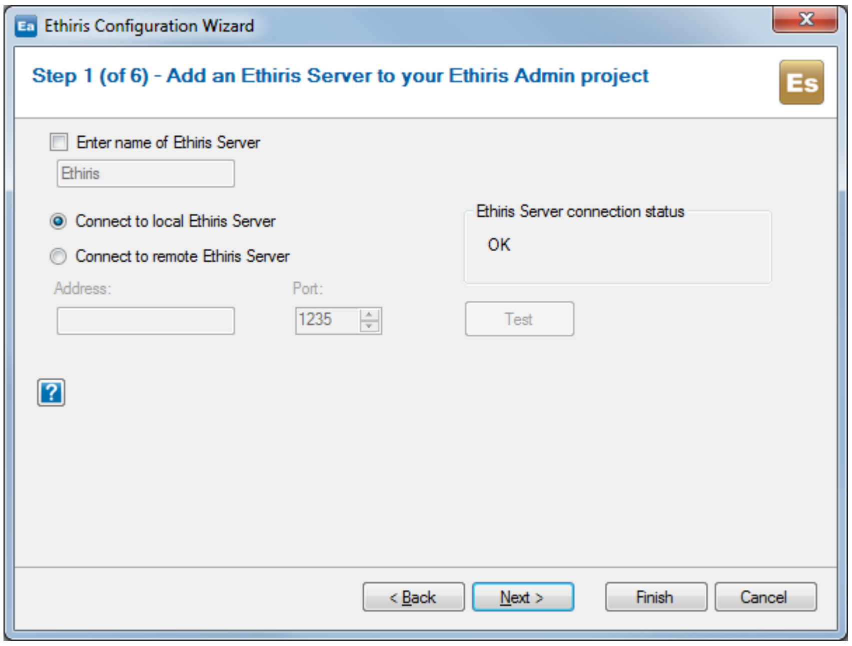
Figure 2.7 Ethiris configuration wizard.

The wizard consists of 6 steps, but you can, at any time, click *Finish*. Then the remaining steps in the wizard are performed using default values.

In the welcome dialog, the 6 steps and their corresponding default values are presented.

In *Step 1,* you are supposed to select an Ethiris Server that should be added to the Ethiris Admin project. As a default, the local Ethiris Server is used, i.e., the

Ethiris Server, which is installed on the computer where you just started Ethiris Admin.


Ethiris Configuration Wizard

Step 1 (of 6) - Add an Ethiris Server to your Ethiris Admin project

☐ Enter name of Ethiris Server: Ethiris

● Connect to local Ethiris Server

○ Connect to remote Ethiris Server

Address:

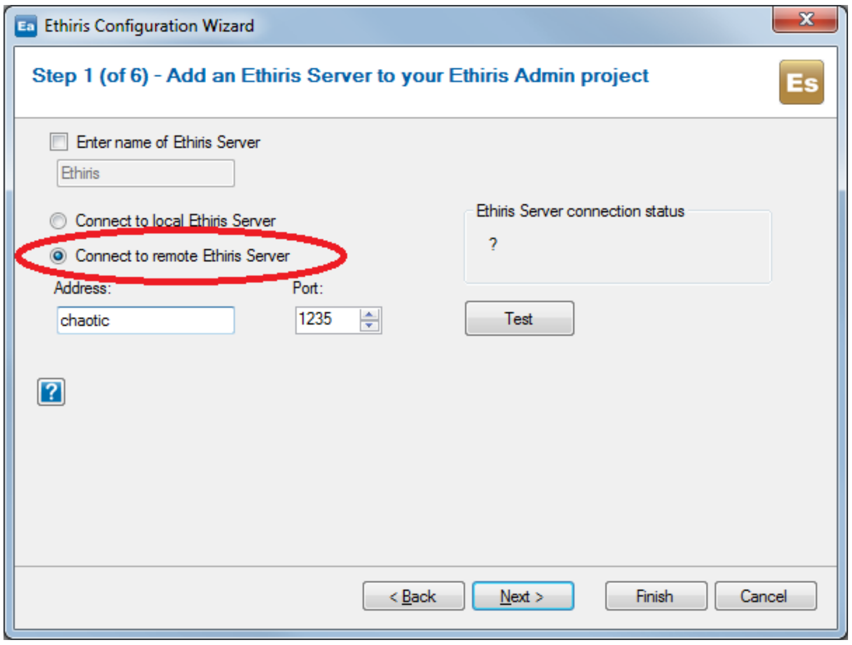
Port: 1235

Ethiris Server connection status: OK

Figure 2.8 Step 1 in the configuration wizard.

The first time you enter this step, the wizard automatically tries to connect to the local Ethiris Server. If this works out, it displays *OK* in the box *Ethiris Server connection status*.

If it doesn't work out or if you, for some other reason, want to connect to an Ethiris Server running on another computer, you have to choose the alternative *Connect to remote Ethiris Server*.


Ethiris Configuration Wizard

Step 1 (of 6) - Add an Ethiris Server to your Ethiris Admin project

☐ Enter name of Ethiris Server: Ethiris

○ Connect to local Ethiris Server

● Connect to remote Ethiris Server

Address: chaotic

Port: 1235

Ethiris Server connection status: ?

Figure 2.9 Connection to another computer.

Enter the desired IP address or computer name in the field *Address*. *Port* is as default *1235*, and there is seldom any reason for changing that. Click the button *Test* to test the connection to the selected Ethiris Server.

You cannot move on with the wizard until a successful connection to an Ethiris Server has been established.

As a default, the name *Ethiris* is used as the name of the Ethiris Server in the treeview of Ethiris Admin. If you want to change the name, check the checkbox *Enter name of Ethiris Server*.

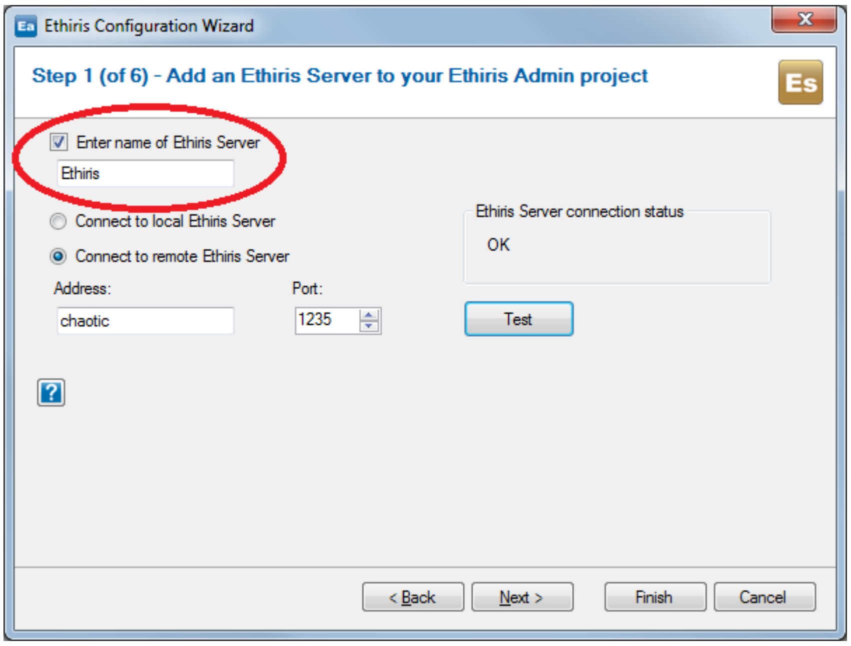
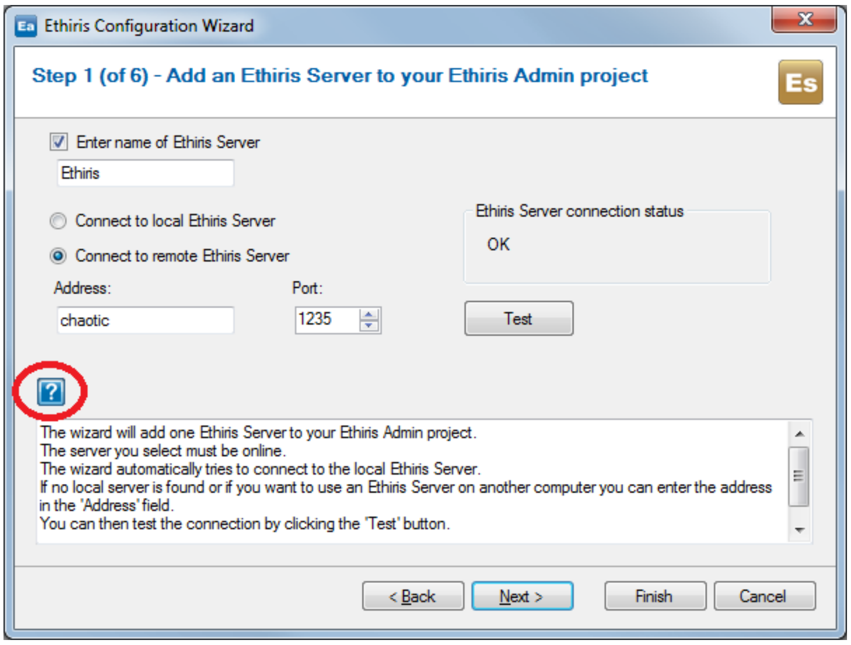

Ethiris Configuration Wizard
Step 1 (of 6) - Add an Ethiris Server to your Ethiris Admin project
Es
Enter name of Ethiris Server
Ethiris
Connect to local Ethiris Server
Connect to remote Ethiris Server
Address:
chaotic
Port:
1235
Ethiris Server connection status
OK
Test
< Back
Next >
Finish
Cancel

Figure 2.10 If you want, you can change the default name of the Ethiris Server.

To get more information about the current step, click the question mark. An information box is displayed at the bottom of the wizard. Click again to hide the information box.


Ethiris Configuration Wizard
Step 1 (of 6) - Add an Ethiris Server to your Ethiris Admin project
Es
Enter name of Ethiris Server
Ethiris
Connect to local Ethiris Server
Connect to remote Ethiris Server
Address:
chaotic
Port:
1235
Ethiris Server connection status
OK
Test
< Back
Next >
Finish
Cancel

The wizard will add one Ethiris Server to your Ethiris Admin project.
The server you select must be online.
The wizard automatically tries to connect to the local Ethiris Server.
If no local server is found or if you want to use an Ethiris Server on another computer you can enter the address in the 'Address' field.
You can then test the connection by clicking the 'Test' button.

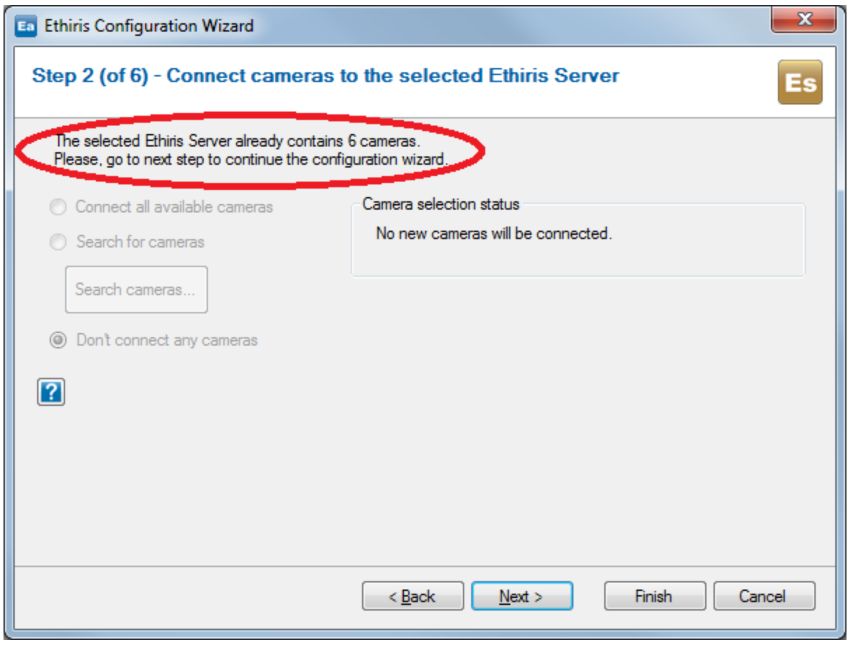
Figure 2.11 Show/hide extra information by clicking the question mark.

When you are satisfied with your selections in step 1, click *Next* to move on to step 2.

Remember that you at any time can click *Finish* upon which you will go directly to a summary where your selections so far will be listed together with the default values of the remaining steps in the wizard; this gives you an extra opportunity to change your selections.

When you enter step 2, an automatic camera search is usually started. If the Ethiris Server you have selected in step 1 already has cameras in its configuration, the wizard will not search for additional cameras.

Then it might look like this:

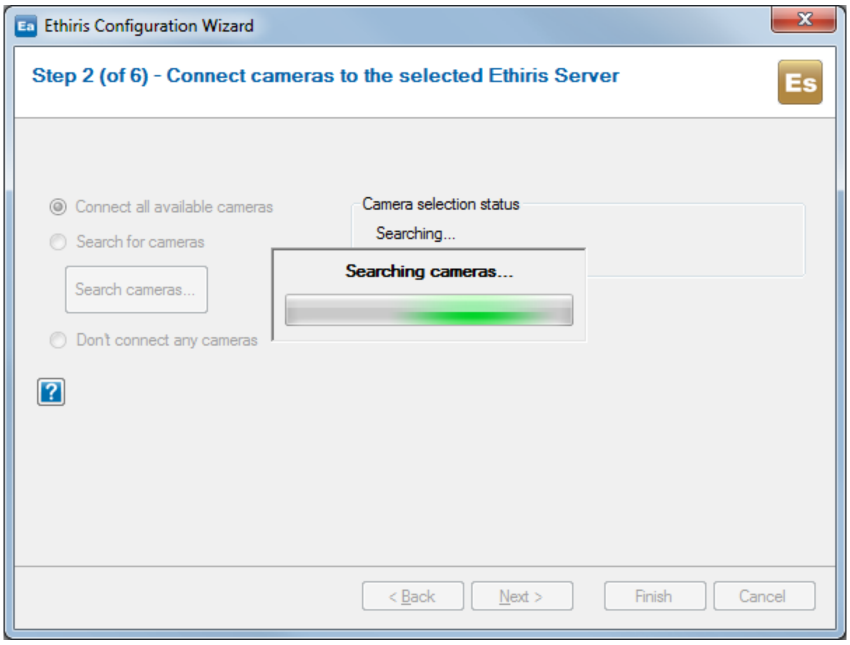

Ethiris Configuration Wizard
Step 2 (of 6) - Connect cameras to the selected Ethiris Server
Connect all available cameras
Search for cameras
Search cameras...
Don't connect any cameras
Camera selection status
< Back
Next >
Finish
Cancel

The selected Ethiris Server already contains 6 cameras.
Please, go to next step to continue the configuration wizard.

No new cameras will be connected.

Figure 2.12 Cameras already exists in the selected Ethiris Server.

But, it usually looks like this:

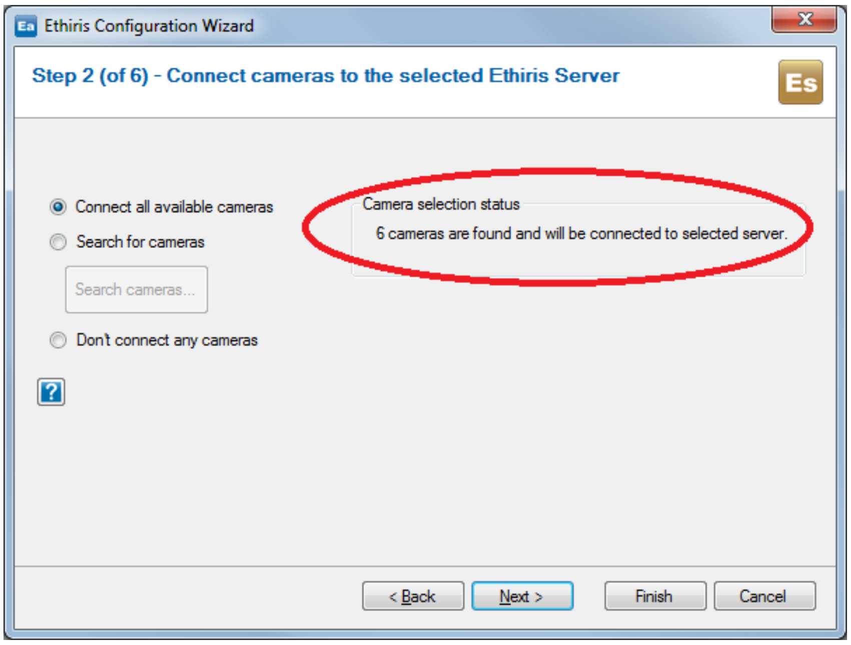

Ethiris Configuration Wizard
Step 2 (of 6) - Connect cameras to the selected Ethiris Server
Connect all available cameras
Search for cameras
Search cameras...
Don't connect any cameras
Camera selection status
< Back
Next >
Finish
Cancel

Searching...

Searching cameras...

Figure 2.13 Automatic search for available cameras.

In our example, the wizard found 6 cameras.



Ethiris Configuration Wizard

Step 2 (of 6) - Connect cameras to the selected Ethiris Server

- [✓ Connect all available cameras]
- Search for cameras
- Don't connect any cameras

Search cameras...

Camera selection status

6 cameras are found and will be connected to selected server.

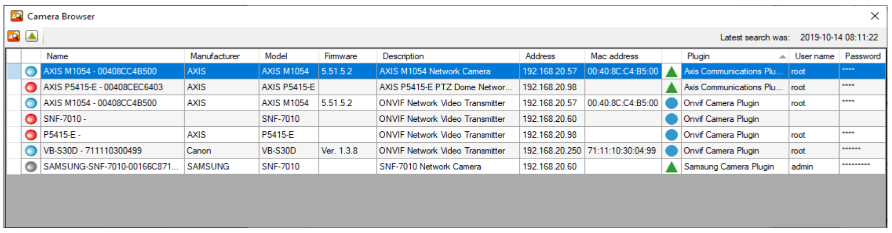
< Back | Next > | Finish | Cancel

Figure 2.14 The wizard found 6 cameras.

The wizard uses both ONVIF and UPnP (Plug-and-Play) in searching for cameras. If the same camera is found by both methods, the wizard will select the UPnP variant since the ONVIF support for a particular camera is less than the corresponding support for the UPnP variant.

There could be cameras on the network that are not found by the wizard. These cameras can be connected manually later on directly in Ethiris Admin via the panel *Cameras*.

If you are not content with the automatic search, you can instead select the alternative *Search for cameras* and click the button *Search cameras…*. Then the same dialog is displayed as when you search for cameras in Ethiris Admin via the panel *Cameras.*



Camera Browser

Latest search was: 2019-10-14 08:11:22

	Name	Manufacturer	Model	Firmware	Description	Address	Mac address		Plugin	User name	Password
●	AXIS M1054 - 00408CC4B500	AXIS	AXIS M1054	5.51.5.2	AXIS M1054 Network Camera	192.168.20.57	00:40:8C:C4:B5:00	▲	Axis Communications Plu...	root	****
●	AXIS P5415-E - 00408CEC6403	AXIS	AXIS P5415-E		AXIS P5415-E PTZ Dome Networ...	192.168.20.98		▲	Axis Communications Plu...	root	****
●	AXIS M1054 - 00408CC4B500	AXIS	AXIS M1054	5.51.5.2	ONVIF Network Video Transmitter	192.168.20.57	00:40:8C:C4:B5:00	●	Onvif Camera Plugin	root	****
●	SNF-7010 -		SNF-7010		ONVIF Network Video Transmitter	192.168.20.60		●	Onvif Camera Plugin		
●	P5415-E -	AXIS	P5415-E		ONVIF Network Video Transmitter	192.168.20.98		●	Onvif Camera Plugin	root	****
●	VB-S30D - 711110300499	Canon	VB-S30D	Ver. 1.3.8	ONVIF Network Video Transmitter	192.168.20.250	71:11:10:30:04:99	●	Onvif Camera Plugin	root	******
●	SAMSUNG-SNF-7010-00166C871...	SAMSUNG	SNF-7010		SNF-7010 Network Camera	192.168.20.60		▲	Samsung Camera Plugin	admin	*********

Figure 2.15 Camera browser where you select desired cameras.

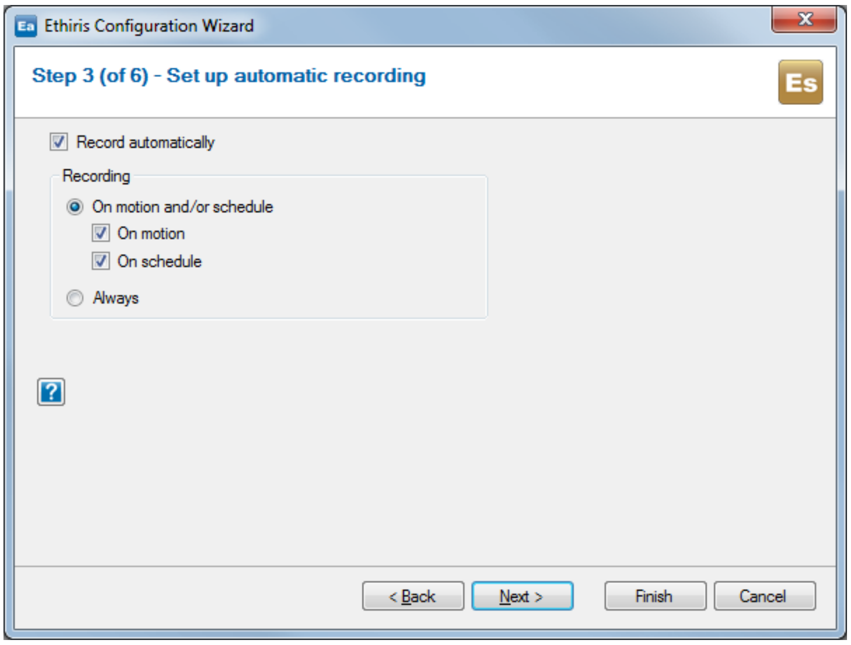
In this dialog, you can select which cameras to connect to the Ethiris Server. Select the desired camera, press the *Ctrl-key* for adding/removing cameras, or the *Shift-key* to choose an interval of cameras.

The third alternative in step 2 is to opt for not connecting any cameras at this time.

Remember, you can always connect cameras at a later time directly in Ethiris Admin via the panel *Cameras*.

When you are content with your camera selections, click *Next* to move on to the next step. If no cameras are added in step 2, the next step is step 4.

Step 3 is about making selections for automatic recording of the cameras selected in step 2. Note that if no cameras were selected in step 2, step 3 would be skipped by the wizard.


Ethiris Configuration Wizard

Step 3 (of 6) - Set up automatic recording

[✓] Record automatically

Recording

● On motion and/or schedule

[✓] On motion

[✓] On schedule

○ Always

Figure 2.16 Make selections for automatic recording.

For starters, you can choose whether the recording will be set up or not by the checkbox *Record automatically*.

The next selection is about whether you want to record always or if the recording shall be triggered by motion detection and/or a schedule.

If you select, *Always,* it results in a *continuous recording* that will go on all the time. This will be indicated by blue lines in the timelines in the video player in Ethiris Client.

If you select the first alternative, *On motion and/or schedule*, you can make further selections. You can choose if the recording will be started due to a detected motion, a schedule, or both.

It will be an *event recording* if *On motion* is selected. Otherwise, it will be a *continuous recording*. Event recordings are indicated by red lines in the Ethiris Client timelines.

The selections made in step 3 will affect the settings in the panel *Camera recording* in Ethiris Admin. In this panel, you can change the settings later.

The motion detectors and schedules that are created by the wizard have default settings, and these can be changed later directly in the corresponding panels in Ethiris Admin.

Click *Next* to move on to step 4.

Now we leave the settings for Ethiris Server and concentrate on the client.

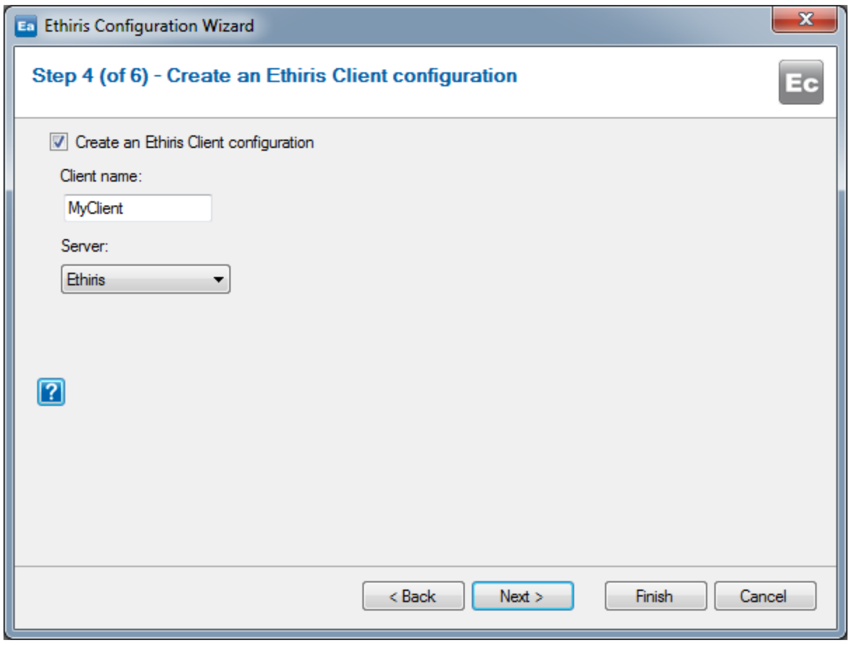

Ethiris Configuration Wizard
Step 4 (of 6) - Create an Ethiris Client configuration
Ec
Create an Ethiris Client configuration
Client name:
MyClient
Server:
Ethiris
< Back
Next >
Finish
Cancel

Figure 2.17 Select if a client configuration will be created.

In step 4, we will, first of all, select if we want to create a client configuration at all. Usually, you want to do that, and then you keep the *Create an Ethiris Client configuration* checkbox checked.

In this step, you decide which name to give the client configuration. This name will be used in the Ethiris Admin treeview. You need to select a configuration server. Possible selections are shown in the servers list.

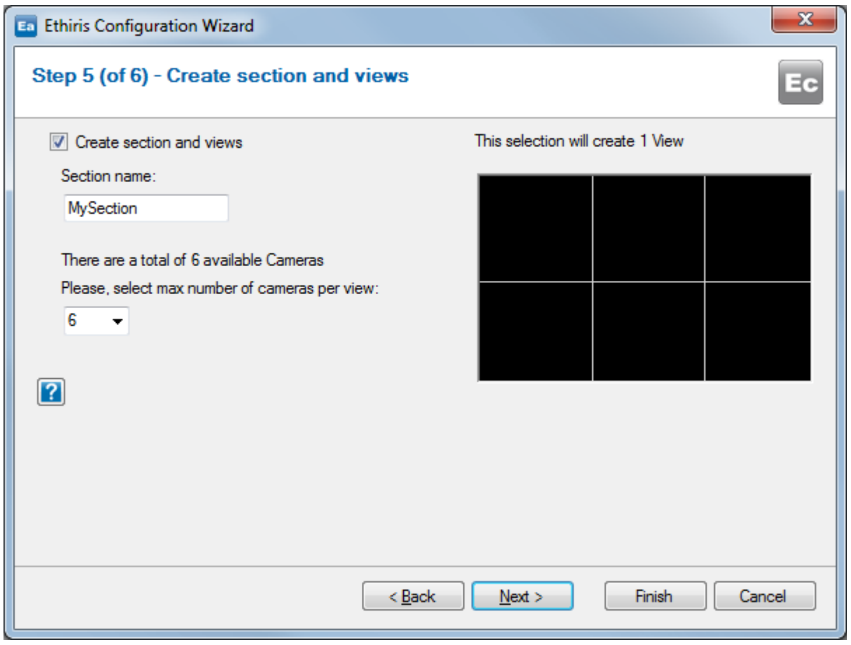
If there is already a configuration with the same name in the configuration server, you will be asked if you want to overwrite that configuration. If you don't want to do that, you have to enter another name for your configuration.

The wizard can create one client configuration. If you want several different client configurations, you can create those later, either manually in Ethiris Admin or by rerunning the configuration wizard by choosing the menu *File->New Ethiris Component->Ethiris Client using wizard...* in Ethiris Admin.

If you choose to create a client configuration, this step will create the client configuration, add it to the Ethiris Admin project and connect it to the Ethiris Server that was selected in step 2, rendering all the servers' cameras available in the client.

Click *Next* for moving on to the next step.

If you have opted for creating a client configuration, you will go to step 5. Otherwise, you will end up in the summary of the wizard.


Ethiris Configuration Wizard
Step 5 (of 6) - Create section and views
< Back
Next >
Finish
Cancel

Create section and views [✓]

This selection will create 1 View

Section name: MySection

There are a total of 6 available Cameras

Please, select max number of cameras per view: 6

Figure 2.18 Select if a section and views will be created.

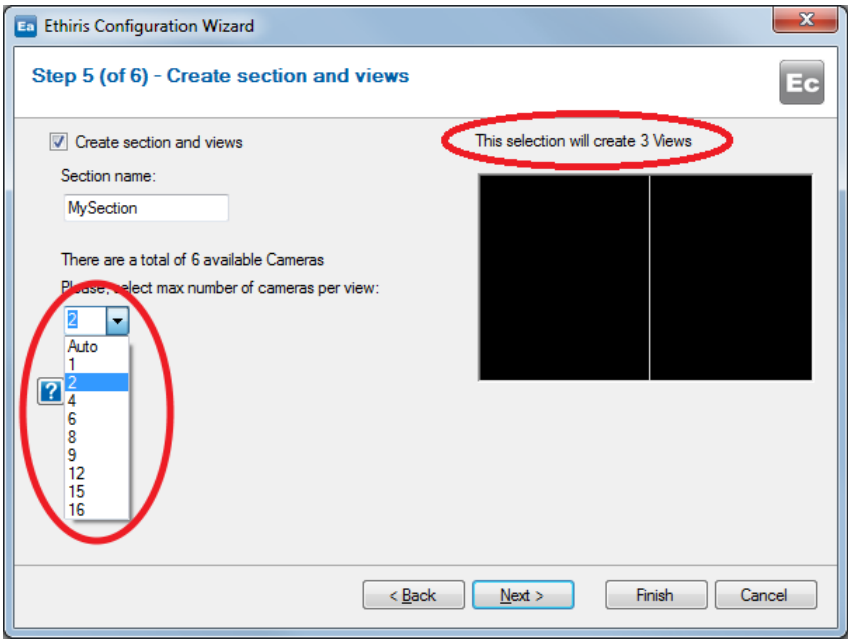
In this step, you can select to create a section with views to use for viewing images from cameras.

You can select a name for the section, which will be used both in the Ethiris Admin treeview and also in Ethiris Client as the name for the section.

You can also decide how the layout of the views will be. You simply select the number of cameras to display per view. The list with the number of cameras will look different depending on the number of cameras added in step 2.

The number of views created is based on the number of cameras you want per view in relation to the number of cameras added. The wizard makes sure a sufficient number of views are created to cover all cameras.

In the example below, 6 cameras have been added, and I have selected two cameras per view, which results in the wizard creating three views.


Ethiris Configuration Wizard
Step 5 (of 6) - Create section and views
Auto
1
2
4
6
8
9
12
15
16
< Back
Next >
Finish
Cancel

Create section and views [✓]

This selection will create 3 Views

Section name: MySection

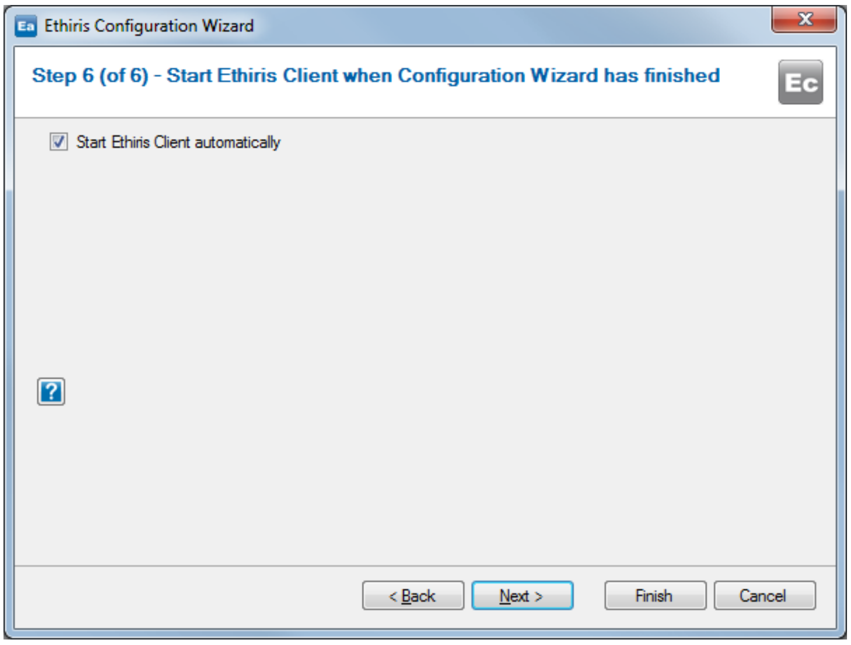
There are a total of 6 available Cameras

Please, select max number of cameras per view: 2

Figure 2.19 3 views will be created.

The selection *Auto* means that the guide will create the number of 4x2 views necessary to cover all cameras and a sufficient number of 3x2 views to cover all cameras. These views will be placed in a section called *Overviews.* The guide will also create an adequate number of 5x4 views with a hotspot and an adequate number of 4x3 views with a hotspot to cover all Cameras. These views will be placed in a section called *Hotspots.* Finally, the guide will create the number of 2x2 views needed to cover all cameras and place them in a section called *Quads.* Later you can delete the views you don't need.

When you are content with the settings, click *Next* to go to the last step.


Ethiris Configuration Wizard

Step 6 (of 6) - Start Ethiris Client when Configuration Wizard has finished

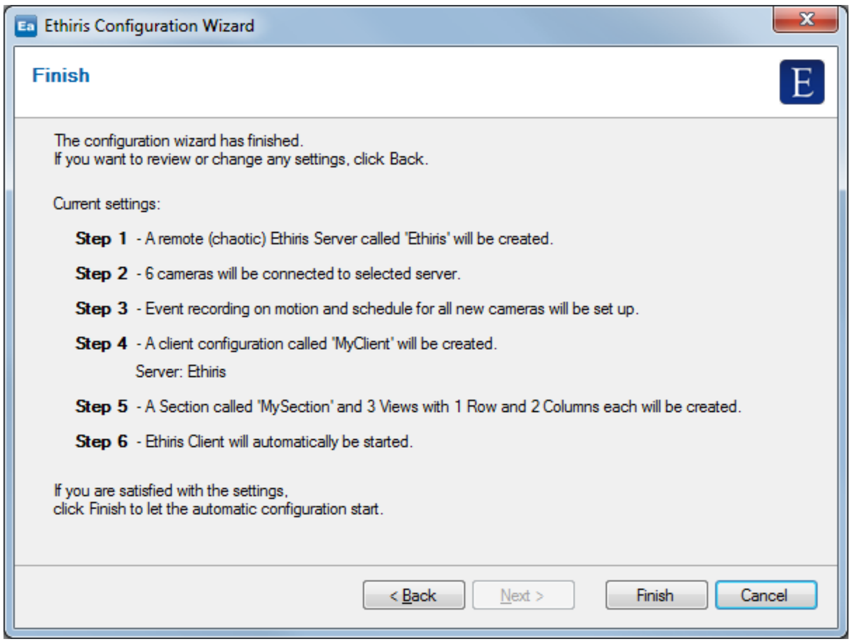
☑ Start Ethiris Client automatically

Figure 2.20 Select if Ethiris will be started automatically.

This choice is only available if Ethiris Client is installed locally.

Obviously, you can start Ethiris Client in other ways, e.g., via the start menu or by right-clicking the client configuration in the Ethiris Admin treeview and select the menu *Open configuration in local client*.

Click *Next* to move on to the summary.



The configuration wizard has finished.
If you want to review or change any settings, click Back.

Current settings:

Step 1 - A remote (chaotic) Ethiris Server called 'Ethiris' will be created.

Step 2 - 6 cameras will be connected to selected server.

Step 3 - Event recording on motion and schedule for all new cameras will be set up.

Step 4 - A client configuration called 'MyClient' will be created.
Server: Ethiris

Step 5 - A Section called 'MySection' and 3 Views with 1 Row and 2 Columns each will be created.

Step 6 - Ethiris Client will automatically be started.

If you are satisfied with the settings,
click Finish to let the automatic configuration start.

Figure 2.21 Summary of your selections.

Here you have the last opportunity to click *Back* to change any of your selections.

If you are content, click *Finish* to let the wizard create the configuration.

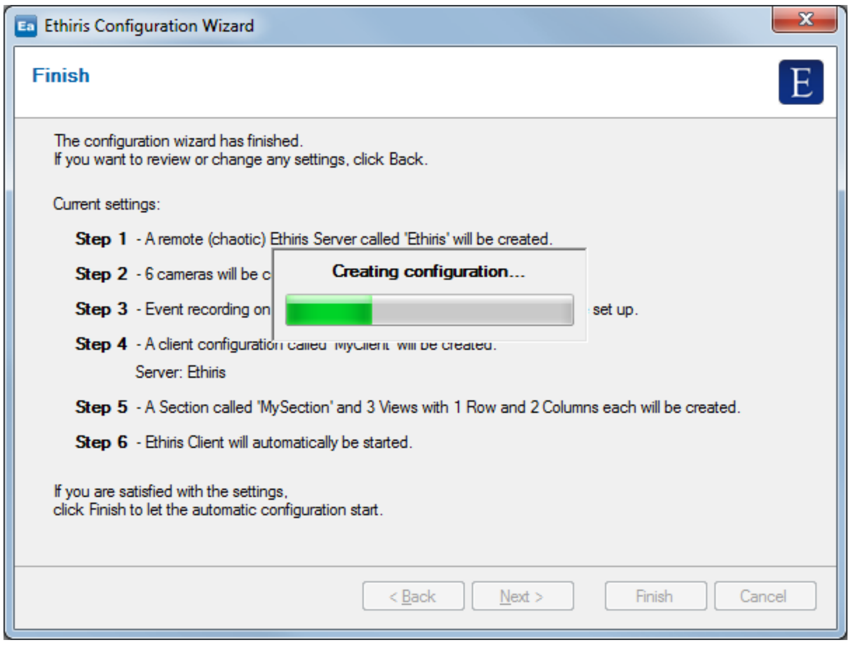
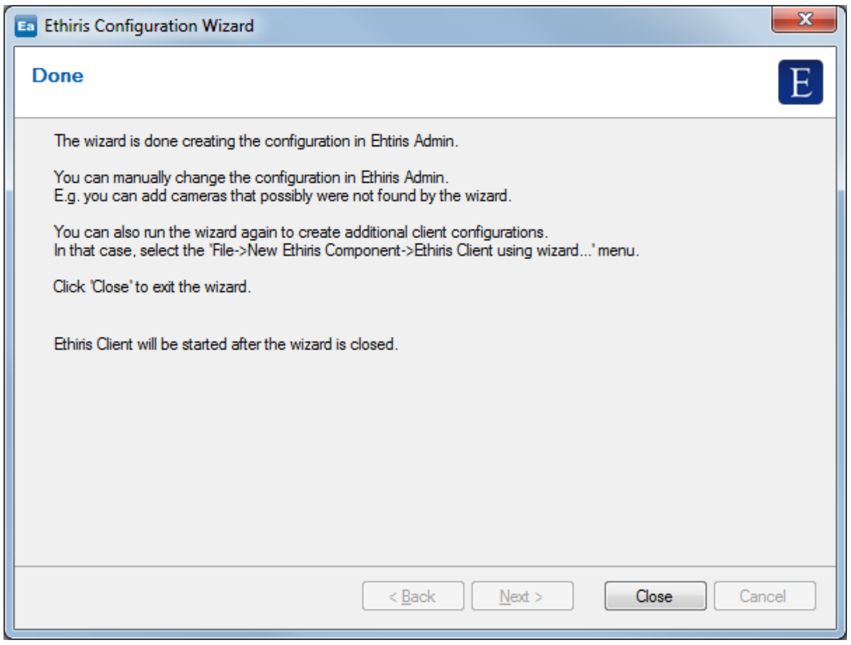


Figure 2.22 The wizard creates the configuration.

While the wizard creates the configuration, you can see the progress in Ethiris Admin.

When the work is done, a dialog is displayed that confirms that the configuration is created, and (if you selected this option), Ethiris Client will be launched with the newly created client configuration as soon as you close the wizard.



The wizard is done creating the configuration in Ehtiris Admin.

You can manually change the configuration in Ethiris Admin.
E.g. you can add cameras that possibly were not found by the wizard.

You can also run the wizard again to create additional client configurations.
In that case, select the 'File->New Ethiris Component->Ethiris Client using wizard...' menu.

Click 'Close' to exit the wizard.

Ethiris Client will be started after the wizard is closed.

Figure 2.23 The wizard is done.

The Ethiris configuration wizard is an excellent way to get started with Ethiris quickly. You can use the automatically created configuration as a starting point and then change the settings directly in Ethiris Admin after the wizard is done. It is usually easier to fine-tune an existing configuration compared to creating it from scratch.

In the next section, we assume you have canceled the wizard and have an empty configuration in Ethiris Admin.

Manually connecting to the local Ethiris Server

If you choose to cancel the configuration wizard, the *Ethiris Explorer* will be empty, except for the *Ethiris Components* node.

Before we go on and explore more windows in Ethiris Admin, we need a configuration to work with. So, if you didn't complete the configuration wizard or, for some other reason, have an empty configuration, you may follow the instructions below to get something to work with.

Select the *File->Connect Ethiris Component->Ethiris Server…* menu item, see *Figure 2.24*.

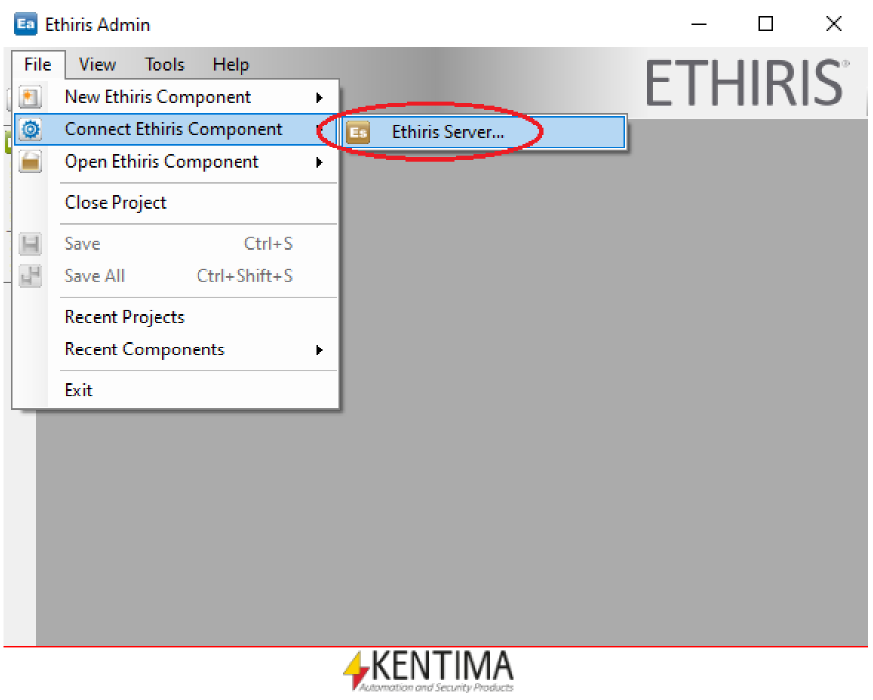


Figure 2.24 Connect to the local Ethiris Server.

In the *Connect dialog*, just **click** the *Connect* button.

By connecting to an Ethiris Server, the current server configuration is read and displayed in the *Ethiris Explorer* treeview, see *Figure 2.25*.

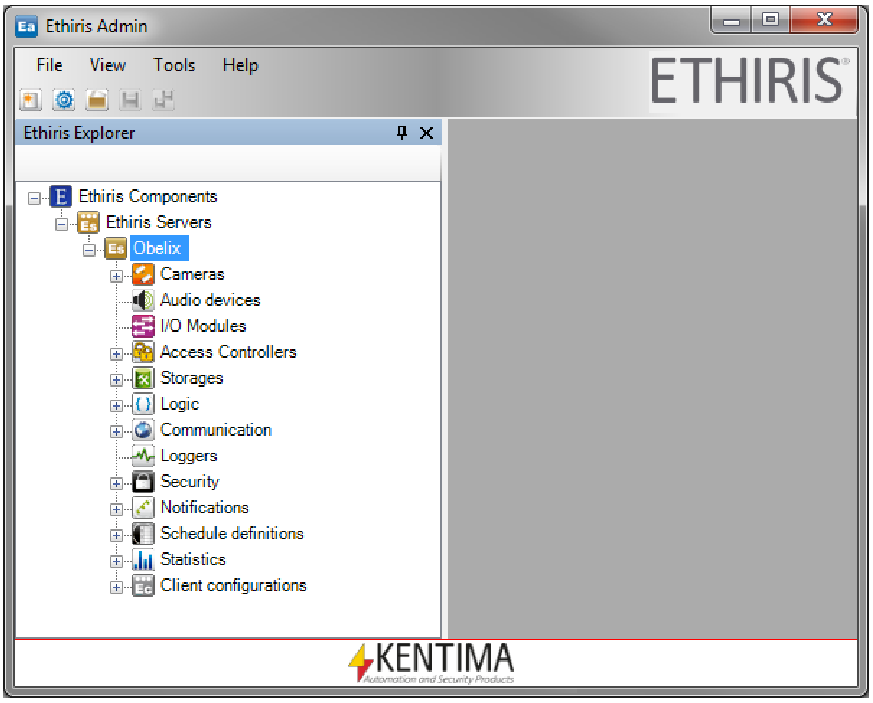
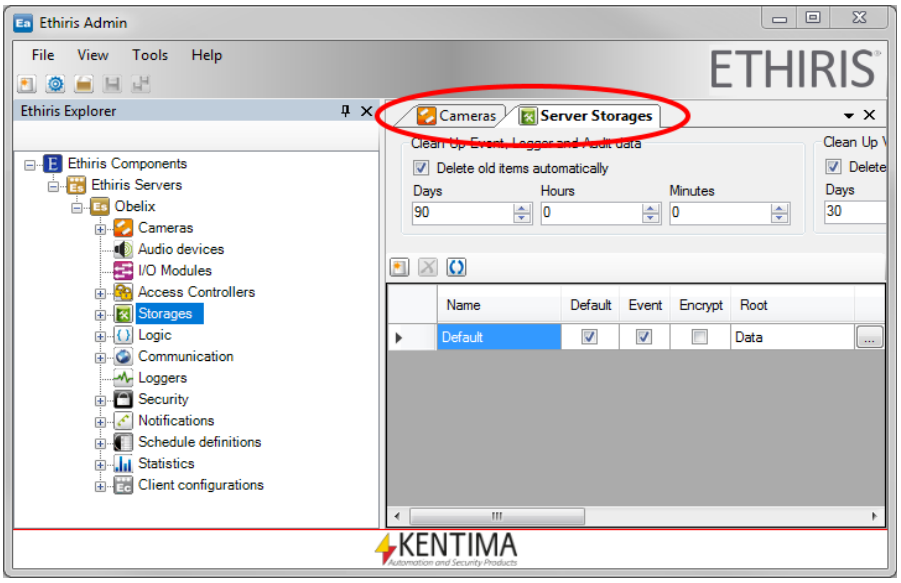


Figure 2.25 The local Ethiris Server is connected, and the current configuration is displayed in the treeview.

Many of the nodes in the treeview have a corresponding panel (property window) that can be opened by double-clicking the node in the treeview. If we, for example, double-click the *Cameras* node and then the *Storages* node, two panels will be opened in the area to the right in the main frame, see *Figure 2.26*.



Clean Up Event, Logger and Audit data

☑ Delete old items automatically

Days	Hours	Minutes
90	0	0

Clean Up

☑ Delete

Days: 30

Name	Default	Event	Encrypt	Root
Default	☑	☑	☐	Data

Figure 2.26 Two panels are opened in Ethiris Admin.

By default, all panels are opened in a *tab group*. In a tab group, only one tab at the time can be visible. In the example above, the *Server Storages panel* is on top and visible. To view another panel in the tab group, simply click the corresponding tab.

If you, e.g., click the Cameras tab, this panel becomes visible instead. See *Figure 2.27*.

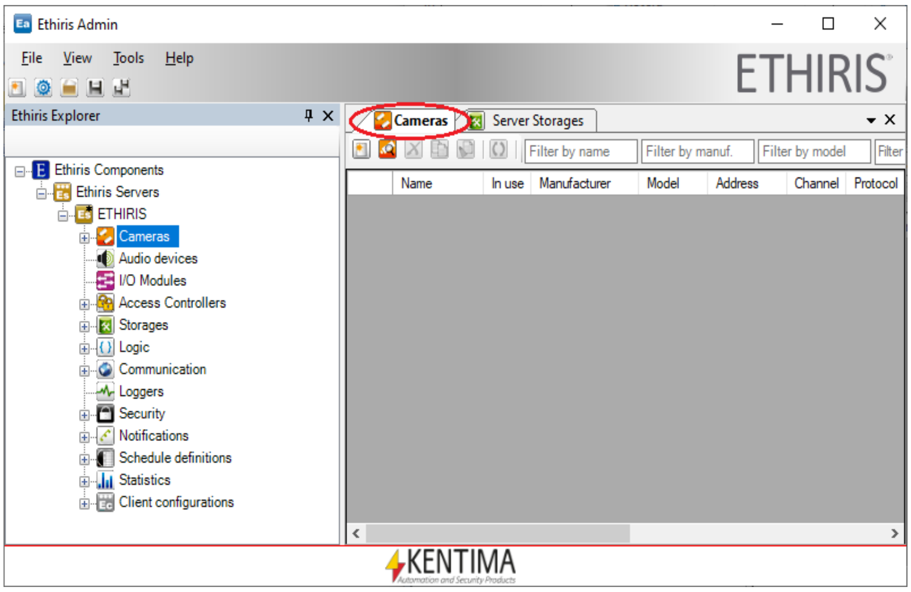


Figure 2.27 Cameras panel on top.

Now, not every user may like the default layout of the various windows. In the next section, we will look at the different possibilities of customizing the user environment.

2.1.2 Changing the layout

Every window in Ethiris Admin can be moved, resized, and docked at various positions. They can even be torn away from the main frame and become floating windows that can be positioned anywhere, even onto another monitor.

As soon as you start moving a panel (by clicking the tab with the windows name, hold the mouse button down and moving the mouse), the whole window becomes translucent. Some visual guides also appear, see *Figure 2.28*.

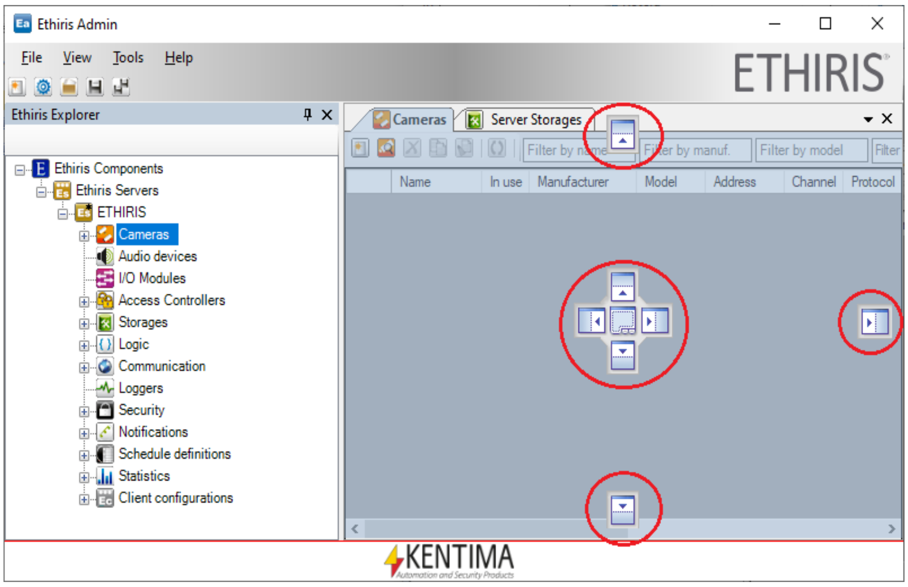


Figure 2.28 About to move the Cameras panel.

As you continue to move the mouse (while still holding down the left mouse button), you get visual feedback on how the window will be positioned when you release the mouse button. In *Figure 2.29,* the mouse pointer is over the visual guide marked by a red ring.

If you change your mind and don't want to move the window, simply press the *Esc* key or right-click the mouse, and the move is aborted.

Press the Esc key to abort the window move.

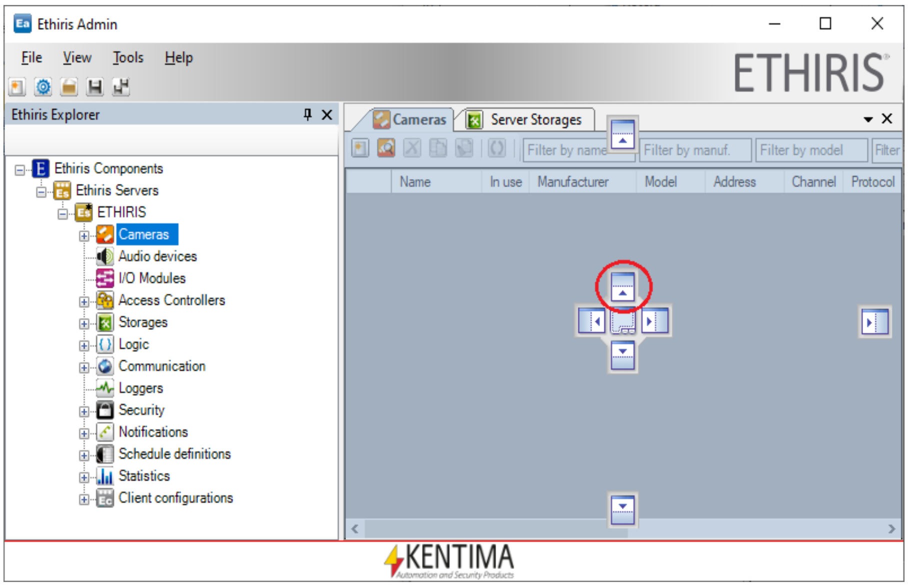
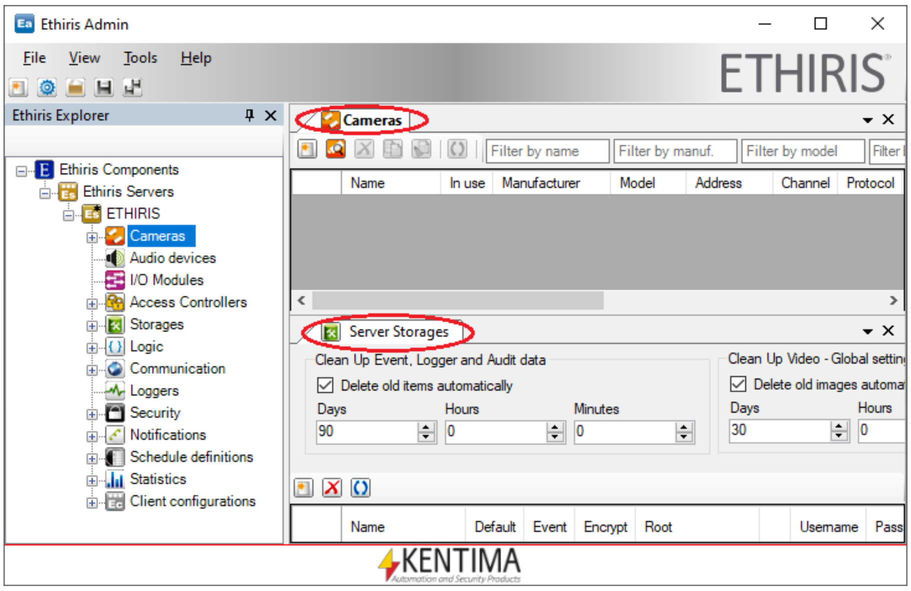


Figure 2.29 Mouse pointer over the "top" visual guide.

Notice how the translucent blue indication shows that if you release the mouse button in this position, the window will that above the other panel. See *Figure 2.30* for an example of how this would look like.



Clean Up Event, Logger and Audit data
☑ Delete old items automatically
Days 90 Hours 0 Minutes 0

Clean Up Video - Global settin
☑ Delete old images automa
Days 30 Hours 0

Figure 2.30 Cameras panel is now moved above the Server Storages panel.

Now, the Cameras panel is in its own tab group above the lower tab group containing the remaining Server Storage panel. It is now possible to view two document windows at the same time. The area is, of course, smaller for each one of them since they have to share what is left in the main frame outside the Ethiris Explorer tool window.

Between the two tab groups is a splitter bar, which you can move to change the vertical size of the two groups, see *Figure 2.31*.

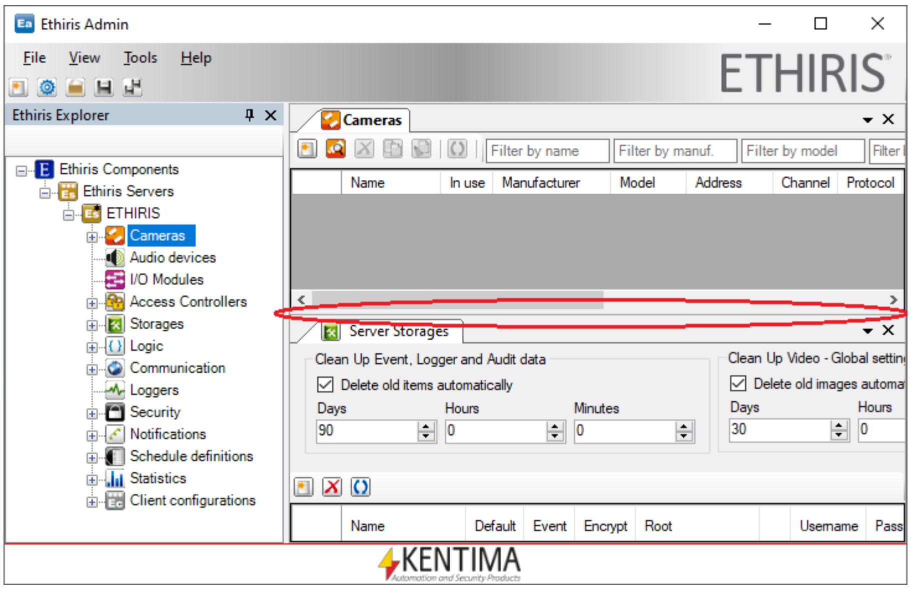


Figure 2.31 Move the splitter bar up/down to resize the windows.

If you want the *Cameras* panel back to the original tab group, you can simply drag the window and drop it on the center visual guide. Note that you have to move the mouse pointer over the lower tab group to make the visual guides visible, see *Figure 2.32*.

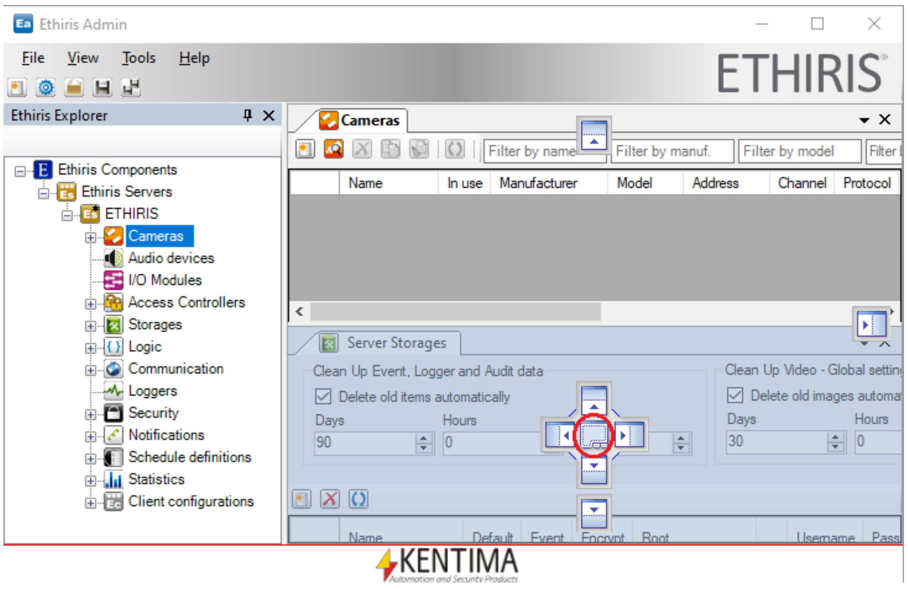


Figure 2.32 Moving the Cameras panel back again.

Change the order of tabs

As you move the *Cameras* panel back to the original tab group, it is positioned last instead of first. To move it back to the first position (left-most position), drag the tab and drop in onto the Server Storages (first) tab, see *Figure 2.33*. In this manner, you can change the order of any panel within a tab group.

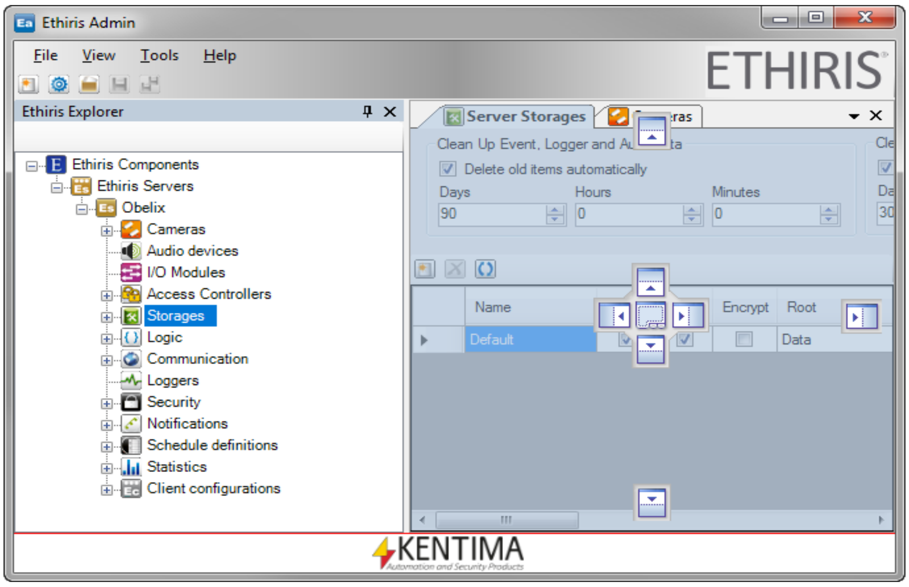


Figure 2.33 Moving the Cameras panel to the left-most position.

Visual guides

When moving a window to a tab group, i.e., onto another window, the visual guides, as in *Figure 2.34,* appears.

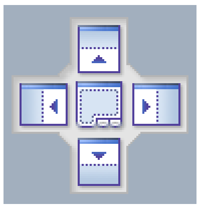


Figure 2.34 Visual guides with 5 different positions.

There are 5 alternative positions. If you release the mouse button over the center guide, the dragged window will become a new tab in the group. If you release the mouse button over the Left, Top, Right, or Bottom guide, the window will be positioned in a new tab group, splitting the current tab group area into two equally large areas.

Docking a window

Another alternative is to dock the window to any of the edges of the main frame. Only the sides where no tool window is pinned are available. In *Figure 2.35,* there are only available at the top, right, and bottom edge of the main frame for docking.

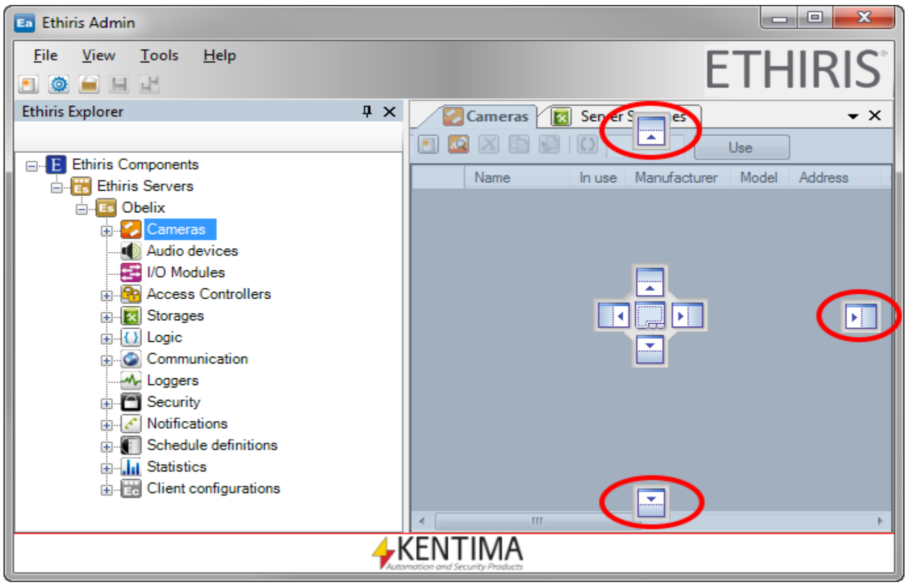
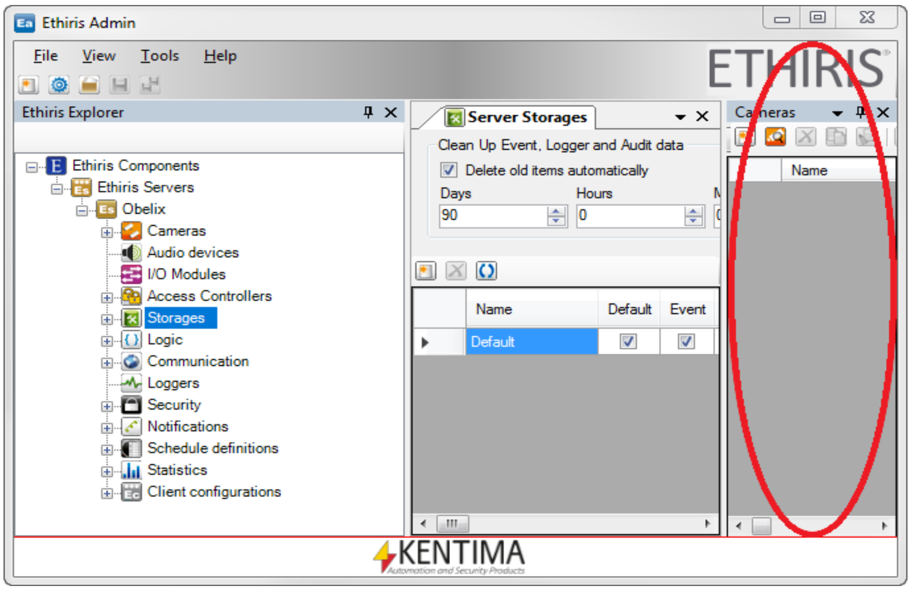


Figure 2.35 You can dock the window on top, right, or bottom.

When you dock a window, it automatically becomes a tool window that can be pinned and unpinned. When you dock a window, it is pinned by default, see *Figure 2.36*.



Clean Up Event, Logger and Audit data

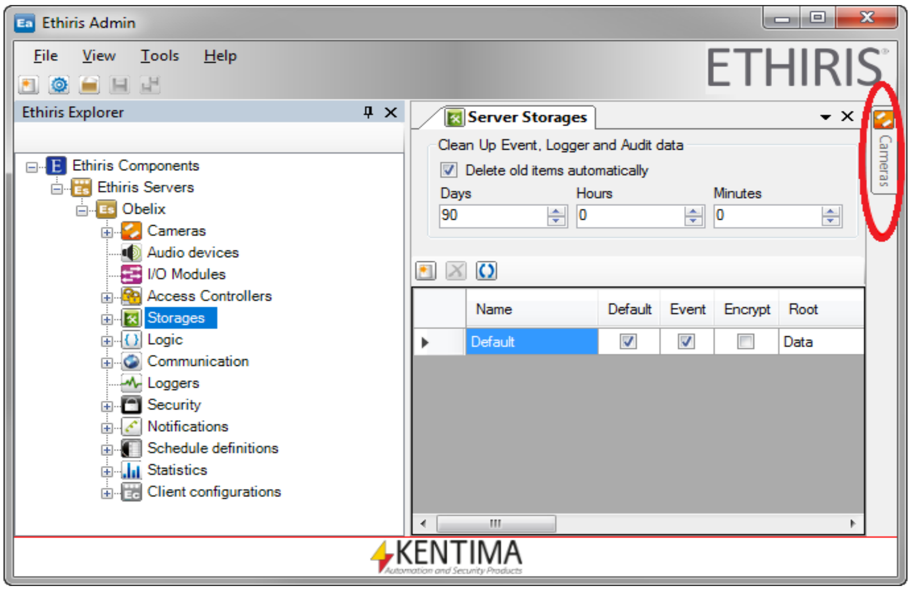
☑ Delete old items automatically

Days	Hours
90	0

Name	Default	Event
Default	✓	✓

Figure 2.36 Cameras panel as a tool window docked and pinned at the right edge of the main frame.

To make the tool window slide in, unpin it, see *Figure 2.37*.



Clean Up Event, Logger and Audit data

☑ Delete old items automatically

Days	Hours	Minutes
90	0	0

Name	Default	Event	Encrypt	Root
Default	☑	☑	☐	Data

Figure 2.37 Cameras panel unpinned at the right of the main frame.

Tab groups

When it comes to the original tool window (Ethiris Explorer), it cannot be positioned in the central tab group, i.e., onto another document window. It can only be dragged to another tool window or docked to one of the edges of the main frame.

Let's say we would like to move the Cameras panel to the Ethiris Explorer tab group. Just drag the Cameras window and drop it onto the center visual guide on the Ethiris Explorer window. See *Figure 2.38*.

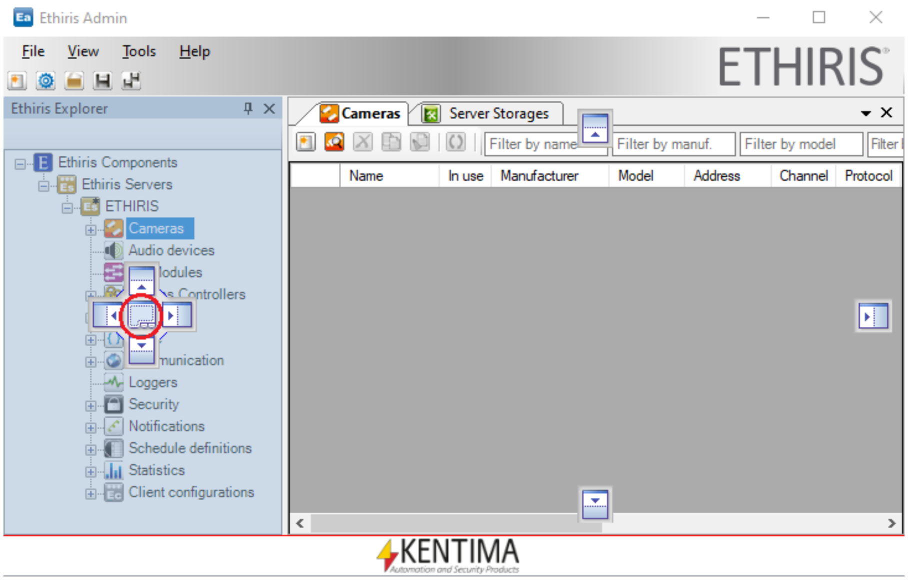


Figure 2.38 Cameras will be moved to the Ethiris Explorer window.

In this case, the Cameras window becomes a new tab in the tab group. Now either the Cameras or the Ethiris Explorer is visible, not both at the same time, see *Figure 2.39*.

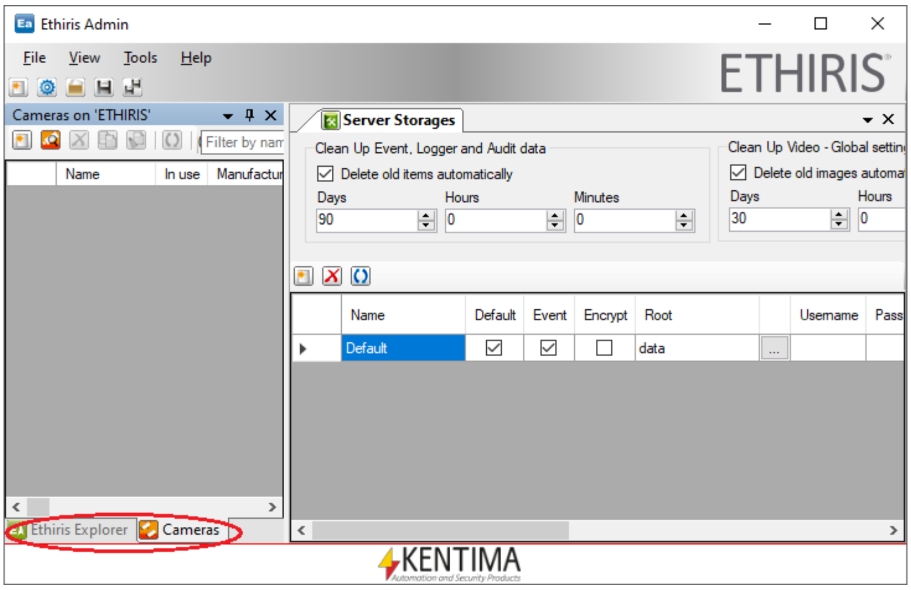


Figure 2.39 Cameras is a tab together with the Ethiris Explorer.

Now the Server Storages panel and the other central document windows have more screen real-estate to the right.

Floating windows

You may not be satisfied with this. Let's tear away the tool window altogether. If you want to move the entire tab group, grab the window title bar. If you only want to move one of the tabs, grab the specific tab instead. In the next example, we will move the whole tab group, i.e., both the *Ethiris Explorer* and the *Cameras,* and make it a floating window. See *Figure 2.40*.

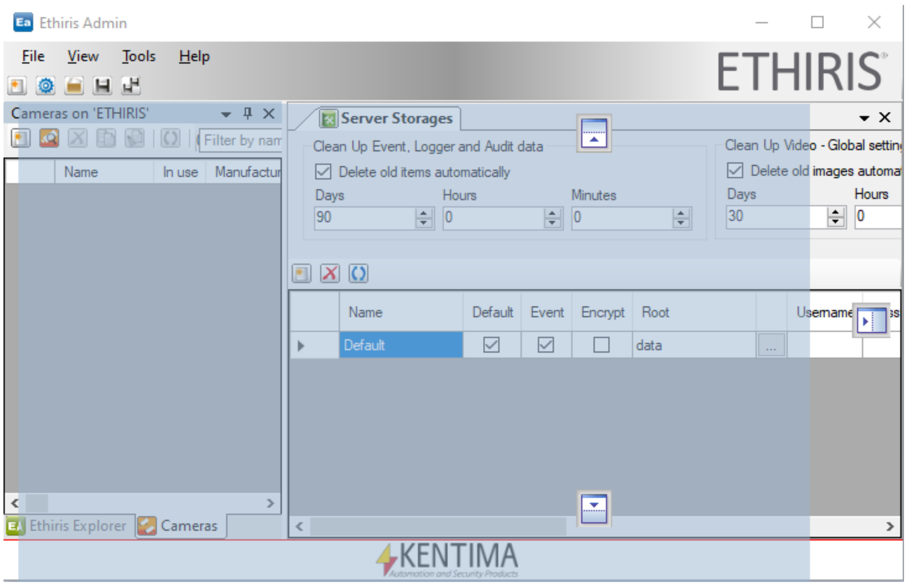


Figure 2.40 The Cameras and Ethiris Explorer tab group are torn away.

If you release the left mouse button in this state, the tool window will become floating, see *Figure 2.41*.

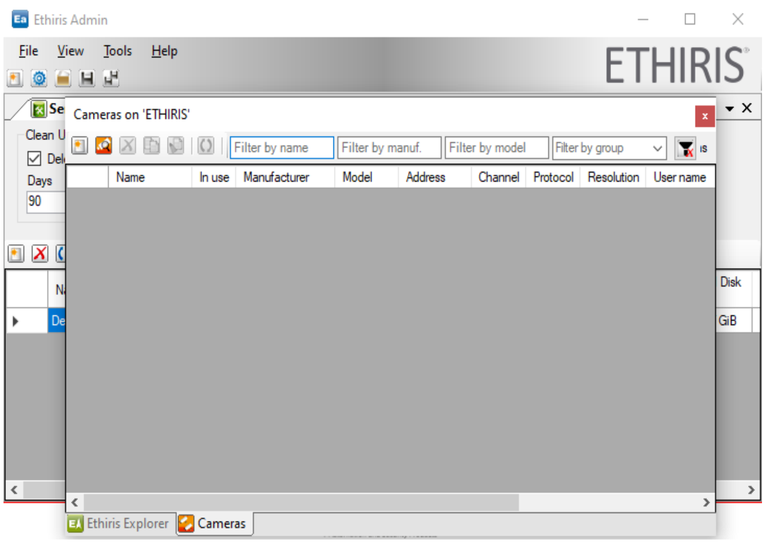


Figure 2.41 A floating tool window.

Double click the window title bar to move it to its previous position.

To move the window back to its previous position, double-click the title bar. Double-click the title bar again, and the window is repositioned again to the last position. This way, you can toggle between the current and previous positions.

Anyway, when the window is floating, you may resize it to the desired size, and you can position the window anywhere on any screen. Naturally, you can tear away any window and make it a floating window.

Tab list

In a tab group with *document windows,* there is a list of all the windows part of the tab group. There is a *down arrow* in the upper right in the tab group. Clicking the arrow opens a list of all the windows in the group.

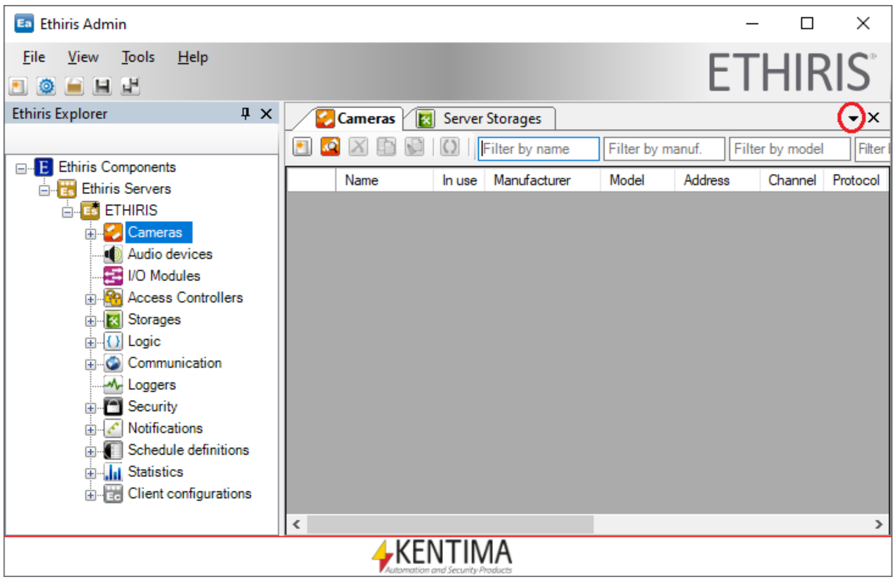


Figure 2.42 Show the tab list by clicking the down arrow in the upper right corner.

If all tabs don't fit, some tabs may be hidden. This is indicated by a line above the down arrow, see *Figure 2.43*.

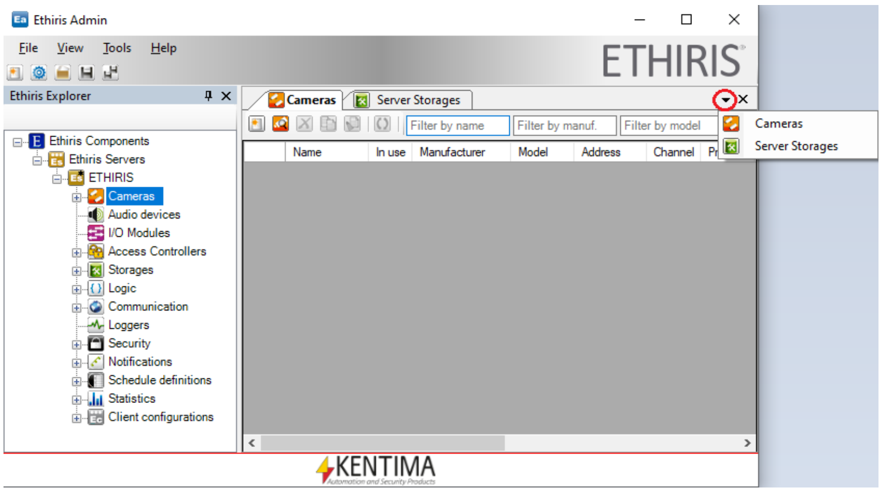


Figure 2.43 Hidden tabs are displayed in the window list.

Closing windows

Some of the windows may not be useful to you. Then, just close it by clicking the Close button in the window title bar. In *Figure 2.44,* the close buttons for the Ethiris Explorer tool window and the Server Storages panel are ringed in.

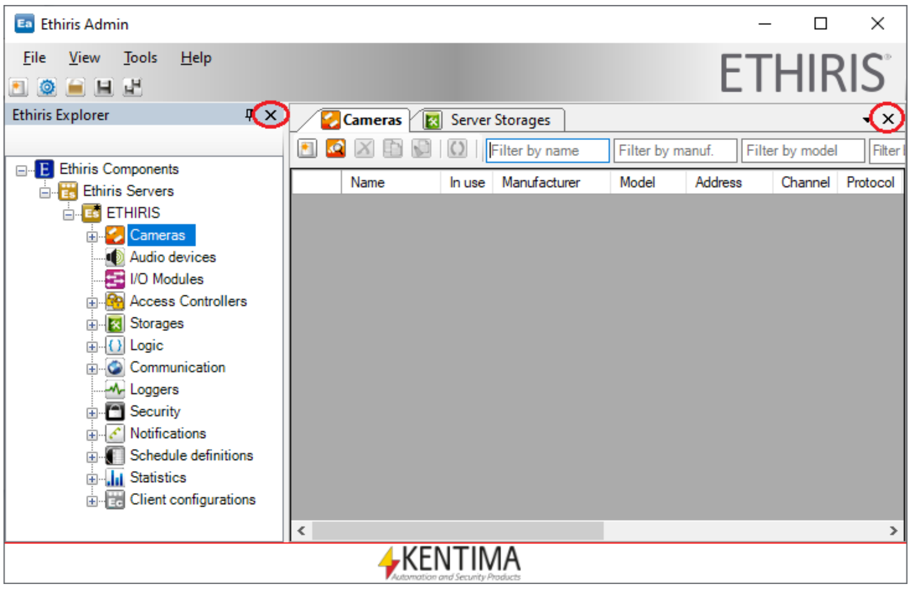


Figure 2.44 Close buttons of various windows.

You can also close one or several panels by using the popup menu in a panel.

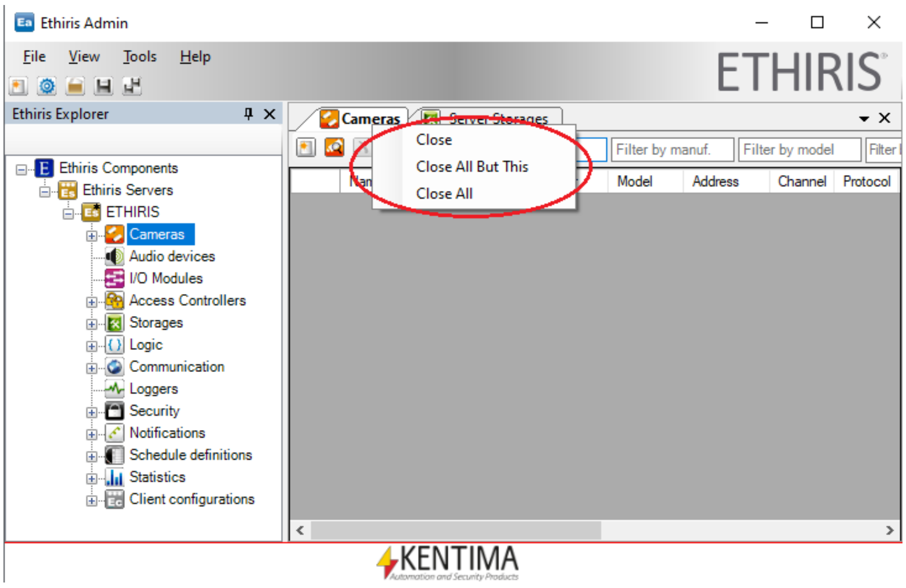


Figure 2.45 Panel popup menu.

Right-click the tab of any panel to popup the menu in *Figure 2.45*.

Close closes the panel you right-clicked on.

Close All But This closes all other open panels but the one you clicked.

Close All closes all open panels, including the one you clicked.

If you accidentally close a window and want it open again, the *Ethiris Explorer* tool window can be opened from the View menu. All other windows have to be opened by double-clicking the corresponding node in the Ethiris Explorer treeview.

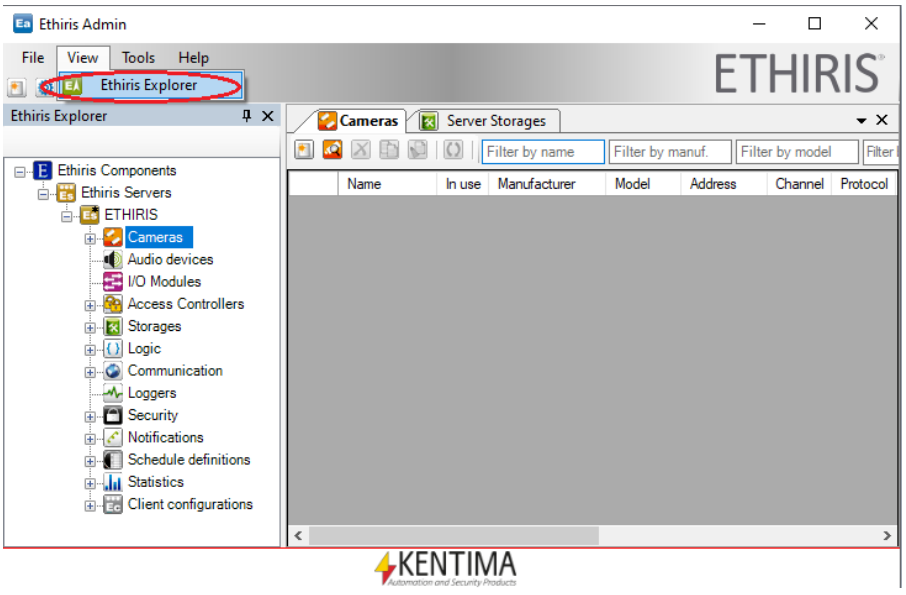


Figure 2.46 The View menu can be used to open Ethiris Explorer in Ethiris Admin.

2.2 Main menu bar

2.2.1 Overview

In this section, we will inspect all the menus and learn what each menu item is for. You can access the menus by selecting them with the mouse, but you can also press the *Alt* key on the keyboard to activate the menus.

When the Alt key is pressed, menu shortcuts appear (underlined letters). Pressing e g *Alt+F* opens the File menu.

You can also move between menus and menu items with the arrow keys. Press *Enter* to select a menu item.

Ethiris components

In Ethiris Admin, various *Ethiris components* are managed. These are *Ethiris Server*, *Ethiris Client,* and *Ethiris Admin Project*.

Ethiris Server and Ethiris Client are real executable programs while an *Ethiris Admin Project* is just a configuration file to keep track of the other components in a project.

Online vs. offline

At this point, it is appropriate to mention the difference between *online* and *offline*.

Online configuration means that Ethiris Admin connects to the Ethiris component online and *asks* for the current configuration. After editing when the configuration is saved, it is in effect sent back to the Ethiris component, and the changes have effect immediately.

Offline configuration, on the other hand, works towards a configuration file. The Ethiris component does not even need to be installed. Changes in the configuration will be saved to the configuration file. To have an effect, the corresponding Ethiris component has to be started, and the configuration file has to be loaded by the Ethiris component.

For the time being, not all Ethiris components support *online* configuration. Furthermore, not all Ethiris components support *offline* configuration.

Online configuration is supported by *Ethiris Server*.

Offline configuration is supported by *Ethiris Client*.

There is also a concept called an *Ethiris Project,* which, by design, is an offline component.

Keyboard shortcuts

A number of keyboard shortcuts may be of use. To access the keyboard shortcuts, press the Alt key. When Alt is pressed the available shortcuts appear as underlined letters in menus and buttons.

By default, only items in the main menu have keyboard shortcuts. These are:

Alt-F – File menu.

Alt-E – Exit menu item in the File menu.

Alt-V – View menu.

Alt-T – Tools menu.

Alt-H – Help menu.

2.2.2 File menu

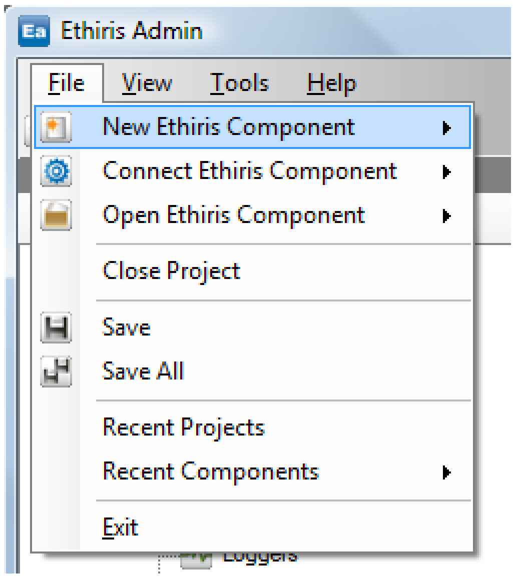


Figure 2.47 The File menu in Ethiris Admin.

File->New Ethiris Component is used for creating a new configuration file *offline* for an Ethiris component. This is typically used to create a new Ethiris Client configuration from the very beginning.

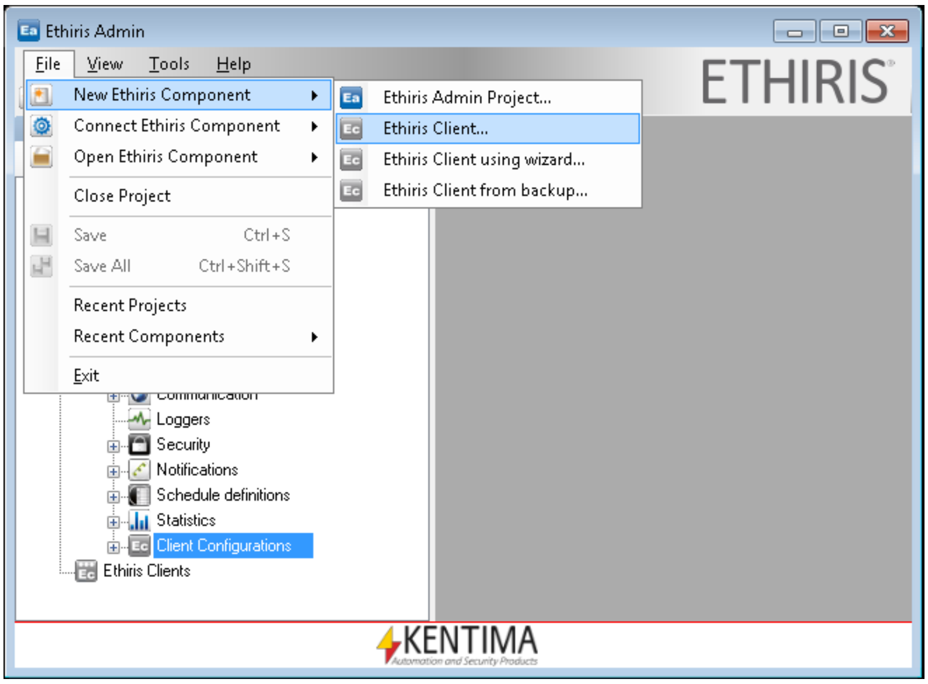


Figure 2.48 The New Ethiris Component menu in Ethiris Admin.

As discussed above, only *Ethiris Admin Project* and *Ethiris Client* are available for *offline* configuration. A client configuration will be created on the selected server, see *Figure 2.49*.

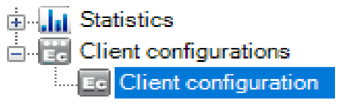


Figure 2.49 A new Ethiris Client configuration was created.

Double click on the node *Client configurations* in the treeview to open the associated panel.

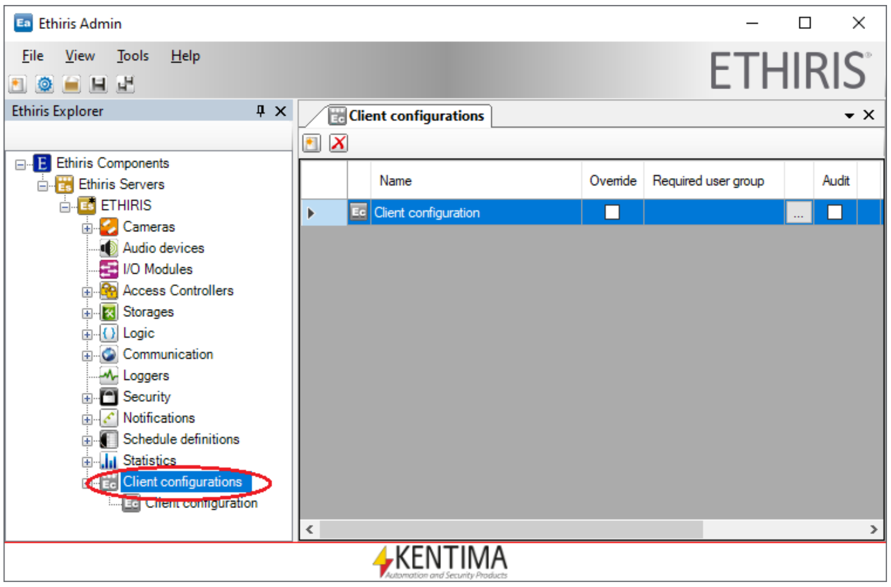
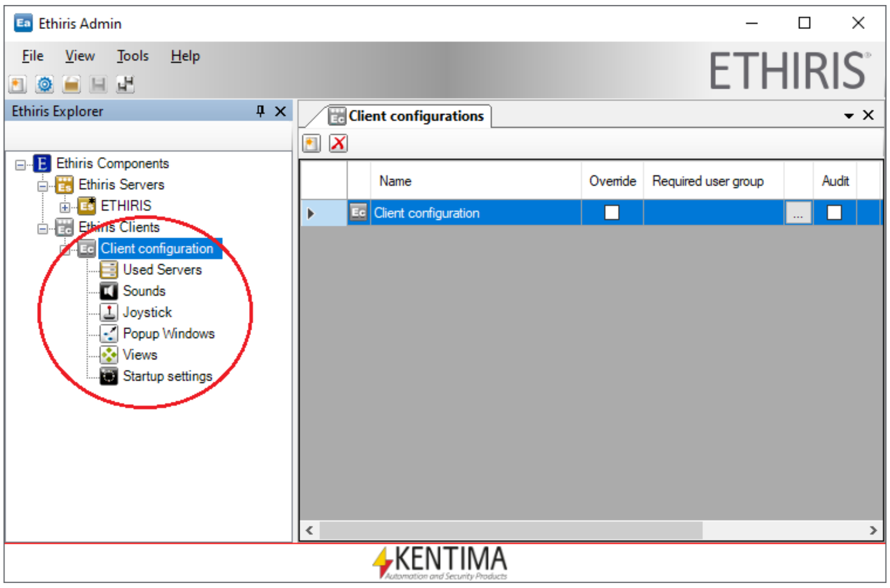


Figure 2.50 The panel Client configurations.

Here you can set a privilege requirement so that users that do not have the required privilege (is a member of the specified user group) cannot see the client configuration.

The client configurations are handled by a *configuration server*, an Ethiris Server. To open the client configuration in Ethiris Admin, just double click on it in the treeview.

You can rename the client configuration either in the Client configurations panel or directly in the client configurations when the configuration is open in Admin.



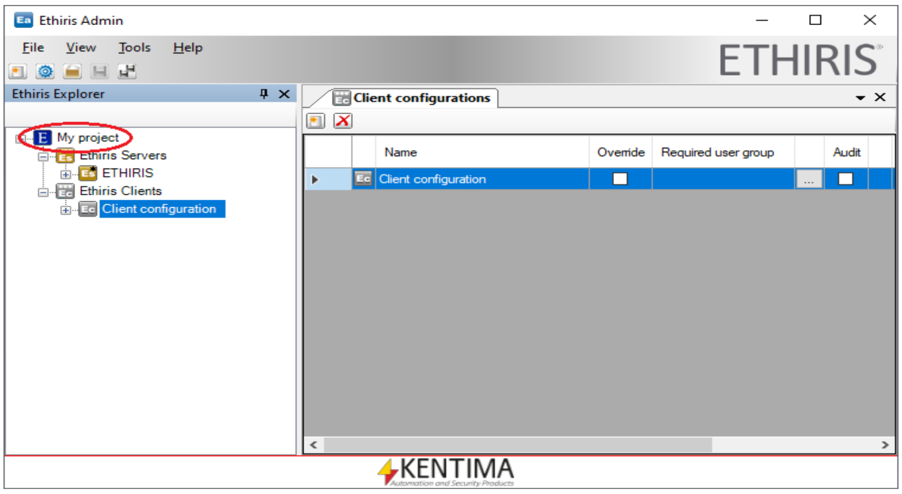
Name	Override	Required user group	Audit
Client configuration	☐	...	☐

Figure 2.51 A new Ethiris Client component created.

If you choose to select the menu, *Ethiris Client using wizard…* the configuration wizard will be started in step 4, i.e., you can't add additional Ethiris Servers via the wizard but only create new client configurations.

The client configuration created by the wizard will connect to all Ethiris Servers in the Ethiris Admin treeview. It may be more than one.

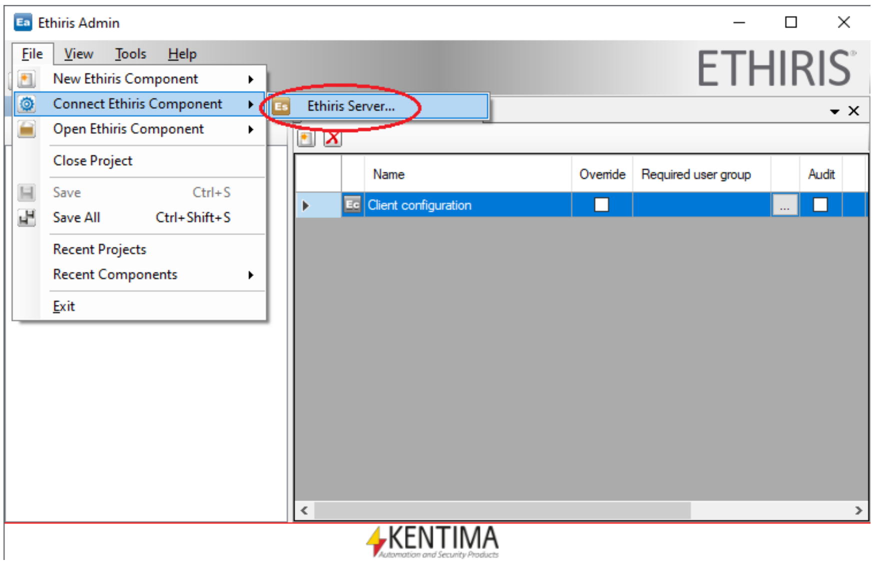
Should you choose to create a new *Ethiris Admin Project,* the current project is closed, and a new project is created containing the current Ethiris components in the treeview. In the example, in *Figure 2.52,* we have created a new project and given it the name *My new project*.



Name	Override	Required user group	Audit
Client configuration	☐	...	☐

Figure 2.52 A new project is created.

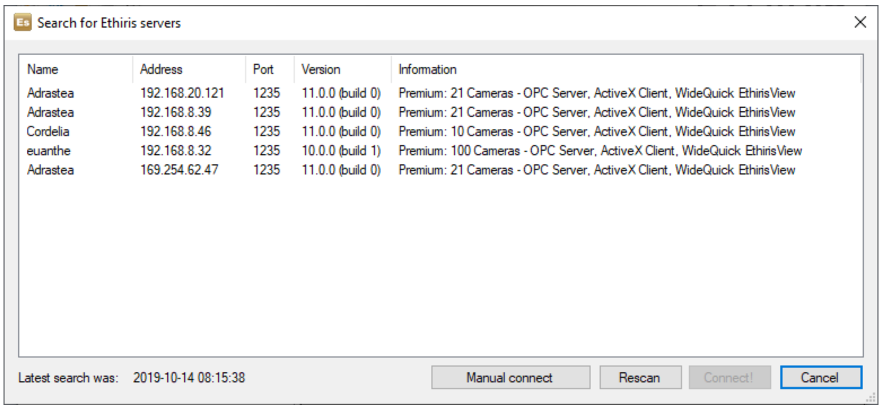
File->Connect Ethiris Component is used for connecting *online* to an Ethiris component. For the time being, an online connection is only available for *Ethiris Server*.



Name	Override	Required user group	Audit
Client configuration	☐		☐

Figure 2.53 The Connect Ethiris Component menu in Ethiris Admin.

A *Connect Component* dialog opens, see *Figure 2.49*.



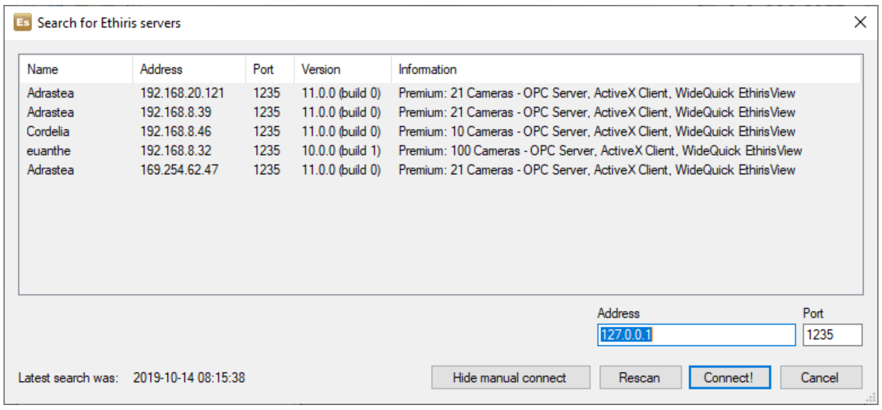
Name	Address	Port	Version	Information
Adrastea	192.168.20.121	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Adrastea	192.168.8.39	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Cordelia	192.168.8.46	1235	11.0.0 (build 0)	Premium: 10 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
euanthe	192.168.8.32	1235	10.0.0 (build 1)	Premium: 100 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Adrastea	169.254.62.47	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView

Latest search was: 2019-10-14 08:15:38

Figure 2.54 The Connect Ethiris Component dialog in Ethiris Admin.

All *Ethiris Servers* that *Ethiris Admin* is able to find are listed, click on the one you would like to connect to, and then on *Connect*. If the server you are looking for does not appear, you can click on *Manual connect*.

Note that *Ethiris Admin* does not search for servers when the dialog is opened. If the server you are looking for was started after Admin was, you might want to try a *Rescan*.



Name	Address	Port	Version	Information
Adrastea	192.168.20.121	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Adrastea	192.168.8.39	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Cordelia	192.168.8.46	1235	11.0.0 (build 0)	Premium: 10 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
euanthe	192.168.8.32	1235	10.0.0 (build 1)	Premium: 100 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Adrastea	169.254.62.47	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView

Address: 127.0.0.1
Port: 1235

Latest search was: 2019-10-14 08:15:38

Figure 2.55 The Connect Ethiris Component dialog set to Manual Connect.

Address is the *IP-address* of the computer where Ethiris Server runs. *127.0.0.1* is an alias for the local computer. You may also enter the computer name, e.g., *Galatea* as long as there is a *DNS* in the network resolving the name.

Port is the port the Ethiris Server listens to for incoming connection requests. This is by default *1235* and should rarely need to be changed.

Click *Connect* to connect online to the Ethiris Server.

After connection, the Ethiris Server component is added to the Ethiris Explorer treeview.

File->Open Ethiris Component is used for opening an existing configuration file for an Ethiris component *offline*. For the moment *Ethiris Client* and *Ethiris Admin project* are available.

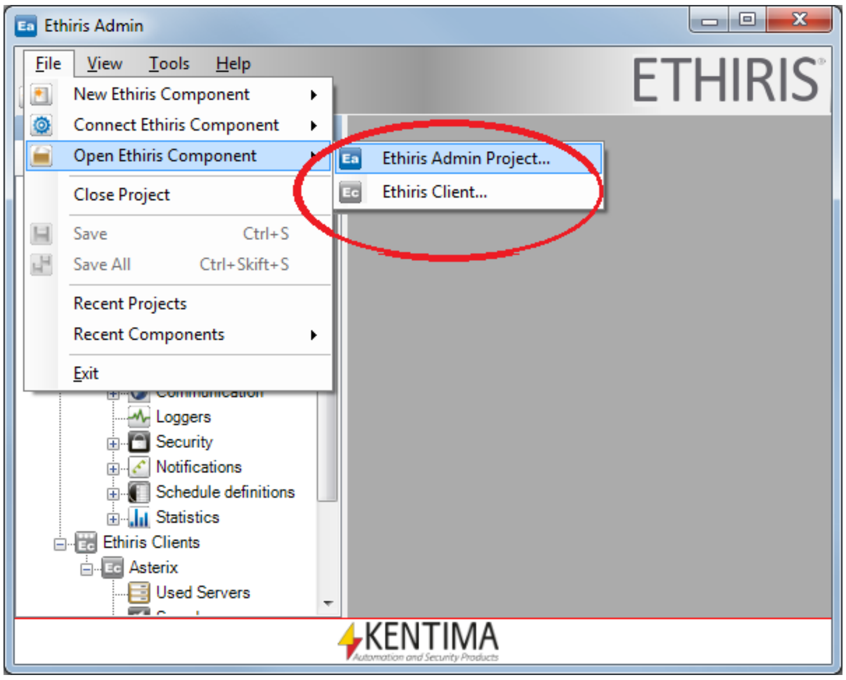
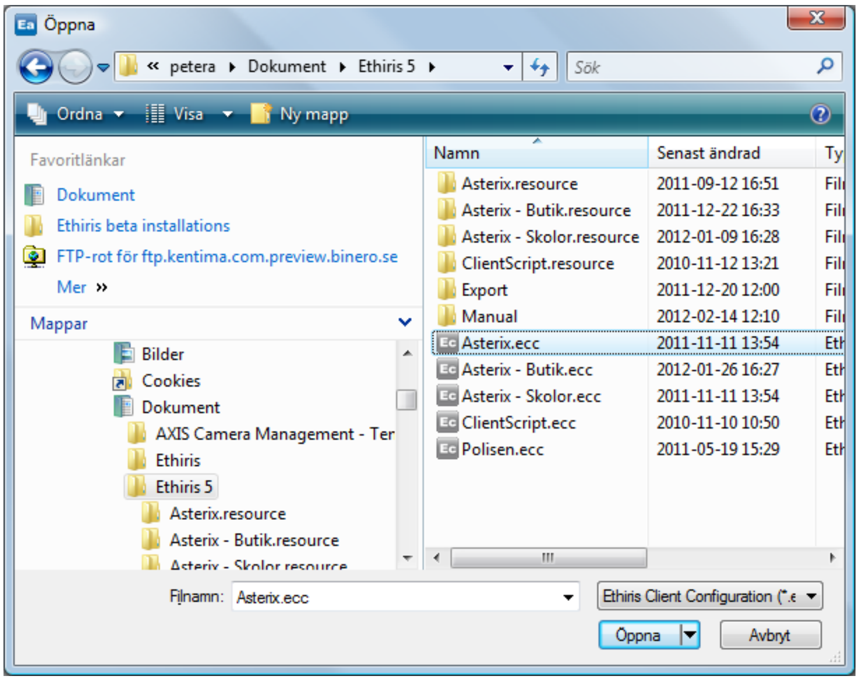


Figure 2.56 The Open Ethiris Component menu in Ethiris Admin.

File->Open Ethiris Component->Ethiris Client... is used for importing an existing old configuration saved in a file to the server that is selected in the treeview.

An *Open Component* dialog opens, see *Figure 2.57*.


Öppna
« petera ▸ Dokument ▸ Ethiris 5 ▸
Sök
Ordna
Visa
Ny mapp
Favoritlänkar
Dokument
Ethiris beta installations
FTP-rot för ftp.kentima.com.preview.binero.se
Mer »
Mappar
Bilder
Cookies
Dokument
AXIS Camera Management - Ter
Ethiris
Ethiris 5
Asterix.resource
Asterix - Butik.resource
Asterix - Skolor.resource
Filnamn:
Ethiris Client Configuration (*.e
Öppna
Avbryt

Namn	Senast ändrad	Ty
Asterix.resource	2011-09-12 16:51	Fil
Asterix - Butik.resource	2011-12-22 16:33	Fil
Asterix - Skolor.resource	2012-01-09 16:28	Fil
ClientScript.resource	2010-11-12 13:21	Fil
Export	2011-12-20 12:00	Fil
Manual	2012-02-14 12:10	Fil
Asterix.ecc	2011-11-11 13:54	Eth
Asterix - Butik.ecc	2012-01-26 16:27	Eth
Asterix - Skolor.ecc	2011-11-11 13:54	Eth
ClientScript.ecc	2010-11-10 10:50	Eth
Polisen.ecc	2011-05-19 15:29	Eth

Filnamn: Asterix.ecc

Figure 2.57 The Open Ethiris Component dialog in Ethiris Admin.

Browse for the desired file, select it, and then click the *Open* button to import the configuration file. The corresponding Ethiris component is added to the Ethiris Explorer treeview under *Client configurations* for the selected server.

File->Close Project is used for closing the current project. In effect, the whole Ethiris Explorer is cleared, and Ethiris Admin looks as it does the first time it is started. See *Figure 2.58*.

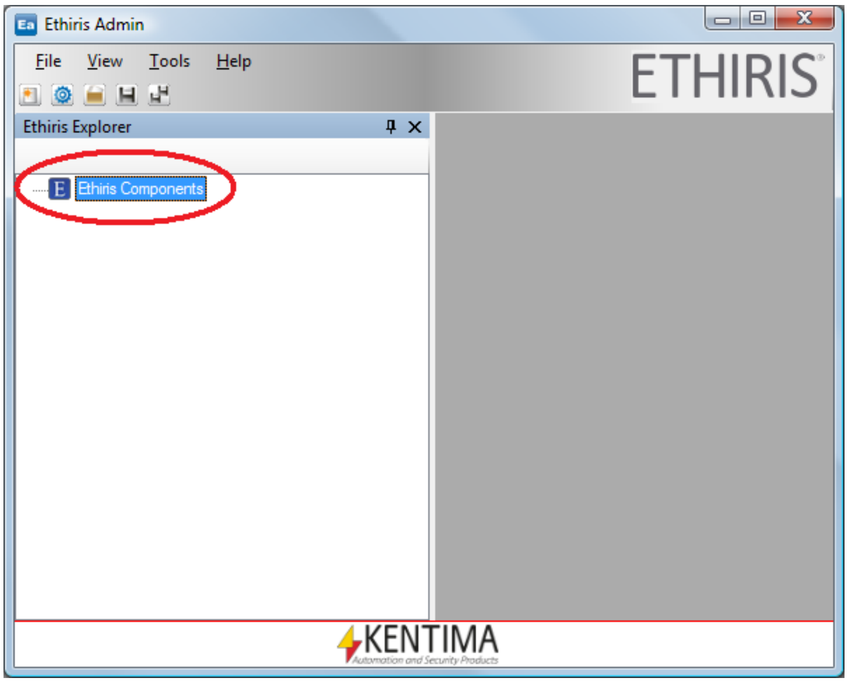


Figure 2.58 Ethiris Admin after selecting the Close Project menu item.

File->Save saves the currently selected component in the Ethiris Explorer treeview. In *Figure 2.59,* the Ethiris Server component *Obelix* will be saved since a sub-node of it is selected in the treeview.

Note the little asterix in the Ethiris Server icon. It indicates that something in the configuration has changed and the configuration needs to be saved.

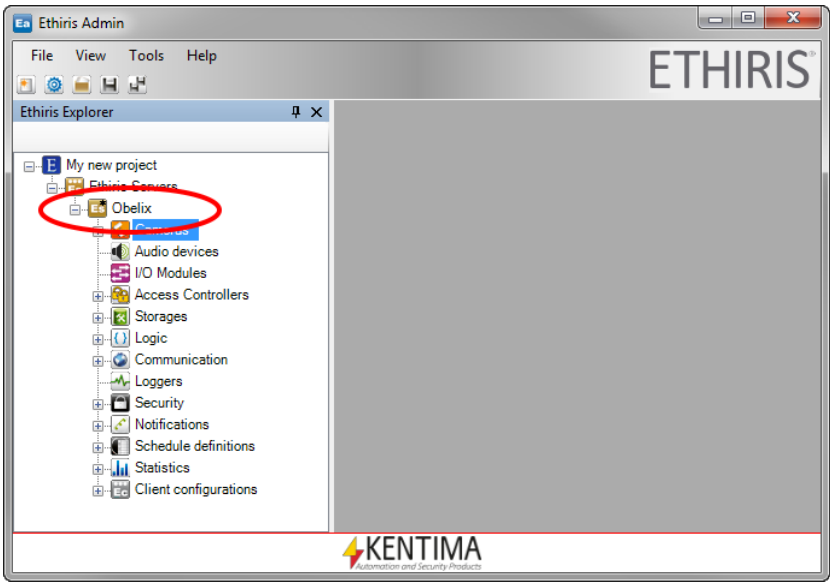


Figure 2.59 The Ethiris Server Obelix is the currently selected Ethiris component.

If the selected component is an Ethiris Server, the configuration will be sent to the online server.

If the selected component is an Ethiris Client, the configuration will instead be saved to the offline configuration file.

File->Save All saves all Ethiris components in the Ethiris Explorer, including the *Ethiris Admin Project* file. As a default, a project is automatically created in Ethiris Admin that keeps track of the comprised Ethiris components. This automatic project does not need to be handled manually but is entirely managed by Ethiris Admin.

File->Recent Projects displays a list of up to 10 Ethiris Admin projects recently used. Selecting a project in the list has the same effect as selecting the *File->Open EthirisComponent->Ethiris Admin Project…* menu item and then browsing for the project configuration file manually.

This is a much more convenient way of opening projects you have recently worked on.

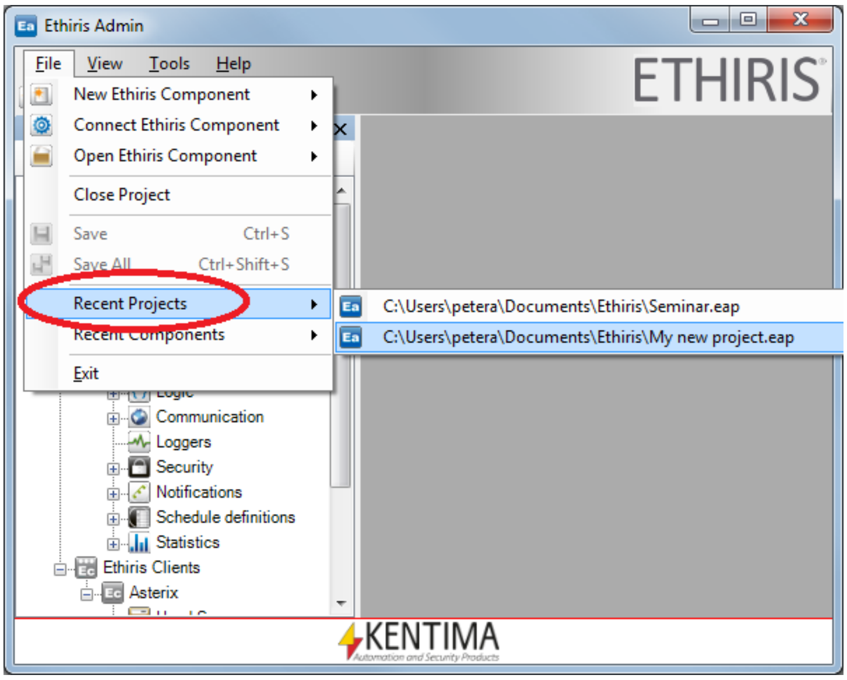


Figure 2.60 Recently used projects menu.

File->Recent Components are similar to the *Recent Projects* menu item. Here the latest *Ethiris Server, Ethiris Cluster* and *Ethiris Client* configurations you have worked on are listed.

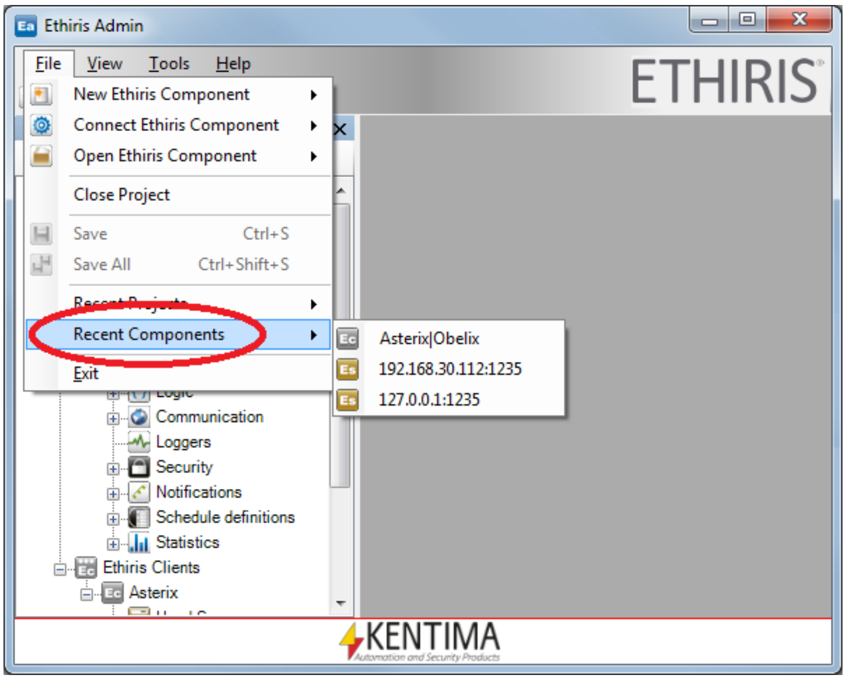


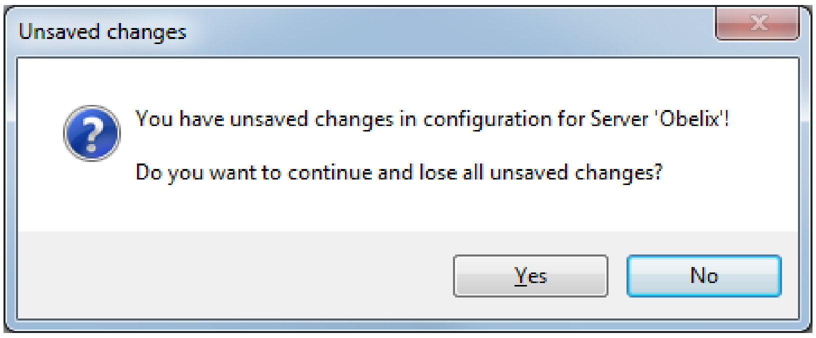
Figure 2.61 Recently used components menu.

If you select an Ethiris Server component it is the same as you selected the *File->Connect Ethiris Component->Ethiris Server…* menu item and then selected the corresponding online server.

If you select an Ethiris Client component in the list, it is the same as if you had selected the *File->Open Ethiris Component->Ethiris Client…* menu item and then selected the corresponding client configuration file.

When you select a component from this list, the component is added to the current project, i.e., it appears in Ethiris Explorer together with the other components.

File->Exit closes Ethiris Admin. If there are any unsaved changes in the configuration, a dialog appears where you can choose to continue closing the application, or you can change your mind (click *No*) and get a chance to save the changes.


Unsaved changes

You have unsaved changes in configuration for Server 'Obelix'!

Do you want to continue and lose all unsaved changes?

Figure 2.62 Dialog notifying you that there are unsaved changes.

2.2.3 View menu

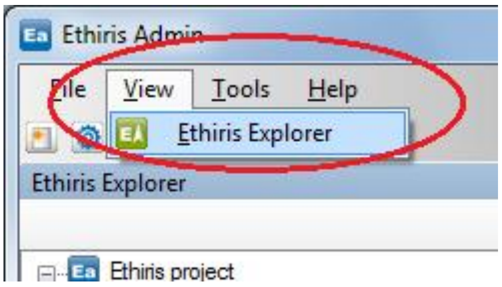


Figure 2.63 The View menu in Ethiris Admin.

The View menu is entirely for opening various windows. If the selected window already is open, it receives focus, and if it is part of a tab group, the corresponding tab is selected so that the window becomes visible.

Ethiris Explorer opens the *Ethiris Explorer* tool window.

2.2.4 Tools menu

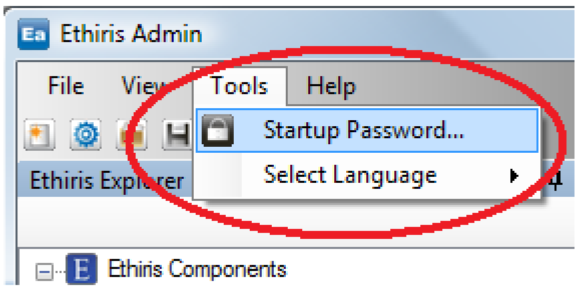


Figure 2.64 The Tools menu in Ethiris Admin.

Startup Password... opens a dialog where you can enter a password that is required for starting Ethiris Admin.

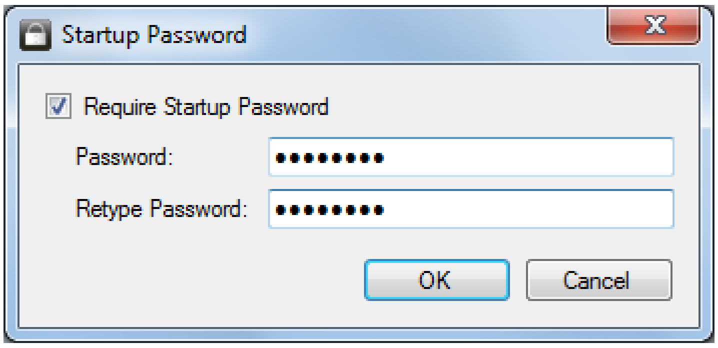


Figure 2.65 The Startup password dialog.

Check the checkbox *Require Startup Password* and enter a *password* if you want to protect Ethiris Admin from unauthorized access.

When you start Ethiris Admin next time, an *Enter password* dialog appears, such as the one in *Figure 2.66*.

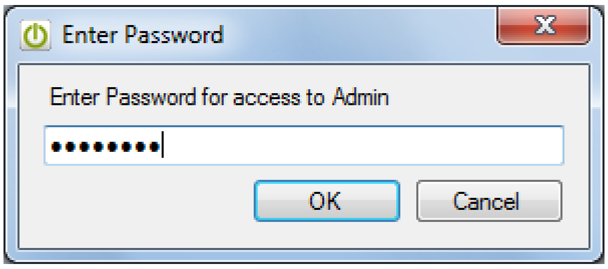


Figure 2.66 The Startup password dialog.

If you don't want a password to be required any longer, simply uncheck the *Require Startup Password* checkbox again.

Select Language displays a sub-menu with available languages.

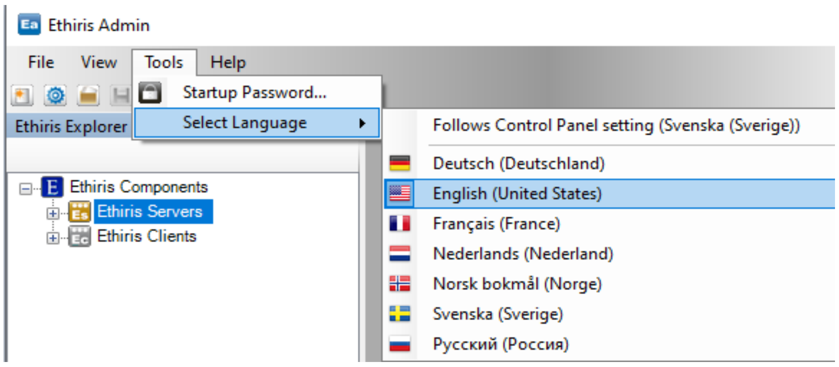


Figure 2.67 Submenu for selecting the current language.

2.2.5 Help menu

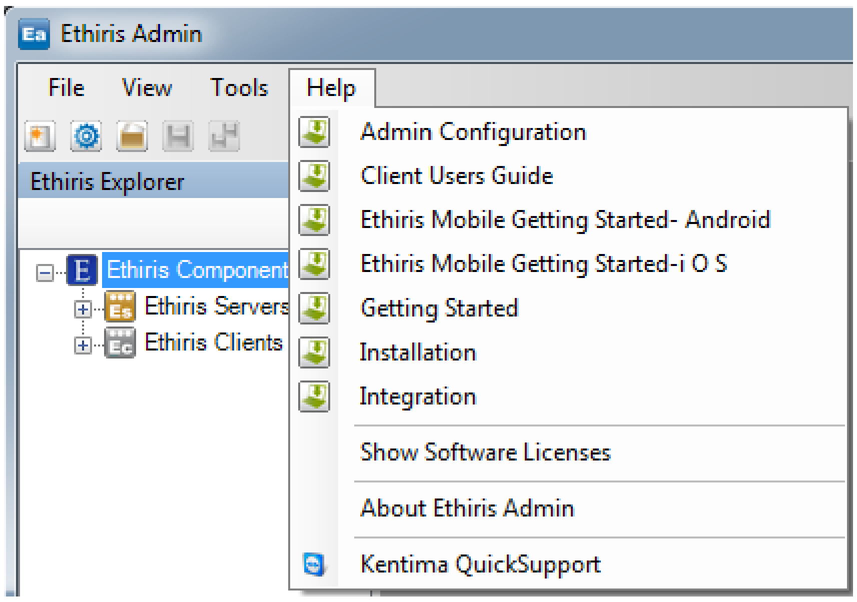


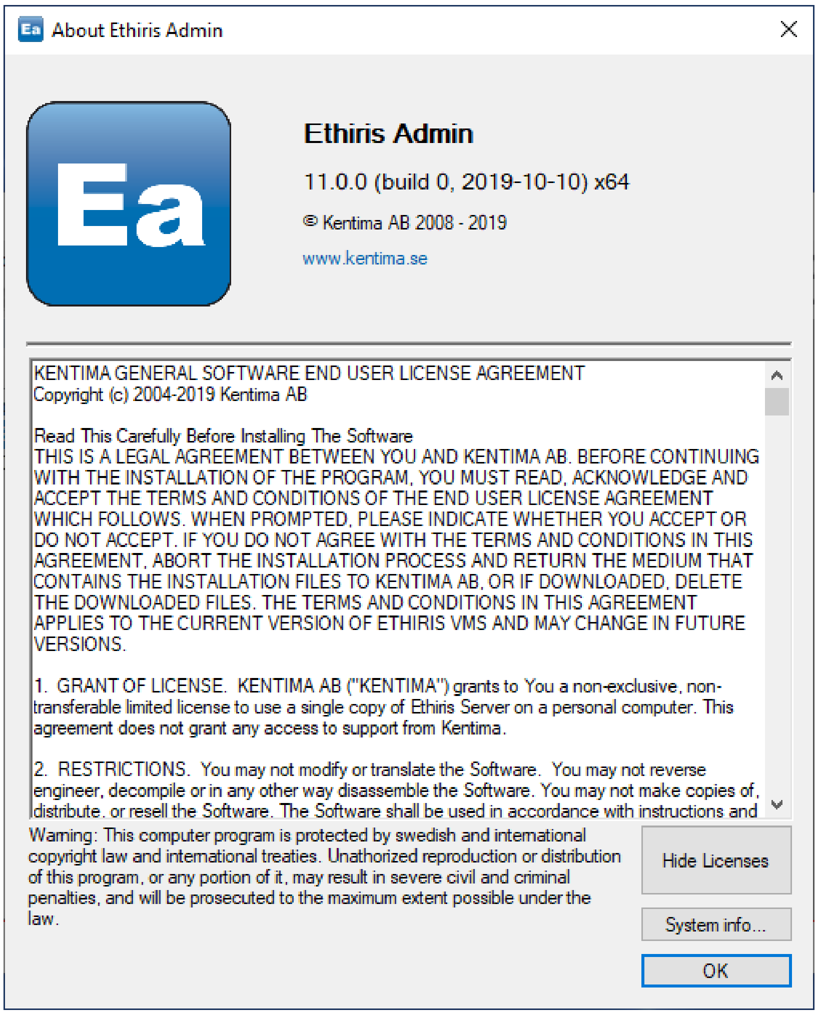
Figure 2.68 The Help menu in Ethiris Admin.

The content of the help menu depends on whether you have opted for installing the manuals or not.

The seven first menu items in the example above are used for opening the pdf version of the corresponding manual. To open them, you need to have a PDF reader installed on the computer, e.g., Adobe Reader.

The next two menu items display the *About* dialog, and the last menu item, *Kentima QuickSupport,* opens a Teamviewer QuickSupport dialog.

Show Software Licenses opens the *About* dialog with the alternative *Show Software Licenses* activated. The dialog displays the Ethiris license agreement and lists the third-party products Ethiris is using.



KENTIMA GENERAL SOFTWARE END USER LICENSE AGREEMENT
Copyright (c) 2004-2019 Kentima AB

Read This Carefully Before Installing The Software
THIS IS A LEGAL AGREEMENT BETWEEN YOU AND KENTIMA AB. BEFORE CONTINUING WITH THE INSTALLATION OF THE PROGRAM, YOU MUST READ, ACKNOWLEDGE AND ACCEPT THE TERMS AND CONDITIONS OF THE END USER LICENSE AGREEMENT WHICH FOLLOWS. WHEN PROMPTED, PLEASE INDICATE WHETHER YOU ACCEPT OR DO NOT ACCEPT. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS IN THIS AGREEMENT, ABORT THE INSTALLATION PROCESS AND RETURN THE MEDIUM THAT CONTAINS THE INSTALLATION FILES TO KENTIMA AB, OR IF DOWNLOADED, DELETE THE DOWNLOADED FILES. THE TERMS AND CONDITIONS IN THIS AGREEMENT APPLIES TO THE CURRENT VERSION OF ETHIRIS VMS AND MAY CHANGE IN FUTURE VERSIONS.

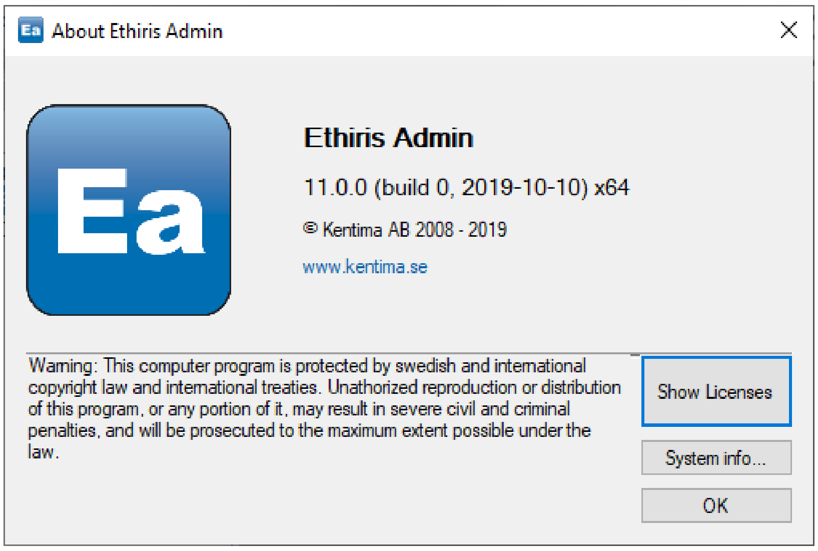
1. GRANT OF LICENSE. KENTIMA AB ("KENTIMA") grants to You a non-exclusive, non-transferable limited license to use a single copy of Ethiris Server on a personal computer. This agreement does not grant any access to support from Kentima.

2. RESTRICTIONS. You may not modify or translate the Software. You may not reverse engineer, decompile or in any other way disassemble the Software. You may not make copies of, distribute, or resell the Software. The Software shall be used in accordance with instructions and

Warning: This computer program is protected by swedish and international copyright law and international treaties. Unathorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 2.69 The About dialog with Show Licenses open.

About Ethiris Admin opens the *About* dialog.



Warning: This computer program is protected by swedish and international copyright law and international treaties. Unathorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure 2.70 The About dialog.

The *About* dialog displays information on the *version number* of the current Ethiris installation.

2.3 Main toolbar

2.3.1 Overview

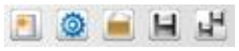


Figure 2.71 The main toolbar in Ethiris Admin.

In this section, we will have a quick look at the main toolbar. In practice, we are already done with this since the buttons in the toolbar are only shortcuts to various menu items. All four toolbar buttons can also be found in the File menu. For a more detailed description of the various functions, please read more in the section *File menu* on *page 2:28*.

New offline component.	Use this button to create a new Ethiris component. This is the same as *File->New Ethiris Component*. A new dialog is opened where you can choose between *Ethiris Admin Project* and *Ethiris Client*.
Connect online to Ethiris Component	Use this button to connect online to an Ethiris component. This is the same as *File->Connect Ethiris Component*. The *Connect to Ethiris Component* dialog is opened, and *Ethiris Server* is the only alternative.
Open Ethiris Component	Use this button to open an existing offline configuration file. This is the same as *File->Open Ethiris Component*. The *Open Ethiris Component* dialog is opened where you can choose between *Ethiris Admin Project* and *Ethiris Client*.
Save selected component	Saves the currently selected Ethiris Component. This is the same as *File->Save*.
Save All components	Saves all Ethiris Components in the Ethiris Explorer. This is the same as selecting the *File->Save All* menu item.

2.4 Ethiris Explorer

2.4.1 Overview

The Ethiris Explorer tool window is by default docked and pinned to the left edge of the main frame. This window contains a treeview of the currently loaded *Ethiris Admin Project*. The project, in turn, is comprised of *Ethiris Components* such as *Ethiris Server* and *Ethiris Client*.

The whole configuration currently loaded is stored internally in Ethiris Admin. Changes made to the configuration are kept in memory until saved. Panels used for configuration changes can be closed without losing the changes. Even the *Ethiris Explorer* itself can be closed without any changes are lost. If you close the whole Ethiris Admin application with unsaved changes, you will be prompted about this and get the chance to save the changes before closing the application.

This section of the manual will describe each type of node that can exist in the Ethiris Explorer treeview.

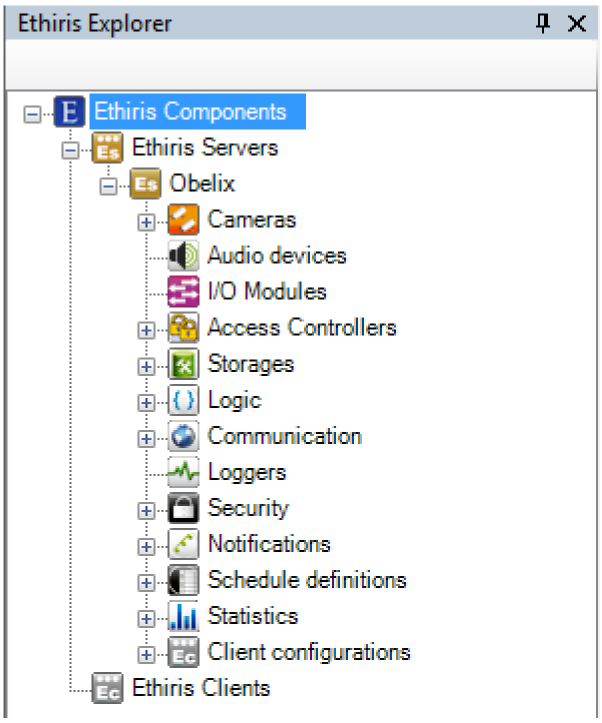


Figure 2.72 The Ethiris Explorer in Ethiris Admin.

The actual content in Ethiris Explorer depends on the current components and their current configuration. Above is an example with one Ethiris Server component and one Ethiris Client component.

2.4.2 Ethiris components node

The *Ethiris components node* is always at the top in the treeview. It represents the whole project.

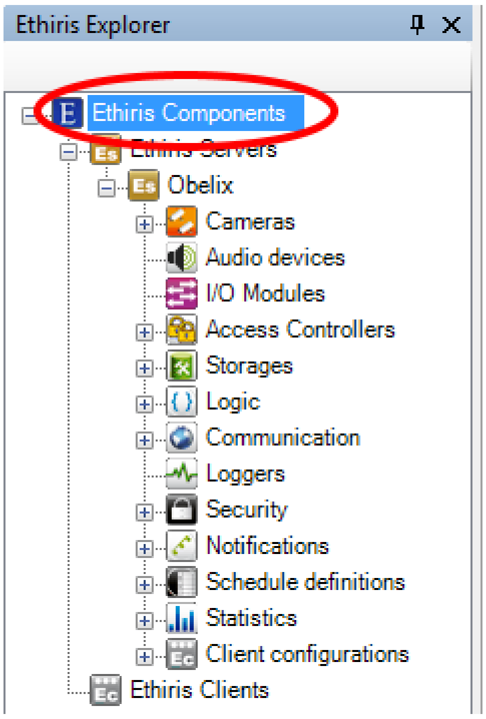


Figure 2.73 The Ethiris components node in Ethiris Explorer treeview.

Ethiris components popup menu

Right-clicking this node brings up a context menu.

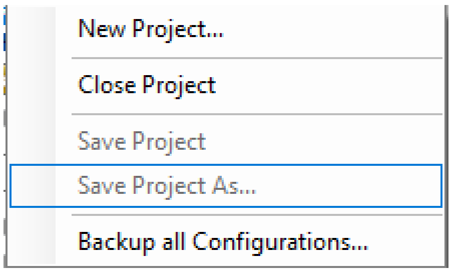


Figure 2.74 The popup menu for the Ethiris components node.

New Project... brings up the *New Project* dialog where you can enter a project name.

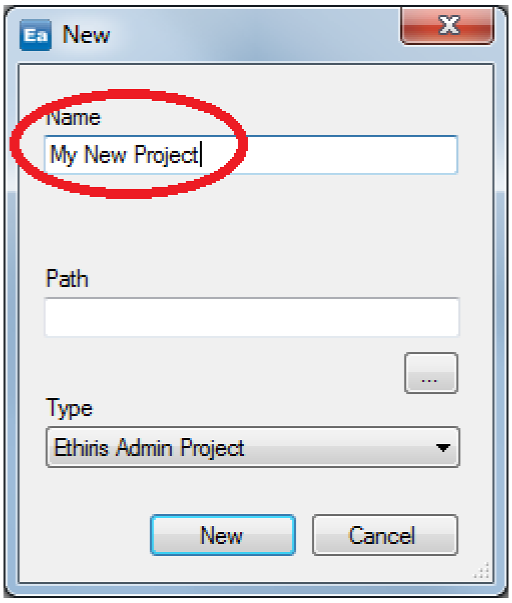


Figure 2.75 Enter a name for your new project.

Click *New* to create a new project with the current Ethiris components. In *Figure 2.76,* you see that the project content remains the same; only the project name is changed.

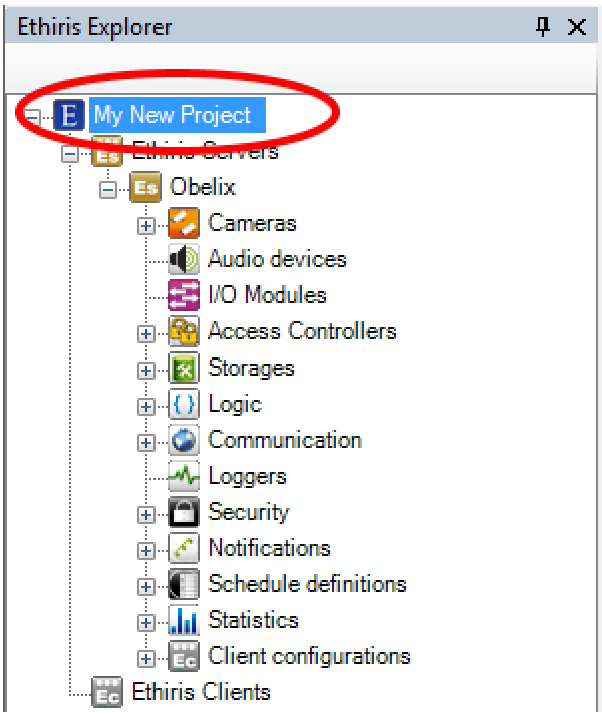


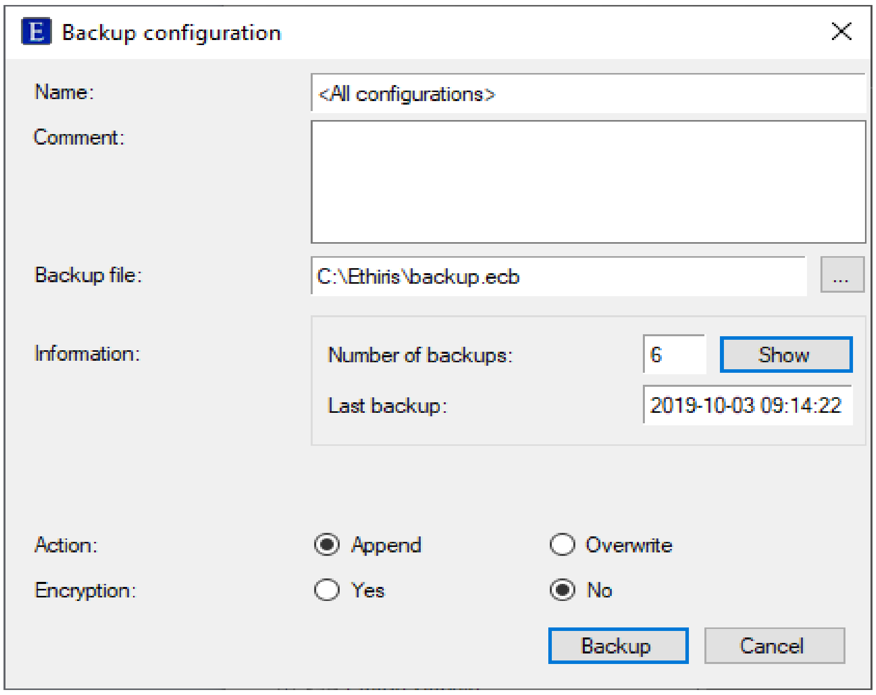
Figure 2.76 After creating a new project the project name is updated.

Close Project closes the current project and removes all Ethiris components from the Ethiris Explorer treeview.

Save Project saves the current project.

Save Project As… saves the current project to a new file.

Backup all Configurations… opens a dialog for backup of those configurations that are loaded into Ethiris Admin.



Backup configuration

Name:	<All configurations>
Comment:	
Backup file:	C:\Ethiris\backup.ecb
Number of backups:	6
Last backup:	2019-10-03 09:14:22
Action:	[✓ Append] Overwrite
Encryption:	Yes [✓ No]

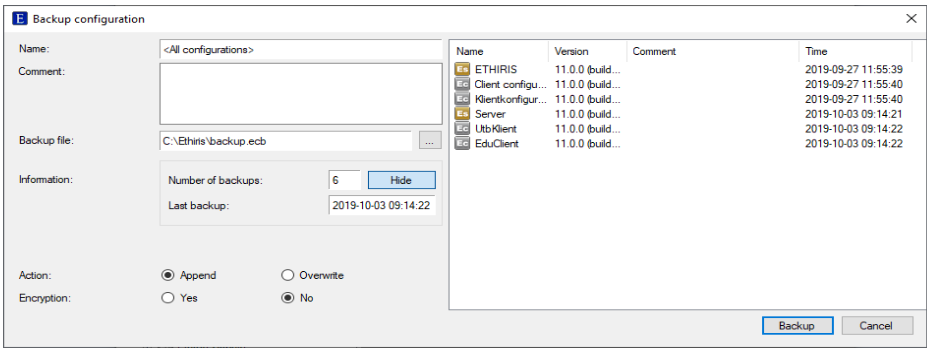
Figure 2.77 Dialog for backup of all configurations.

Name is read-only and is used for information about which configuration(s) to backup.

Comment is optional. The comment is displayed in the list of earlier backups in the backup file. This information helps determine which backup to use in case of a restore.

Backup file is the name and path to the current backup file. The first time you need to select a backup file. The following times you do a backup, Ethiris remembers which file you used last time and will use that as default.

Information is only interesting when a backup has already been done for the current backup file. Click the *Show* button to display a list of backups in the file.



Backup configuration

Name:	<All configurations>
Comment:	
Backup file:	C:\Ethiris\backup.ecb
Number of backups:	6
Last backup:	2019-10-03 09:14:22
Action:	[✓ Append] Overwrite
Encryption:	Yes [✓ No]

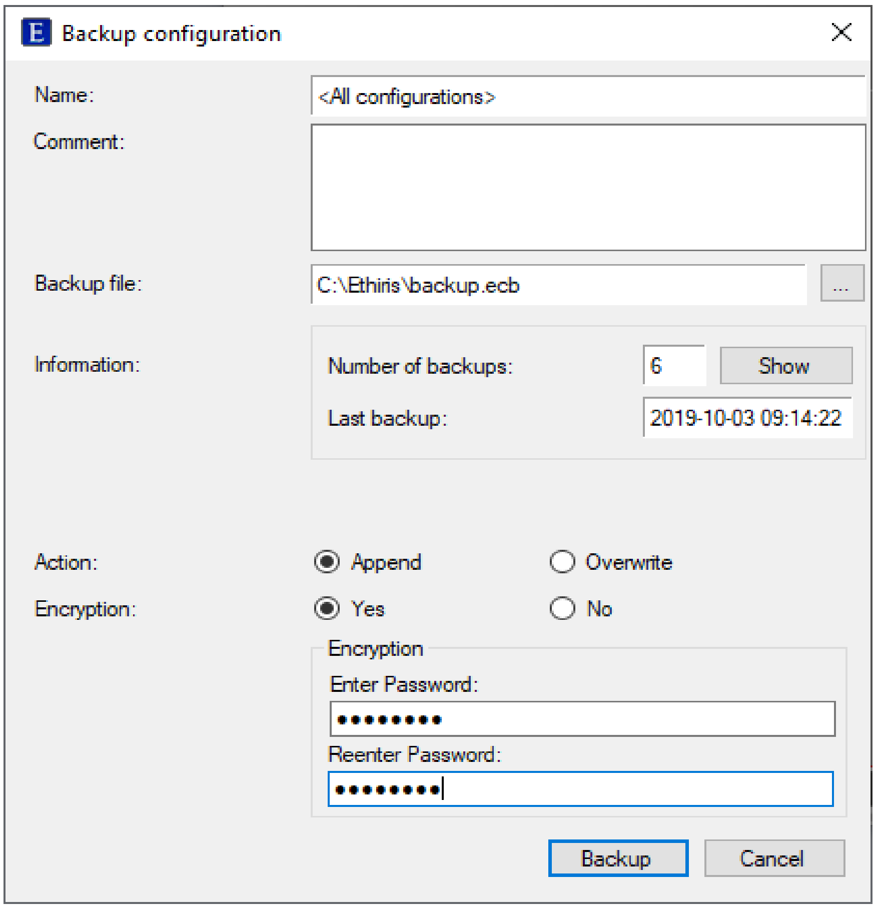
Name	Version	Comment	Time
ETHIRIS	11.0.0 (build...		2019-09-27 11:55:39
Client configu...	11.0.0 (build...		2019-09-27 11:55:40
Klientkonfigur...	11.0.0 (build...		2019-09-27 11:55:40
Server	11.0.0 (build...		2019-10-03 09:14:21
UtbKlient	11.0.0 (build...		2019-10-03 09:14:22
EduClient	11.0.0 (build...		2019-10-03 09:14:22

Figure 2.78 List of earlier backups displayed.

In this example, we can see that the current backup file contains two backups; one for an Ethiris Server and one for an Ethiris Client. You can also see when the last backup was performed.

Action determines whether you *append* the new backup to the existing ones or if you *overwrite* existing backups with the new one.

Encryption determines whether the backup will be encrypted or not. If you choose to encrypt the backup, you have to enter a password which you will have to enter on restoring the backup at a later time.



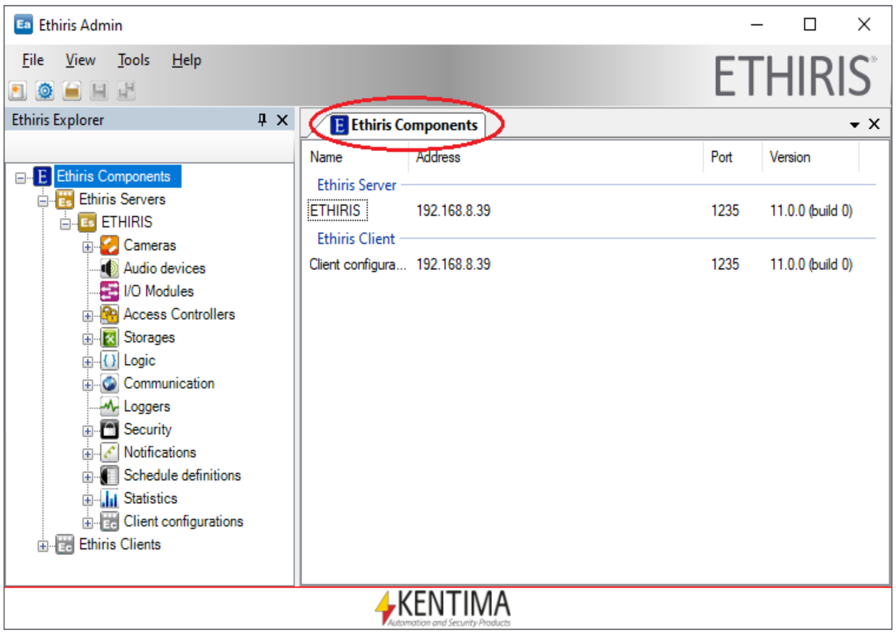
Backup configuration

Name:	<All configurations>
Comment:	
Backup file:	C:\Ethiris\backup.ecb
Number of backups:	6
Last backup:	2019-10-03 09:14:22
Action:	[✓ Append] Overwrite
Encryption:	[✓ Yes] No
Enter Password:	••••••••
Reenter Password:	••••••••

Figure 2.79 Encryption selected.

Ethiris components panel

Double-clicking the *Ethiris Components* node in the treeview opens the corresponding panel.



Name	Address	Port	Version
ETHIRIS	192.168.8.39	1235	11.0.0 (build 0)
Client configura...	192.168.8.39	1235	11.0.0 (build 0)

Figure 2.80 The Ethiris Components panel.

Here you find information about the Ethiris components that are part of the project.

Name is the name you gave the component when adding it to the project.

Address is the IP-address or DNS name for *online* components, and for *offline* components, the address is the path and name of the corresponding configuration file.

Port is only used for online components. It is the TCP/IP port that is used for connecting to the component.

Version is only available for online components. It is the Ethiris version number of the online component.

2.4.3 Ethiris Servers node

The *Ethiris Servers node* is just a collection node for all connected Ethiris Servers in the current project. There is neither a popup menu nor a panel associated with this node.

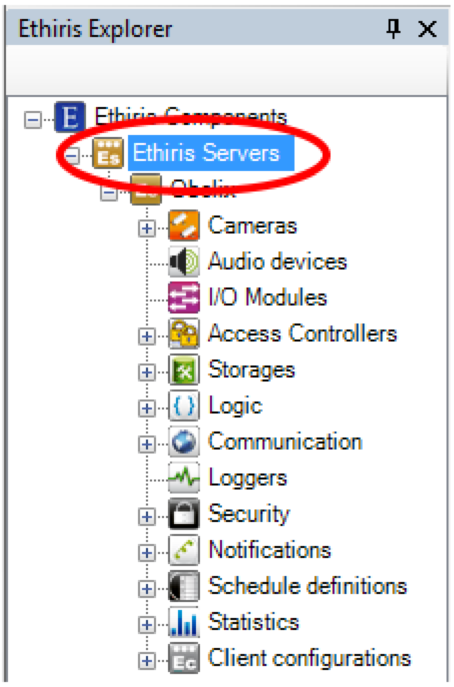


Figure 2.81 The Ethiris Servers node in Ethiris Explorer treeview.

2.4.4 Ethiris Server node

Under the *Ethiris Servers node,* there might be one or more *Ethiris Server* nodes, each one representing an Ethiris Server in the system.

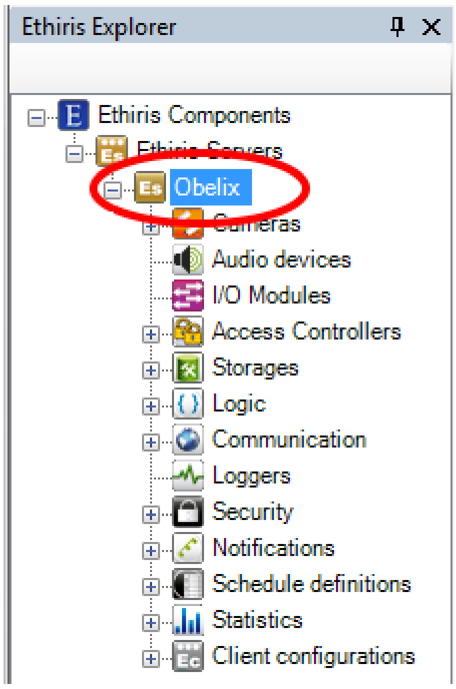


Figure 2.82 The Ethiris Server node in Ethiris Explorer treeview.

Ethiris Server popup menu

Right-clicking this node brings up a context menu.

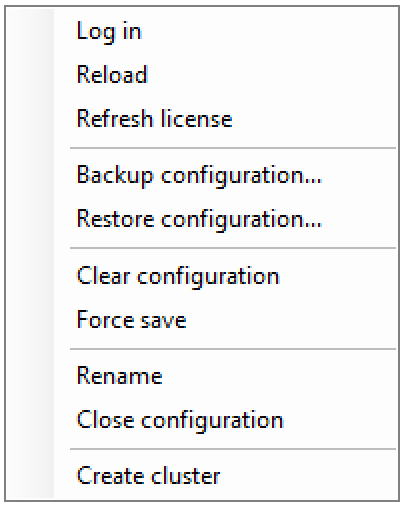


Figure 2.83 The popup menu for the Ethiris Server node.

Log in brings up the *Log in* dialog where you can log in to the Ethiris Server represented of this node in the treeview.

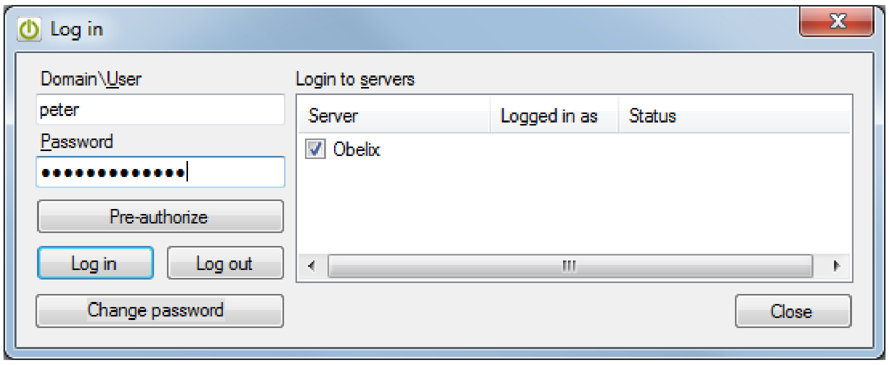


Figure 2.84 Log in dialog.

Enter a user name and password for a Windows user account and click *Log in*. Depending on which user group requirements are set for various Ethiris functions and to which user groups the account is a member, you have access to certain Ethiris functions.

The button *Pre-authorize* is used when double log in is required. In the panel *Security,* you can enter *User Group for Pre-authorization*. When this group is defined, a user belonging to the pre-authorization group has to first log in by the *Pre-authorize* button before a regular user can log in by the *Log in* button.

The button *Change password* is only displayed if you are logged in as an Ethiris user. Click the button to display the dialog *Change password*.

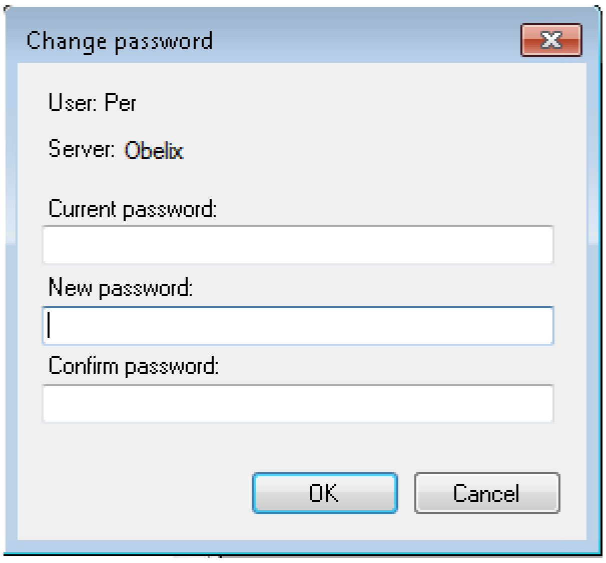
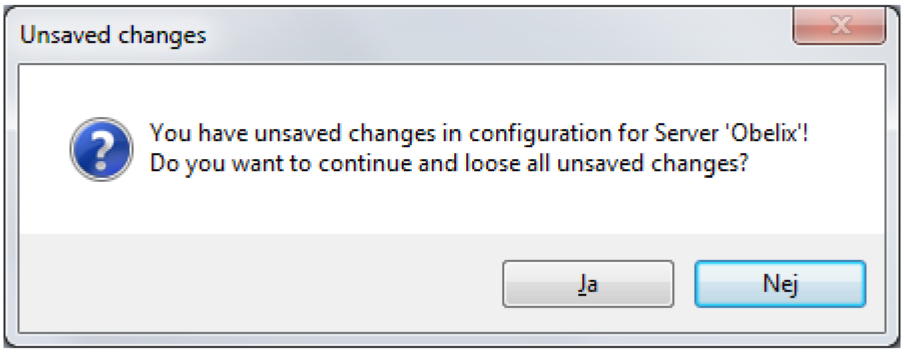


Figure 2.85 Change password dialog.

Enter the current password together with the new password and click OK.

Reload reads the current server configuration online from the Ethiris Server. Any open panels belonging to the server will be closed. If there are unsaved changes in the server configuration, you will be prompted about this.



You have unsaved changes in configuration for Server 'Obelix'!
Do you want to continue and loose all unsaved changes?

Figure 2.86 Unsaved changes dialog.

Click *Yes* for reloading anyway or click *No* to not reload and get the chance to save your changes first.

Refresh license lets Ethiris Server read the current license information. This function can be used if, e.g., you have updated the Ethiris license via the tool *Kentima License Handler*.

Backup configuration… is more or less the same function as described in section *2.4.2 Ethiris components node* on page *2:44*, with the small difference that in this context backup is performed only for the current Ethiris Server. In addition, it is possible to back up all client configurations maintained by this server.

Restore configuration… is used for restoring the Ethiris Server configuration from a backup. Keep in mind that when restoring a Cluster, the Cluster members will be kept as currently defined and not restored. This is so the configuration can be restored on a different cluster or one that has been upgraded with additional members.

Also, client configurations can be restored from the backup. Any client configuration that is restored this way will be maintained by this server. Note that restoring a client configuration this way will actually create a copy of the client configuration. To restore an existing client configuration to a previous state, select *Restore configuration…* from the popup menu on the client configuration node.

Clear configuration Empties the configuration and undoes all changes. If the path d:\EthirisStorage exists, that path will be used by Ethiris Server for storing video and data files.

Force save Forces a complete save and reload of the server configuration, thus bypassing any possible online configuration update. If applied to a cluster, all members will save and reload their configurations.

Rename sets the node in the treeview in change mode. You can enter a new name directly in the treeview.

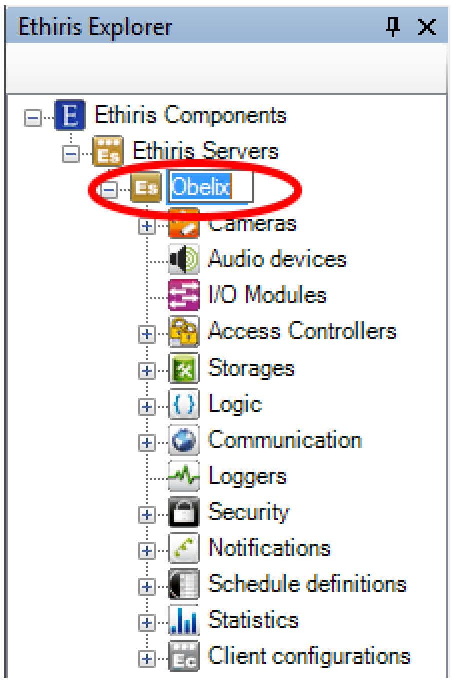


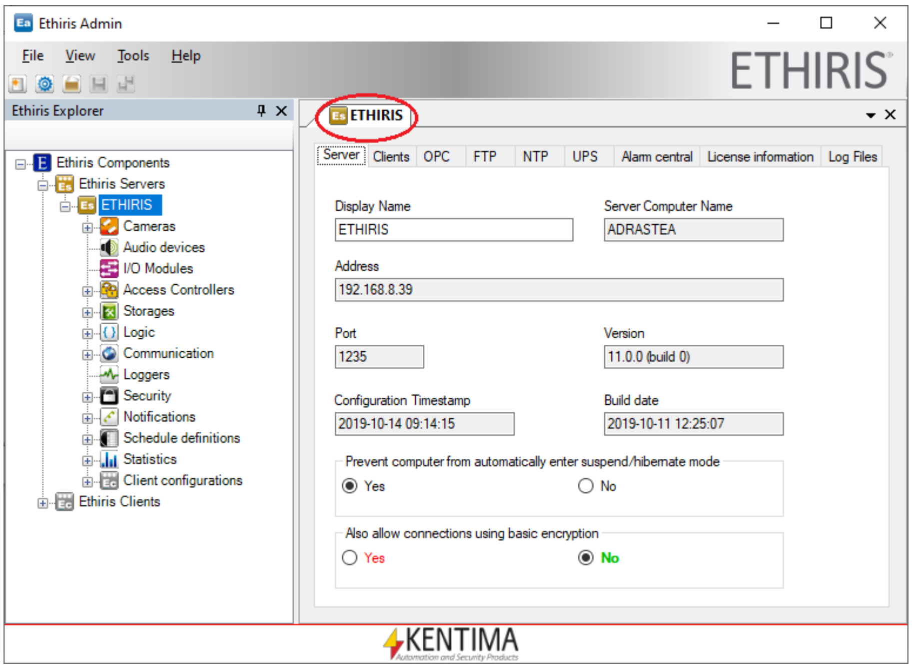
Figure 2.87 Rename server.

Close configuration removes the Ethiris Server from the current project. Note that the configuration in Ethiris Server remains unchanged. It is only the current project in Ethiris Admin that will be affected.

Create cluster turns the configuration into a cluster, see section *2.4.5 Cluster Members node* on page *2:67* for more information.

Ethiris Server panel

Double-clicking the *Ethiris Server* node in the treeview opens the corresponding panel.



Display Name: ETHIRIS
Server Computer Name: ADRASTEA
Address: 192.168.8.39
Port: 1235
Version: 11.0.0 (build 0)
Configuration Timestamp: 2019-10-14 09:14:15
Build date: 2019-10-11 12:25:07
Prevent computer from automatically enter suspend/hibernate mode: [✓ Yes] No
Also allow connections using basic encryption: Yes [✓ No]

Figure 2.88 The Ethiris Server panel.

The *Ethiris Server* panel consists of eight tabs; *Server*, *Clients*, *OPC*, *FTP*, *NTP*, *Alarm central, SIA Server, License information,* and *Log Files*.

Server

Here you find information about the Ethiris Server itself.

Display Name is the name you gave the Ethiris Server component when adding it to the project. It can be changed. As you change the name, it is immediately updated in the treeview. This name is only used for displaying an appropriate name for the server in the treeview.

Server Computer Name is the actual name of the computer where the Ethiris Server runs. This field is read-only.

Address is the IP-address of the computer running Ethiris Server. This field is read-only. The IP-address is established when you connect to the Ethiris Server the first time in your project.

Port is the TCP/IP port that is used for connecting to the Ethiris Server. This field is read-only. The port is established when you connect to the Ethiris Server the first time in your project.

Version displays the current version of the Ethiris Server.

Configuration Timestamp indicates when the server configuration was last saved. This field is read-only.

Build date indicates when the executable file was built (compiled) by Kentima. It is used when Kentima support needs to test something with the exact same version of the Ethiris Server program.

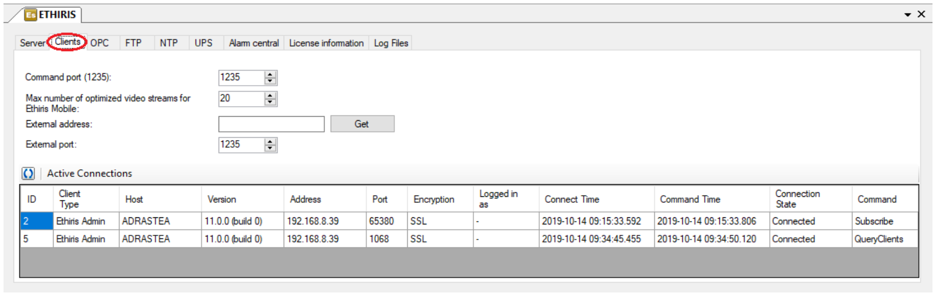
Prevent computer from automatically enter suspend/hibernate mode specifies whether Ethiris Server should allow the computer to enter sleep/hibernate mode. This should normally not be allowed when Ethiris Server is running.

Also, allow connections using basic encryption specifies whether Ethiris Server should allow older clients (before 9.0) to connect. These clients support only basic encryption and hence can potentially make your system less secure. You should only allow this if you are using older clients, and you don't have the possibility to upgrade the right now. When you select *Yes*, you will get a warning about this that you have to accept. Also, each time you save the server configuration, you will get a warning that the server accepts connections using only basic encryption.

When upgrading from a version below 9.0, the server will automatically allow connections using only basic encryption. This is to let existing clients connect to the server. When you have upgraded all clients, turn off this to make your system as secure as possible.

Clients

In this tab is information about how Ethiris Clients communicate with this Ethiris Server.



Command port (1235):	1235
Max number of optimized video streams for Ethiris Mobile:	20
External address:	
External port:	1235

Active Connections

ID	Client Type	Host	Version	Address	Port	Encryption	Logged in as	Connect Time	Command Time	Connection State	Command
2	Ethiris Admin	ADRASTEA	11.0.0 (build 0)	192.168.8.39	65380	SSL	-	2019-10-14 09:15:33.592	2019-10-14 09:15:33.806	Connected	Subscribe
5	Ethiris Admin	ADRASTEA	11.0.0 (build 0)	192.168.8.39	1068	SSL	-	2019-10-14 09:34:45.455	2019-10-14 09:34:50.120	Connected	QueryClients

Figure 2.89 The Clients tab in the Ethiris Server panel.

Command Port is the TCP/IP port that Ethiris Server listens to for incoming commands from Ethiris Clients. As a default, this is 1235. There is very seldom any reason to change this.

Max number of optimized video streams for Ethiris Mobile determines the maximum number of concurrent so-called optimized video streams to one or several Ethiris Mobile clients.

In Ethiris Mobile, there is a setting called *Custom Video stream* that is checked by default. When this option is selected, Ethiris Mobile always requests an optimized video stream from Ethiris Server. If there are any available optimized streams, everything is fine. Otherwise, Ethiris Server sends the images as they come directly from the camera. As long as the video stream is MJPEG, it works anyway in Ethiris Mobile even though the performance is slightly worse.

An optimized video stream means that Ethiris Server scales the image to precisely the size that Ethiris Mobile requires. This results in performance advantages for the mobile unit both when it comes to sending images over, e.g., the 4G network, and when it comes to handling the image in the mobile unit. The drawback is that there is an extra load on Ethiris Server, hence the possibility to set a limit.

Worth notice is that possible cameras utilizing H.264 encoding can only be displayed in Ethiris Mobile if the images are fetched via an optimized video stream. In other words, make sure to set a limit high enough to cover the need for simultaneous viewing of all the H.264 cameras connected to the server.

This limit only affects live images in Ethiris Mobile. Recorded video is always fetched via an optimized video stream.

Max GPU utilization for transcoding (%) determines how much of the GPU in percent that may be used by Ethiris Server when transcoding video streams for Ethiris Mobile. Requires license level Advanced or higher.

External address is the external address of the server, i.e., the public IP-address of the site where Ethiris Server is run. If you want to access Ethiris Server from outside of your local network, you need to access it using this address. By clicking the *Get* button, Ethiris tries to retrieve the external address automatically. If that doesn’t work, enter it manually. If you have a dyn DNS address, you can enter it manually. This information is only used when accessing the system from an alarm central.

External port is the external port used to access the server from outside of the local network. This port needs to be configured properly in your router. Read

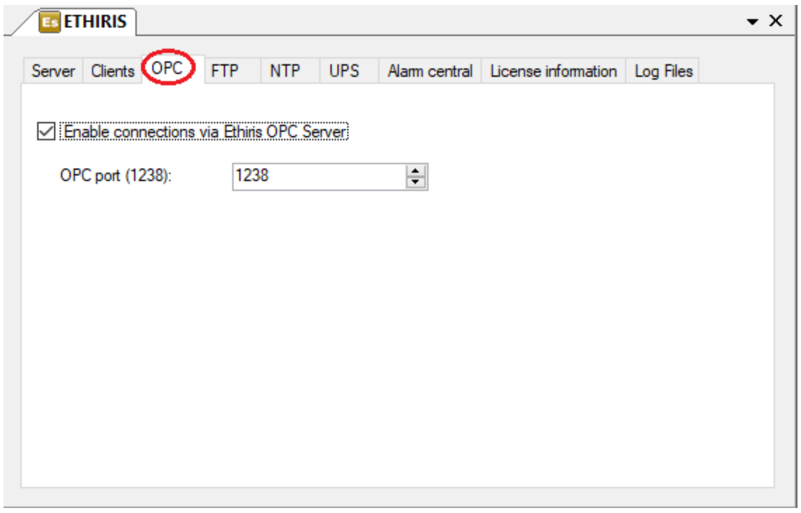
KENTIMA
Automation and Security Products

more about that in your router’s manual. This information is only used when accessing the system from an alarm central.

Active Connections. Click the button to refresh the list with current connections to the Ethiris Server.

OPC

In this tab, you enable OPC Server communication to this Ethiris Server. When enabled, Ethiris Server listens to incoming calls from *Ethiris Server OPC Server*. This communication has to be enabled if other programs should be able to connect to Ethiris Server via an OPC client.



☑ Enable connections via Ethiris OPC Server

OPC port (1238): 1238

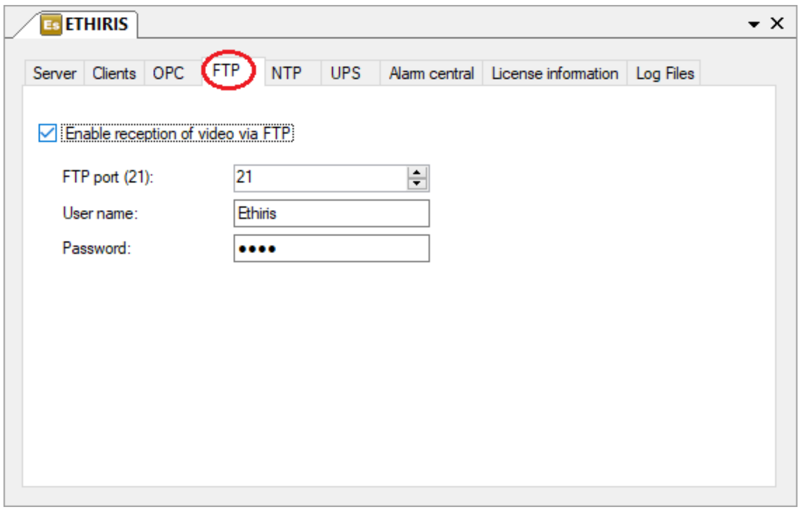
Figure 2.90 The OPC tab in the Ethiris Server panel.

Enable connections via Ethiris OPC Server should be checked to enable OPC communication via Ethiris Server OPC Server.

OPC Port is the TCP/IP port that Ethiris Server listens to for incoming calls from Ethiris Server OPC Server. As a default, this is 1238. There is very seldom any reason to change this.

FTP

In this tab, you can enable FTP communication to this Ethiris Server. When enabled, Ethiris Server listens to incoming FTP requests. The purpose is to receive video from cameras via FTP when an alarm condition is detected by the camera. In this way, it is not necessary to send video from the camera to Ethiris Server unless an alarm occurs. The load of the network is thus decreased. For more information, please see *Task 6* in the *Getting Started with Ethiris* manual.



☑ Enable reception of video via FTP

FTP port (21): 21

User name: Ethiris

Password: ••••

Figure 2.91 The FTP tab in the Ethiris Server panel.

Enable reception of video via FTP should be checked to enable cameras to send video via FTP to Ethiris Server.

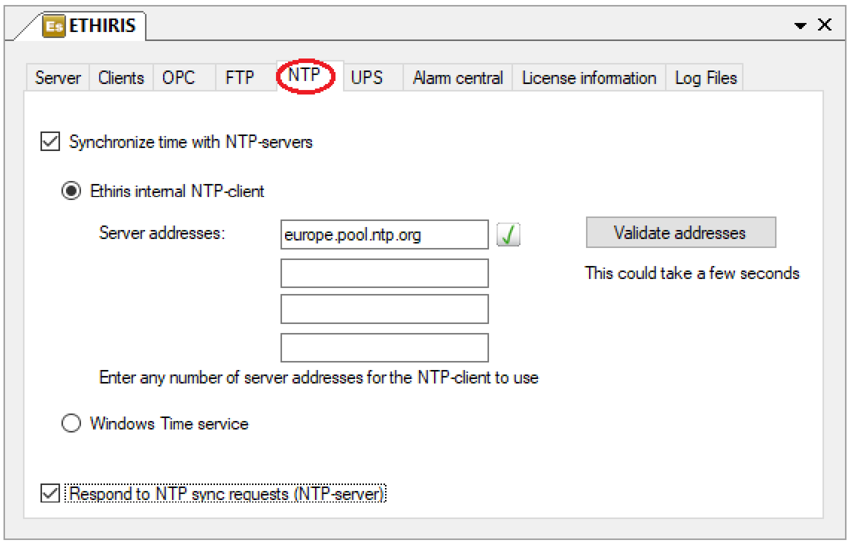
FTP Port is the TCP/IP port that Ethiris Server listens to for incoming FTP requests from cameras/video encoders. As a default, this is 21. There is very seldom any reason to change this.

User name is the user name a camera/video encoder has to use when connecting to Ethiris Server.

Password is the password a camera/video encoder has to use when connecting to Ethiris Server.

NTP

In this tab, you can enable time synchronization with up to four NTP servers. An NTP server keeps time much more accurately than a standard computer and lets you know this time if asked.



☑ Synchronize time with NTP-servers

● Ethiris internal NTP-client

Server addresses: europe.pool.ntp.org ✓

This could take a few seconds

Enter any number of server addresses for the NTP-client to use

○ Windows Time service

☑ Respond to NTP sync requests (NTP-server)

Figure 2.92 The NTP tab in the Ethiris Server panel.

Synchronize time with NTP-servers should be checked for time synchronization to happen at all.

Ethiris internal NTP-client means that the internal NTP-client in Ethiris Server is used. Which servers it should try connecting to can be defined here. Ethiris

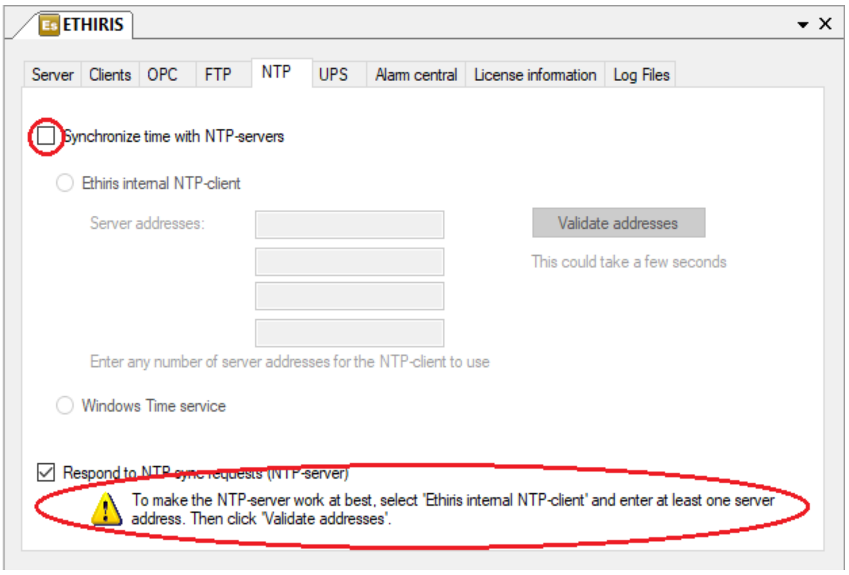
Server will then, with regular intervals, contact all the NTP servers, determine which one it has the best connection to and update the system with the time provided.

Validate addresses checks that the addresses are valid NTP server that Ethiris Server can reach and get an expected response from. If an error occurs, a red exclamation mark will be displayed next to the faulty address. Otherwise, a green tick will indicate a working address.

Windows Time Service means that Ethiris Server will use the built-in time synchronization in Windows. If settings need to be changed, they have to be changed in Windows.

Respond to NTP sync requests (NTP-server) means that Ethiris Server will act as an NTP server on the network. You can use this to synchronize the clocks in your cameras and/or other computers on the network.

To achieve the best functionality, we recommend using the build-in NTP-client in Ethiris to synchronize the clock in the Ethiris Server. Even if you don't have internet access, Ethiris can still synchronize the clocks in other units on the network. Of course, the clock in the computer running Ethiris server will be the reference clock in your network. In this case, a warning will be displayed in the dialog.



☐ Synchronize time with NTP-servers

○ Ethiris internal NTP-client

Server addresses:

This could take a few seconds

Enter any number of server addresses for the NTP-client to use

○ Windows Time service

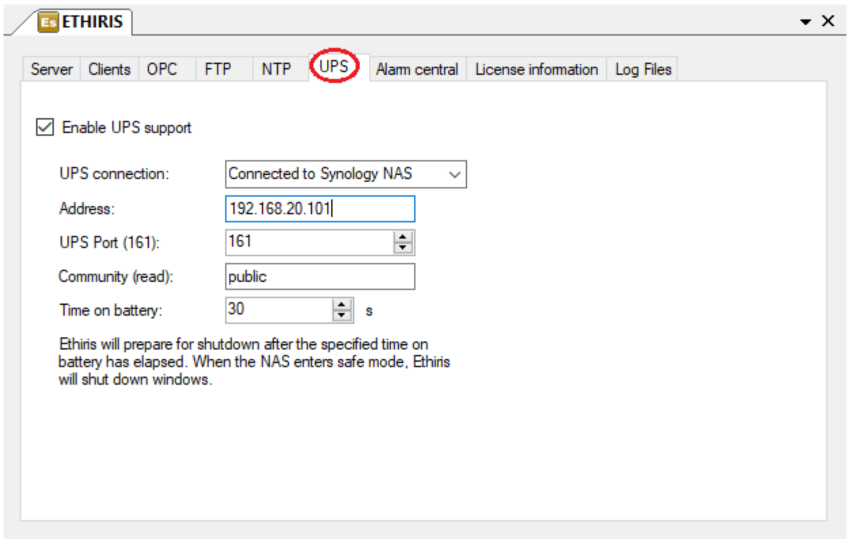
☑ Respond to NTP sync requests (NTP-server)

To make the NTP-server work at best, select 'Ethiris internal NTP-client' and enter at least one server address. Then click 'Validate addresses'.

Figure 2.93 The NTP tab in the Ethiris Server panel.

UPS

In this tab, you can enable UPS support. Which makes it possible for Ethiris to shut down the system in the event of a power outage to prevent the loss of data. Currently Directly to APC Smart-UPS via Ethernet and UPS connected by USB to a Synology NAS is supported.



☑ Enable UPS support

UPS connection:	Connected to Synology NAS
Address:	192.168.20.101
UPS Port (161):	161
Community (read):	public
Time on battery:	30 s

Ethiris will prepare for shutdown after the specified time on battery has elapsed. When the NAS enters safe mode, Ethiris will shut down windows.

Figure 2.94 The NTP tab in the Ethiris Server panel.

Enable UPS Support should be checked to enable communication with a UPS.

UPS connection denotes if communication is made Directly to APC Smart-UPS (ethernet) or via a Synology NAS with UPS connected by USB.

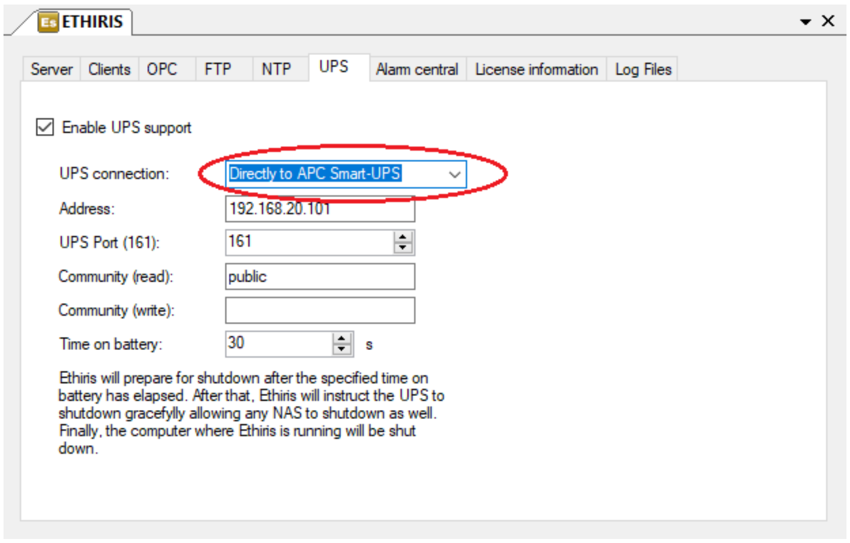
Address is the IP-address of the APC Smart-UPS/Synology NAS on the local network.

UPS port (161) is the TCP/IP port that the communication will use. As a default, this is 161 (SNMP). There is very seldom any reason to change this.

Community (read) is the Community that is needed to be able to read information from the device using SNMPv1. As a default, this is public.

Community (write) is the Community that is needed to be able to write information to the device using SNMPv1.

Time on battery is the time for when Ethiris will begin a graceful shutdown of the UPS to allow any NAS to complete shutdown before complete system shutdown. Afterward, the system will begin its shutdown. As a default, this is 30 seconds and will require to be changed depending on the size of the UPS to be used, and the uptime needed during a power outage before shutdown.



☑ Enable UPS support

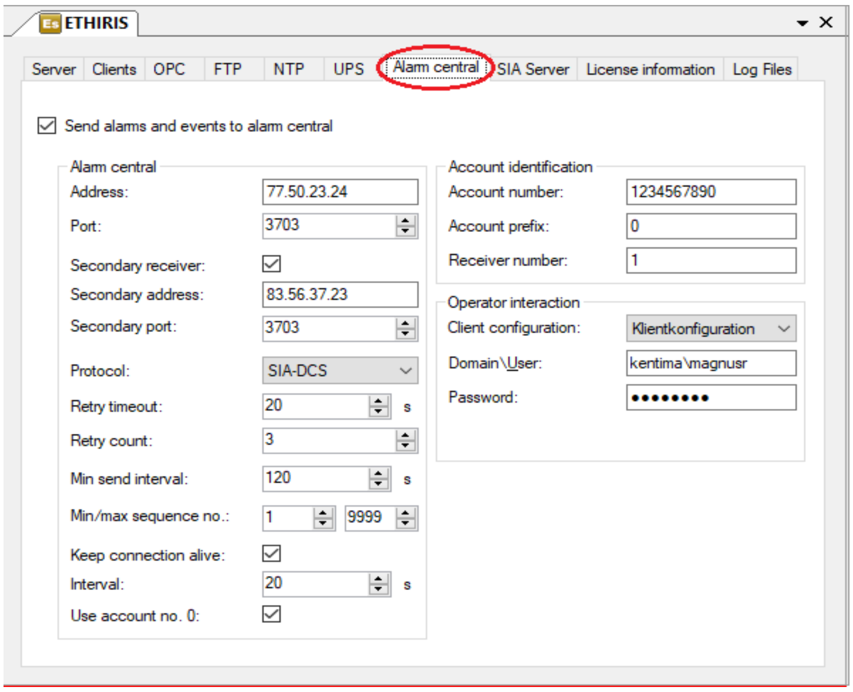
UPS connection:	Directly to APC Smart-UPS
Address:	192.168.20.101
UPS Port (161):	161
Community (read):	public
Community (write):	
Time on battery:	30 s

Ethiris will prepare for shutdown after the specified time on battery has elapsed. After that, Ethiris will instruct the UPS to shutdown gracefylly allowing any NAS to shutdown as well. Finally, the computer where Ethiris is running will be shut down.

Figure 2.95 The NTP tab in the Ethiris Server panel.

Alarm central

In this tab, alarm central settings are displayed to allow Ethiris to send alarm messages to an alarm central. This tab is only visible if you have high enough license level, read more about this in the license levels document for the current version.



☑ Send alarms and events to alarm central

Alarm central

Address:	77.50.23.24
Port:	3703
Secondary receiver:	☑
Secondary address:	83.56.37.23
Secondary port:	3703
Protocol:	SIA-DCS
Retry timeout:	20 s
Retry count:	3
Min send interval:	120 s
Min/max sequence no.:	1 9999
Keep connection alive:	☑
Interval:	20 s
Use account no. 0:	☑

Account identification

Account number:	1234567890
Account prefix:	0
Receiver number:	1

Operator interaction

Client configuration:	Klientkonfiguration
Domain\User:	kentima\magnusr
Password:	••••••••

Figure 2.96 The Alarm central tab in the Ethiris Server panel.

Send alarms and events to alarm central Check this box if you have an agreement with an alarm central that can receive SIA-messages from Ethiris.

Fill out all fields in this tab if you want to use this functionality. You will get information about these settings from the alarm central once you have an agreement with them. More configuration is required in Ethiris to send signals to the alarm central.

Alarm central

Address Enter the IP-address Ethiris should connect to at the alarm central.

Port Enter the port Ethiris should use when connecting to the alarm central.

Secondary receiver Check this box if the alarm central supports a secondary receiver. This means that Ethiris can send alarms to the secondary receiver if the primary receiver doesn’t respond.

Secondary address is only shown if you have checked the box *Secondary receiver*. Enter the IP-address of the secondary receiver.

Secondary port is only shown if you have checked the box *Secondary receiver*. Enter the port used by the secondary receiver.

Protocol for now only *SIA-DCS* is supported, and there is only one alternative in the list.

Retry timeout denotes how long Ethiris will wait for a response from the alarm central on a message sent.

Retry count denotes the number of times Ethiris will repeat the same message if no response is received within the *Retry timeout*.

Min send interval denotes the number of seconds that must pass until the same message can be sent again to the alarm central.

Min/max sequence no. denotes the minimum and maximum values the sequence number will use. Beginning at minimum when the server is started.

Keep connection alive tells if the connection to the alarm central should be kept open at all times. If the connection should go down, the alarm central can start investigating why this has happened.

Interval denotes the interval at which Ethiris will send keep-alive messages to the alarm central if no other messages are sent.

Use account# 0 denotes if the keep-alive messages should contain account number 0 instead of the actual account number set in the field *Account number*.

Account identification

Account number Enter your account number at the alarm central here.

Account prefix Enter account prefix if used by the alarm central. Otherwise, enter 0.

Receiver number Enter receiver number here if used by the alarm central. Otherwise, enter 1.

Operator integration

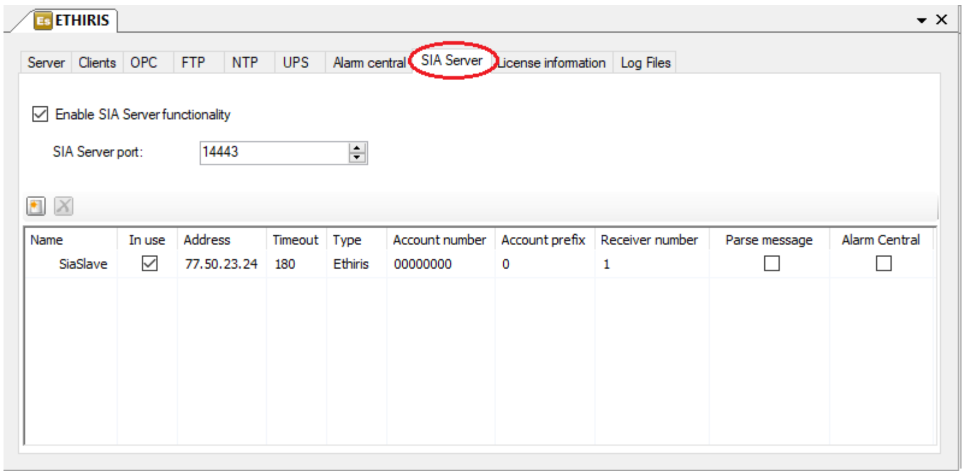
Client configuration If the alarm central has Ethiris Client installed locally and you have an agreement that they should connect to your system when an alarm is sent, enter the client configuration they will use. The client configuration must be handled by this server, and you must have entered the public IP-address and port in the fields *External address* and *External port.* Your router must be configured to allow connections to Ethiris from outside of your local network.

Domain\User Enter username the alarm central should use when connecting to your system. If no login is required, leave the field blank.

Password Enter the password the alarm central should use when connecting to your system. If no login is required, leave the field blank.

SIA Server

In this tab, alarm central settings are displayed to allow Ethiris to receive and relay alarm messages to an alarm central. This tab is only visible if you have the necessary option, read more about this in the license levels document for the current version.



☑ Enable SIA Server functionality

SIA Server port: 14443

Name	In use	Address	Timeout	Type	Account number	Account prefix	Receiver number	Parse message	Alarm Central
SiaSlave	☑	77.50.23.24	180	Ethiris	00000000	0	1	☐	☐

Figure 2.97 The SIA Server tab in the Ethiris Server panel.

Enable SIA Server functionality Check this box to enable the SIA Server to be able to receive SIA messages in Ethiris.

SIA Server port Enter the port the SIA Server should use when receiving SIA messages. By default, this is 14443.

Name is the name of the SIA Slave and must be unique within the configuration. An identical name cannot be entered.

In Use is ticked as default, meaning that the SIA slave is in use, and Ethiris can receive SIA messages from it. If not ticked, Ethiris will ignore messages sent from this SIA Slave.

Address is the IP address for the SIA Slave (or DNS name if this was set up).

Timeout is the max time that may pass without communication from the SIA Slave. After that, a communication error is activated.

Type the alternatives Ethiris and Other is possible. Only used to represent what kind of SIA Slave is configured visually. E.g., Ethiris Server or an alarm.

Account number Enter your account number at the alarm central here. This replaces what is configured for the Alarm Server.

Account prefix Enter account prefix used by the alarm central. This replaces what is configured for the Alarm Server.

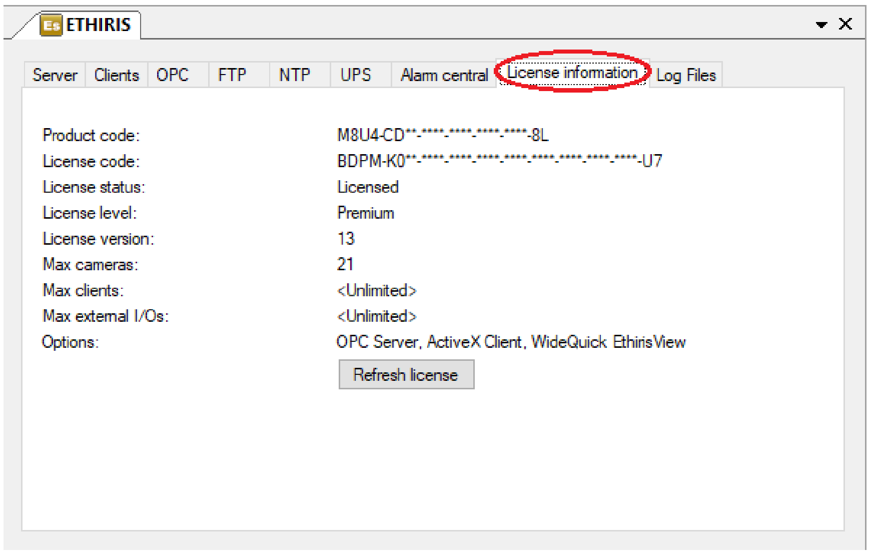
Receiver number Enter receiver number here used by the alarm central. This replaces what is configured for the Alarm Server.

Parse message denotes if Ethiris should parse messages when received. Exposing them for use in scripts. It is, by default, not checked.

Alarm Central is ticked as default, meaning that the messages from this SIA Slave will be relayed to the Alarm Central configured on the Alarm Central tab. Requires that an Alarm Central is configured.

License information

In this tab, license information is displayed.



Product code:	M8U4-CD**-****-****-****-****-8L
License code:	BDPM-K0**-****-****-****-****-****-****-****-****-U7
License status:	Licensed
License level:	Premium
License version:	13
Max cameras:	21
Max clients:	<Unlimited>
Max external I/Os:	<Unlimited>
Options:	OPC Server, ActiveX Client, WideQuick EthirisView

Figure 2.98 The License information tab in the Ethiris Server panel.

Product code displays the product code used on the installation of the Ethiris Server.

License code displays the current license code. The license code may be updated, upgraded, and extended after the initial installation.

License status displays the current status for the license. When everything is OK, the status is *Licensed*.

License version denotes the major version of Ethiris this license applies to.

Max cameras indicate the maximum number of cameras that can be connected to this Ethiris Server.

Max clients indicate the maximum number of clients that can be connected to this Ethiris Server. The number of clients is the sum of simultaneously connected *Ethiris Client*, *Ethiris Mobile*, *Ethiris ActiveX,* and *WideQuick EthirisView*.

Max external I/Os indicates the maximum number of external I/Os that can be connected to this Ethiris Server. It applies to I/Os that are connected via an OPC Server. I/O that are available via cameras and video encoders are always available in Ethiris and are not considered part of the number of external I/Os.

Options indicate any options that are included in the license for this Ethiris Server. These can be *OPC Server*, *ActiveX Client*, *WideQuick EthirisView, Sia Server, Ethiris NVR,* and *Retailer Demo*.

OPC-Server, this option enables the possibility to connect Ethiris Server OPC-Server to this Ethiris Server, i.e., other systems can via an OPC client connect to this Ethiris Server to read/write variables in Ethiris Server's data store. If this option is missing Ethiris Server will not respond to requests from Ethiris Server OPC-Server.

ActiveX Client, this option enables the possibility to connect an Ethiris ActiveX component to this Ethiris Server and receive live video. Ethiris ActiveX component is used for displaying live video in a system that can handle ActiveX components, e.g., an HMI/SCADA system.

WideQuick EthirisView, this option enables the possibility to connect WideQuick built-in Ethiris View objects to this Ethiris Server. Only available for license level *Premium*.

Sia Server, this option enables the ability to receive Alarms from other Ethiris Servers and SIA messages from other devices and relay them to an Alarm Central. Available only for license level *Premium*.

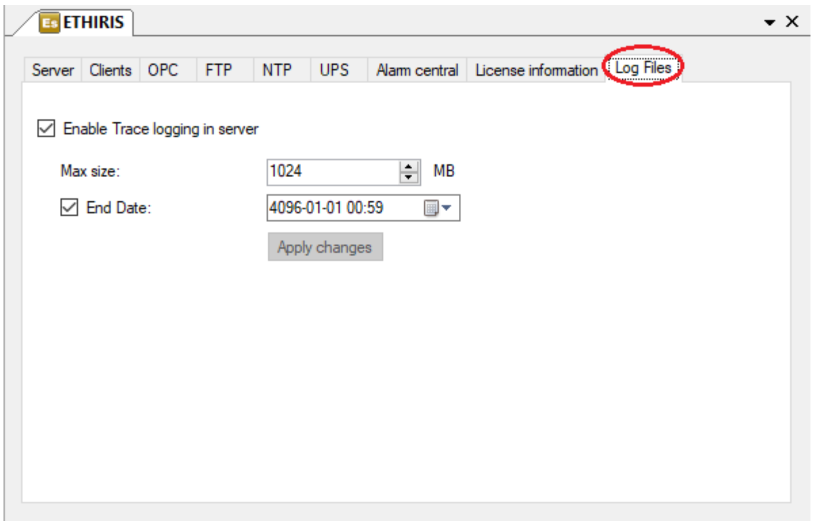
Ethiris NVR, only applies to licenses delivered pre-installed on Ethiris NVR systems.

Retailer Demo, this option denotes that the license is a so-called retailer demo. This type of license is a full license except for that Ethiris Server closes the communication with all cameras after 4 hours.

Refresh license is used if you have upgraded the license and entered a new license code via *Kentima License Handler*. Instead of having to restart Ethiris Server, you can click this button to make Ethiris Server read the license code and adapt to the new license.

Log Files

In this tab, you can choose to activate the so-called trace log.



☑ Enable Trace logging in server

Max size: 1024 MB

☑ End Date: 4096-01-01 00:59

Figure 2.99 The tab Log Files in the Ethiris Server panel.

Enable Trace logging in server should be checked if you want to create a trace log. The file is named *EthirisServerTrace.log* and is located in the Ethiris installation folder. This log file contains detailed information about how Ethiris works and what is happening in Ethiris Server. The information in the file can be of great service to Kentima personnel in a problem-solving situation.

Each time Ethiris Server is restarted, a copy of the old trace log is saved as *EthirisServerTrace1.log*. There can be up to 5 older log files.

Max size determines how large the log file is allowed to get. If the file gets full, the file will be closed, and logging will continue in a new file.

End date determines a point in time when Ethiris Server shall stop logging information to the file.

Apply changes should be clicked when you want to apply your changes for logging.

2.4.5 Cluster Members node

An Ethiris Server can be represented by a cluster. This means that one or more different physical machines can act as one unit with redundancy. If a member is disconnected, the others can automatically handle the member´s tasks with a barely noticeable impact on any potential clients.

To run a cluster, all members included must have a Premium license. The total number of camera licenses in the cluster determines how many of the included members can be disconnected with sustained functionality in the cluster. If the cluster is made up of 4 servers with 10 camera licenses each, the cluster has a total of 40 camera licenses. If the facility has 20 cameras, which means that any two of the servers in the cluster can be disconnected with sustained functionality, by being excessive with the number of camera licenses, one can decide how redundant the system will be.

The Cluster Members node only shows up below an Ethiris Server if a cluster exists. To create a cluster if there is none, right-click on the server node and choose *Create Cluster*. This will create a cluster with the chosen server as its *Master* and only member.

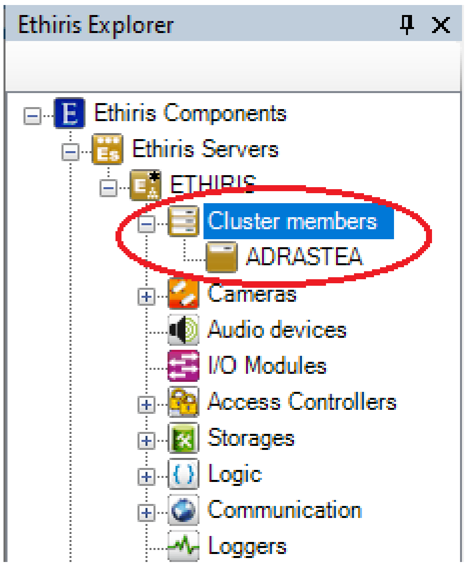


Figure 2.100 The Cluster Members node in the Ethiris Explorer treeview

The Master server is responsible for the functionality that can only run on a single machine, Script for example, and determines the distribution of tasks to the remaining members. There is always a Master server in the cluster; if no one exists or the current one disappears, a new one is appointed automatically. Once the cluster is running functions are configured as usual in Ethiris Admin. The Master server will distribute any changes in the cluster to its members.

Cluster Member Popup Menu

Right-clicking on *Cluster members* node opens a menu.

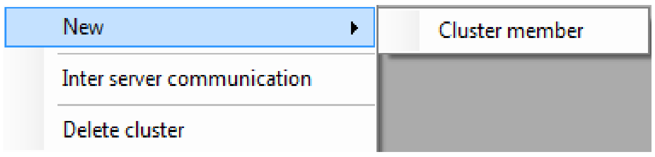
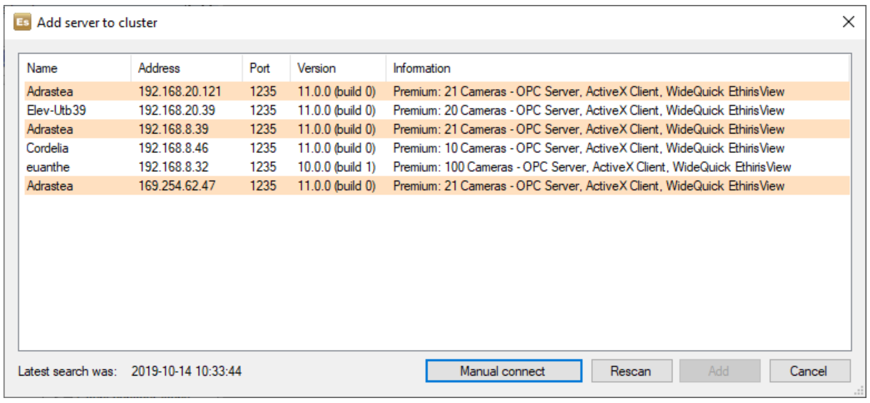


Figure 2.101 The popup menu for the Cluster Members node.

New->Cluster Member starts the process of adding a cluster member to the cluster and will ask for an address to the new member.



Add server to cluster

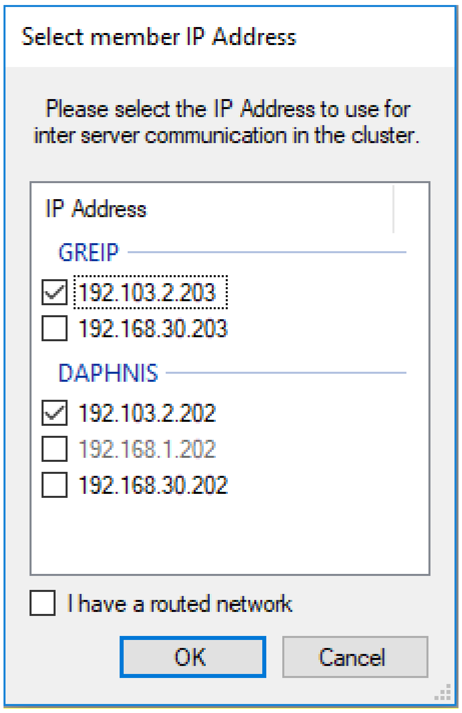
Name	Address	Port	Version	Information
Adrastea	192.168.20.121	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Elev-Utb39	192.168.20.39	1235	11.0.0 (build 0)	Premium: 20 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Adrastea	192.168.8.39	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Cordelia	192.168.8.46	1235	11.0.0 (build 0)	Premium: 10 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
euanthe	192.168.8.32	1235	10.0.0 (build 1)	Premium: 100 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView
Adrastea	169.254.62.47	1235	11.0.0 (build 0)	Premium: 21 Cameras - OPC Server, ActiveX Client, WideQuick EthirisView

Latest search was: 2019-10-14 10:33:44

Figure 2.102 The Add server to cluster Dialog in Ethiris Admin

If the cluster contains members with multiple IP addresses, a dialog will appear and make you choose which IP addresses to use for inter-server communication.

All communication between the members in the cluster should be made on the same subnet. If you have chosen IP addresses for the various servers from different subnets, the *OK* button will be disabled unless you check *I have a routed network*.



Select member IP Address

Please select the IP Address to use for inter server communication in the cluster.

IP Address

GREIP
- [x] 192.103.2.203
- [ ] 192.168.30.203

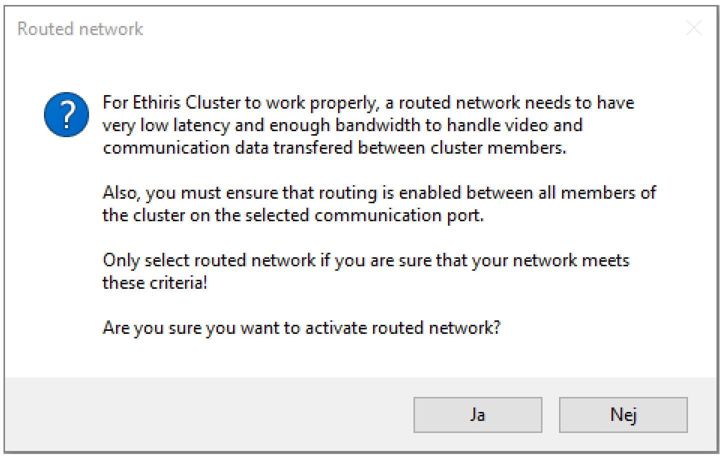
DAPHNIS
- [x] 192.103.2.202
- [ ] 192.168.1.202
- [ ] 192.168.30.202

- [ ] I have a routed network

Figure 2.103 In this dialog, you choose which IP addresses to use.

A routed network means that a router sends traffic between different subnets. If you want to use a routed network, you must ensure that it has very low latency and enough bandwidth to handle video and communication data transferred between cluster members.

When you check *I have a routed network,* the following information will be displayed.


Routed network

For Ethiris Cluster to work properly, a routed network needs to have very low latency and enough bandwidth to handle video and communication data transfered between cluster members.

Also, you must ensure that routing is enabled between all members of the cluster on the selected communication port.

Only select routed network if you are sure that your network meets these criteria!

Are you sure you want to activate routed network?

Figure 2.104 Information shown when you check, I have a routed network.

If the server does not have an empty configuration, it will be cleared so that the server can use the cluster's configuration. If the server is open in Ethiris Admin, it will be closed as it gets added to the cluster instead. The member then shows up in the treeview as a new node in the cluster. If the cluster panel is open, it will show up there as well.

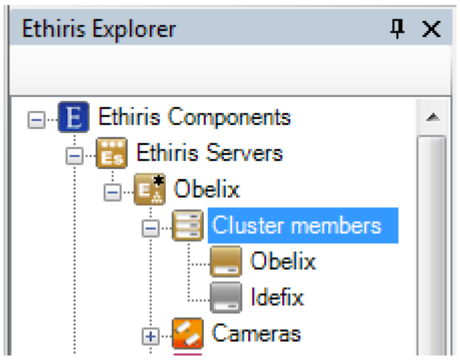


Figure 2.105 A new member has been added to the cluster

Note that the member icon is grey. This is because we have not yet saved the configuration, so it is not yet a part of the cluster.

Make sure to save the Client's configuration whenever you change the structure of the cluster

The client needs to know of all the servers in a cluster so it can connect to them. If the client's configuration is open while something is changed in the structure of the cluster or if it's opened after the changes are made, relevant changes in the client will be made automatically. It has to be saved manually, however.

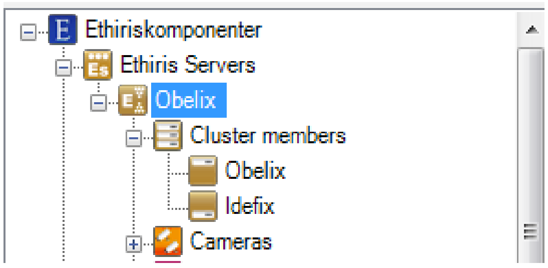


Figure 2.106 The Cluster is now running

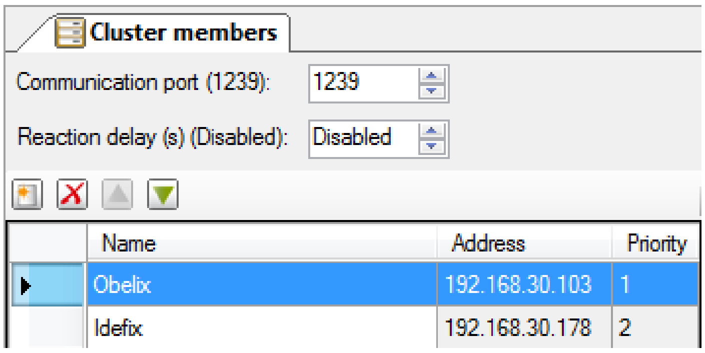
After saving, the cluster will start operating. The icons show that both members are online and running. They also show that Obelix is the Master and that Idefix is a regular member.

Inter server communication will show the dialog for selecting which IP-address(es) to use for communication within the cluster.

Delete cluster will remove the entire cluster. The Master server is turned into a stand-alone server with the cluster's configuration, and other members receive empty configurations.

Cluster Members Panel

Double-clicking on the Cluster Members node in the treeview opens the corresponding panel.



Communication port (1239): 1239

Reaction delay (s) (Disabled): Disabled

	Name	Address	Priority
▶	Obelix	192.168.30.103	1
	Idefix	192.168.30.178	2

Figure 2.107 The Cluster members panel

This panel contains a list of all members currently in the cluster.

Communication *port* determines which TCP port that will be used for communication between the members in the cluster. This port should normally not have to be changed.

Reaction delay (s) determines how long the cluster will wait if a cluster member loses network contact with other members in the cluster due to temporary network drops. The function is used to make the cluster not reconfigure itself at short network drops. Usually, this function is disabled.

At the top of the panel, there is a toolbar.

Cluster members panel toolbar

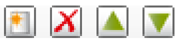


Figure 2.108 The toolbar in the Cluster members panel

Add a new server	Use this button to create a new cluster member. This is the same as *New->Cluster Member* in the popup menu described above. A new cluster member is added to the server configuration.
Delete selected server(s).	Use this button to delete the selected server(s). You can choose more than one server by using the *Ctrl*-button and/or the *Shift*-button. When a member is deleted, it will be converted to an empty stand-alone server.
/ Move up/down	Use these buttons to change the priority of a member. When multiple members in the cluster are online simultaneously the server with the highest priority will take the Master role.

To keep in mind when configuring a cluster

The cluster functionality in Ethiris is designed to be used in a local network. This means that it is not suitable to place the servers that will be part of the same cluster in such a way that communication between them has to go through a connection with lower performance than a modern local network with a Gbit-connection and low latency.

It is also important that all servers in the cluster can reach each other, so there can be no limitation of this aspect in the network. Make sure that every member's firewall allows for communication using Ethiris, both ingoing and outgoing traffic. By default, communication between cluster members takes part on port 1239 TCP. This port has to be open for each server in a possible firewall.

All servers in the cluster should have access to a shared resource to save events. This could be a shared folder on a server or even better, a redundant NAS. Lacking this, the cluster will still run fine, but the event list in the clients will not be replicated properly if a cluster server is disconnected. Keep in mind that Ethiris Server normally cannot reach files or folders on the network, see *2.4.29 Storages node* on page *2:158* for more information.

The script runs on the server, which is currently master. All available functionality in the script is still usable. Though keep in mind that execution of the script may be moved to a different server if the master is disconnected. The servers in the cluster continuously synchronize the internal data store and internal script objects. This is to make sure the other servers will always be ready to continue the execution of the script. The script must be written in such a way that locally created objects can be properly recreated when another server starts the execution of the script. This is usually not a problem, only in very advanced scripts would this limitation mean that it needs to be somewhat rewritten.

If the configuration contains incoming TCP- or HTTP-connections, the external systems that connect to the cluster have to handle that connections are only possible with the current master server.

When using security features, see *2.4.56 Security node* on page *2:255*, it is essential only to use user groups that are available in the same domain as all cluster members, or even better, utilize users/user groups defined in Ethiris, see *2.4.56 Security node*. If local user groups for only one server were to be used as a requirement for authorization problems would ensue, as only that specific server would be able to access those features. The selection of local user groups has been simplified by dividing user groups into two categories when selecting, for example, ”Required User Group”, local user groups in the cluster, and local user groups who are missing on some members in the cluster. Only the user groups in the first category are available for selection.

Generally, Ethiris supports all versions of Windows from Windows 7 SP1 and later and is server operating system, Windows Server 2008 SP2 and later. We recommend running the cluster on later versions of Windows or Windows Server.

2.4.6 Cameras node

Under each Ethiris Server in the treeview, there is a *Cameras* node. This is a container node for all network cameras and video encoders that are connected to the server.

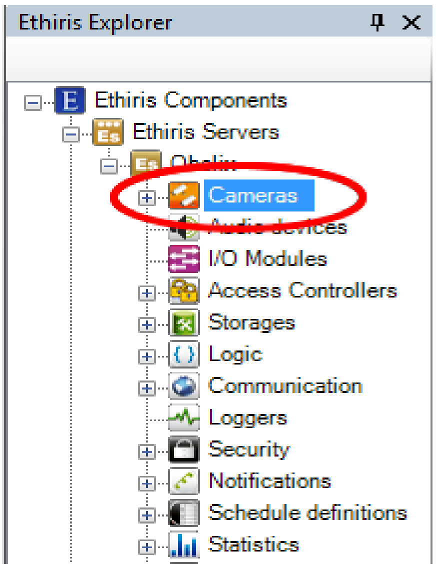


Figure 2.109 The Cameras node in Ethiris Explorer treeview.

Cameras popup menu

Right-clicking this node brings up a context menu.

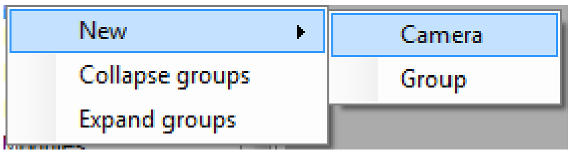


Figure 2.110 The popup menu for the Cameras node.

New->Camera adds a new camera to the server configuration. It is immediately visible in the treeview as a new camera node. Should you have opened the *Cameras* panel, the new camera would be added there too.

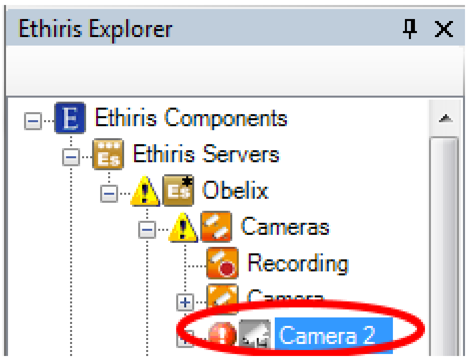


Figure 2.111 New camera added.

Notice the error icon to the left of the new camera node. This is due to the new camera has no IP address yet. There are warning icons further up in the treeview indicating an error somewhere in the configuration.

Also, notice the camera icon itself. It indicates, by its grey color, that there is no communication with the camera. The small diskette symbol indicates that the camera has not yet been saved to Ethiris Server, hence the non-existing communication.

New->Group adds a new camera group to the server configuration. It is immediately visible in the treeview. The camera groups are only visible in the treeview.

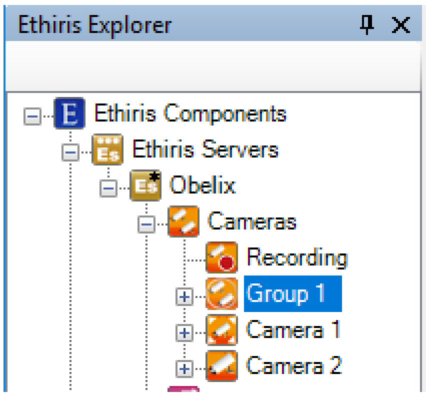


Figure 2.112 New camera group added

When one or more camera groups are created, it is possible to drag and drop cameras to/from groups directly in the treeview. It is also possible to right-click the cameras and select in the popup menu to move the camera to or from a group.

The camera groups are used to make the configuration of many cameras easier. Both storage settings and privilege settings made on the group level are automatically inherited by all cameras in the group. That means that all changes made on the group level automatically affect all cameras in the group. There is the possibility to override specific settings on the cameras to have different settings for cameras in the group.

The camera groups can also be used from script to easily start recording on all cameras in the group simultaneously or activate all cameras in the group based on a schedule. Read more about this in chapter 3.

Collapse groups close all groups in the treeview.

Expand groups opens all groups in the treeview and reveals any cameras in each group.

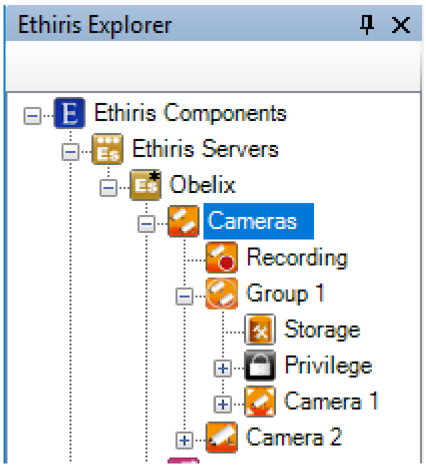
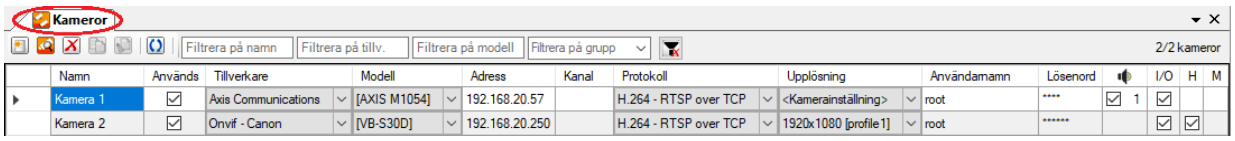


Figure 2.113 Camera group expanded

Cameras panel

Double-clicking the *Cameras* node in the treeview opens the corresponding panel.



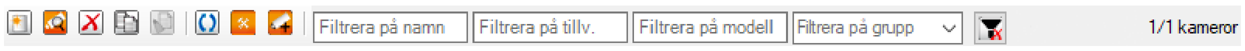
	Namn	Används	Tillverkare	Modell	Adress	Kanal	Protokoll	Upplösning	Användarnamn	Lösenord	🔈	I/O	H	M
▶	Kamera 1	☑	Axis Communications	[AXIS M1054]	192.168.20.57		H.264 - RTSP over TCP	<Kamerainställning>	root	****	☑ 1	☑		
	Kamera 2	☑	Onvif - Canon	[VB-S30D]	192.168.20.250		H.264 - RTSP over TCP	1920x1080 [profile1]	root	******		☑	☑	

Figur 2.114 The Cameras panel.

This panel consists of a list of all cameras currently part of the server configuration.

At the top of the panel, there is a toolbar.

Cameras panel toolbar



Figur 2.115 The toolbar in the Cameras panel.

New Camera.	Use this button to create a new camera. This is the same as *New->Camera* in the popup menu described above. A new camera is immediately added to the server configuration. See *Figure 2.117* for an example of how it looks in the camera list.
Browse for camera(s)	Use this button to browse the network for available cameras. This works for cameras with support for UPnP (Plug-and-play) and/or ONVIF. See *Figure 2.118* for an example of how the browse dialog looks like.
Delete camera	Use this button to delete the selected camera(s) from the configuration. You can select more than one camera by using the *Ctrl*-button and/or the *Shift*-button.
Copy	Use this button to copy the selected camera from the configuration. Use the *Paste* button to create a new camera with the same settings as the copied camera.
Paste	Use this button to paste the camera you recently copied. You can paste several cameras with the same settings by clicking the *Paste* button several times.
Refresh cameras	Use this button to force a refresh of all selected cameras that are probed. The effect is that Ethiris Server sends a request to all selected and probed cameras in the list (that has an IP address) for various properties such as *manufacturer, model, PTZ, resolutions/profiles, etc*. This is required after manually adding ONVIF and most Axis cameras since they are probed.

Camera settings	The button is only visible if only one camera is selected in the list, and that camera is an ONVIF camera. This button opens a new window in which you can edit the profiles of the camera. Changes made here are saved directly and immediately in the camera, not in the Ethiris Server Configuration.

In the *Profiles* tab, it's possible to add new profiles and delete existing ones (as long as the camera permits).

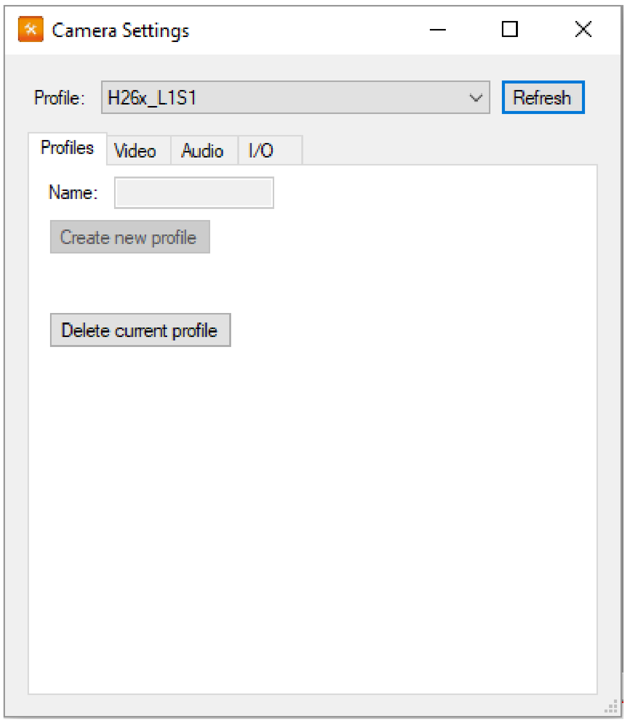


Figure 2.116 The Camera Settings dialog

In the *Video* tab, it's possible to edit various video settings in the currently selected profile.

In the Audio tab, it's possible to activate/deactivate support for audio out from the unit. Ethiris only supports audio using G.711 format.

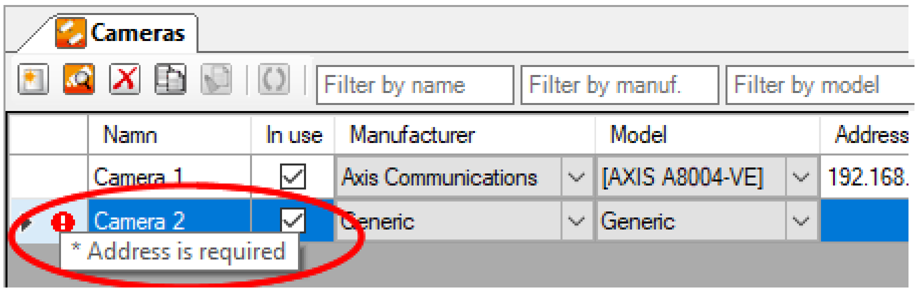
If the camera has digital I/O, the tab *I/O* enables editing of various settings for I/O.

Add camera to video encoder	The button is only visible if only one camera is selected in the list, and that camera is a camera connected to a video encoder or a network camera with support for multiple video channels. The button adds a new camera to the same video encoder or network camera as the selected camera is connected to.
Filter by name	This field is used to filter the camera list by camera name. Only cameras whos name match what you type in the field will be listed. The filter is case insensitive, and multiple keywords are separated using | (pipe).
Filter by manufacturer	This field is used to filter the camera list by the camera manufacturer. Only cameras whos manufacturer match what you type in the field will be listed. The filter is case insensitive, and multiple keywords are separated using | (pipe).
Filter by model	This field is used to filter the camera list by model. Only cameras whos model match what you type in the field will be listed. The filter is case insensitive, and multiple keywords are separated using | (pipe).
Filter by group	Check one or several camera groups in the list to show only cameras belonging to these groups. The filter is case insensitive, and multiple keywords are separated using | (pipe).
Clear filter	Click the button to clear all the filters.

X/y cameras

States how many cameras that are shown in the list (x) based on the current filtering. Y states the total number of cameras in the configuration.

New Camera



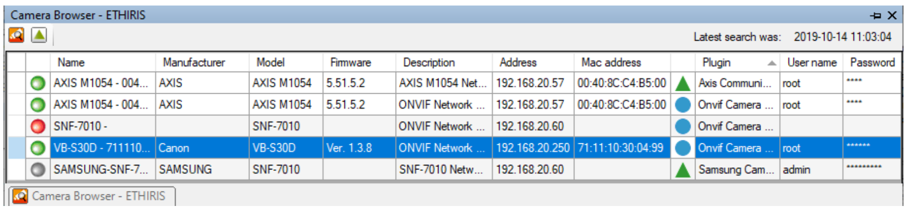
	Namn	In use	Manufacturer	Model	Address
	Camera 1	✓	Axis Communications	[AXIS A8004-VE]	192.168.
●	Camera 2	✓	Generic	Generic	

* Address is required

Figure 2.117 Camera added to the camera list.

When adding a new camera, no IP address is entered. In this case, it has to be done manually. Use *Browse for cameras* to get camera properties automatically.

Browse for camera(s)



Latest search was: 2019-10-14 11:03:04

	Name	Manufacturer	Model	Firmware	Description	Address	Mac address		Plugin	User name	Password
●	AXIS M1054 - 004...	AXIS	AXIS M1054	5.51.5.2	AXIS M1054 Net...	192.168.20.57	00:40:8C:C4:B5:00	▲	Axis Communi...	root	****
●	AXIS M1054 - 004...	AXIS	AXIS M1054	5.51.5.2	ONVIF Network ...	192.168.20.57	00:40:8C:C4:B5:00	●	Onvif Camera ...	root	****
●	SNF-7010 -		SNF-7010		ONVIF Network ...	192.168.20.60		●	Onvif Camera ...		
●	VB-S30D - 711110...	Canon	VB-S30D	Ver. 1.3.8	ONVIF Network ...	192.168.20.250	71:11:10:30:04:99	●	Onvif Camera ...	root	******
●	SAMSUNG-SNF-7...	SAMSUNG	SNF-7010		SNF-7010 Netw...	192.168.20.60		▲	Samsung Cam...	admin	*********

Figure 2.118 The Browse for cameras dialog.

In the *Browse for camera(s)* dialog, you can select the desired camera(s) and click the *Add* button to add the cameras to the configuration. You can use the *Ctrl*-button and *Shift*-button to select more than one camera. ONVIF cameras and cameras with support for Plug-and-Play will be found as long as they are present in the same subnet as Ethiris Server.

Enable UPnP to be able to browse for cameras with support for Plug-and-Play.

Make sure that UPnP is enabled on the computer to be able to browse the network for available cameras with support for Plug-and-Play. This is done in the Windows Firewall applet in the Control Panel. There is a tab called *Exceptions* where you can tick UPnP.

Cameras panel camera list

The camera list consists of several columns.

Name is the name of the camera. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the camera in the list indicating the problem.

In use is ticked as default, meaning that the camera is in use, and Ethiris Server retrieves video from the camera. If not ticked, Ethiris Server will not request video from the camera. In this case, of course, live video will not be available, and a new recording of video is not possible. However, existing recordings can be displayed or exported.

Manufacturer is the manufacturer of the camera. Each manufacturer has a specific set of models available. This column consists of a list of available manufacturers.

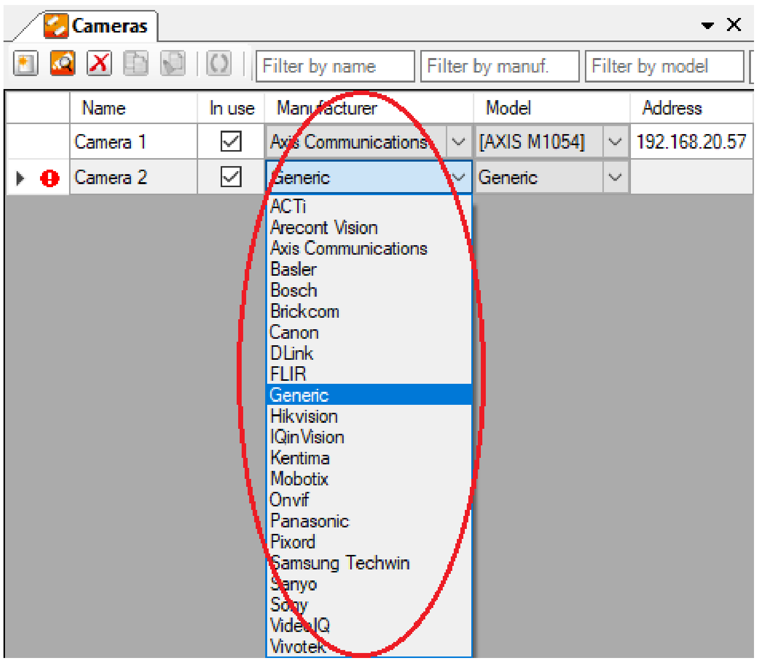


Figure 2.119 The Manufacturer list.

Model is the model of the camera. This column consists of a list of available camera models. In some cases, it says *[Probe]* (or model name inside [ ]) in the list. This means Ethiris can probe the camera and, in that way, retrieve the model and which functions the camera has (PTZ, I/O, etc.). For some manufacturers, there are some models listed and also probe. That means the listed models don't support probing but are a so-called known type in Ethiris.

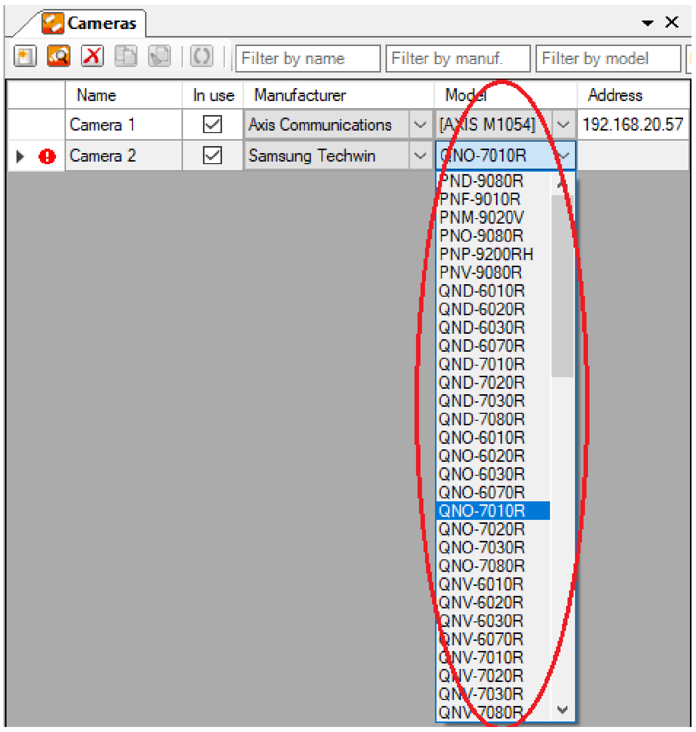
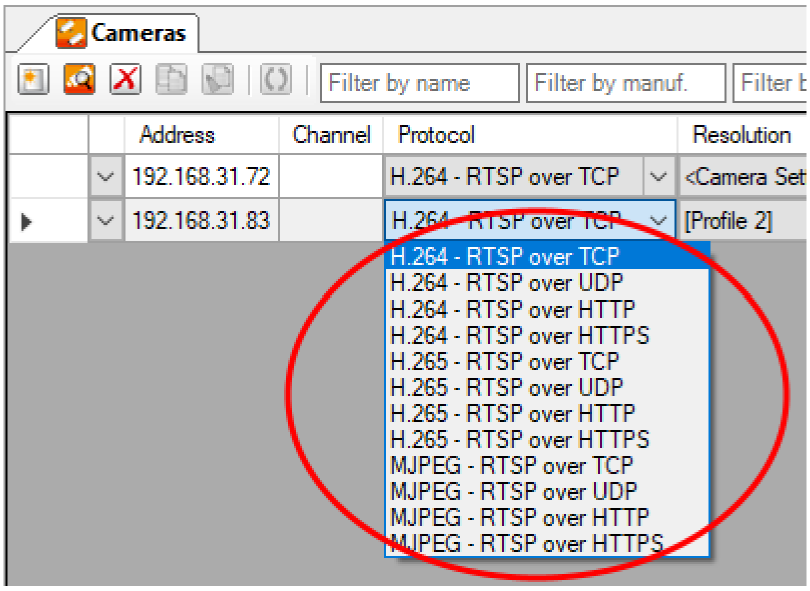


Figure 2.120 The Camera model list.

Address is the IP address for the camera (or DNS name if this was set up).

Channel is only used for video encoders and network cameras with multiple channels. In that case, it determines which channel to use for this camera.

Protocol determines what video compression method and transport protocol to use. Different camera models may have different available protocols. This column consists of a list of available combinations of video encodings and transport protocols.

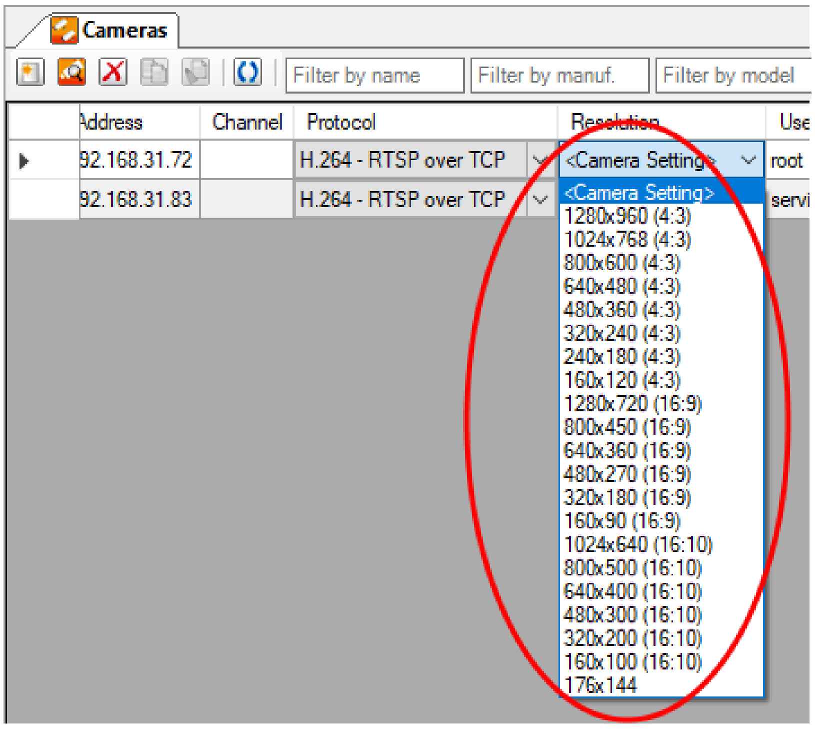


Address	Channel	Protocol	Resolution
192.168.31.72		H.264 - RTSP over TCP	<Camera Set
192.168.31.83		H.264 - RTSP over TCP	[Profile 2]

H.264 - RTSP over TCP
H.264 - RTSP over UDP
H.264 - RTSP over HTTP
H.264 - RTSP over HTTPS
H.265 - RTSP over TCP
H.265 - RTSP over UDP
H.265 - RTSP over HTTP
H.265 - RTSP over HTTPS
MJPEG - RTSP over TCP
MJPEG - RTSP over UDP
MJPEG - RTSP over HTTP
MJPEG - RTSP over HTTPS

Figure 2.121 The Protocol list.

Resolution determines what resolution to use. Different camera models may have different available resolutions. This column consists of a list of available resolutions. If you select *<Camera Setting>*, the resolution is determined by the settings directly in the camera. In this case, if you change the resolution in the camera, you will also affect the video sent to Ethiris. If you, on the other hand, specifically select a resolution in the Ethiris server configuration, this resolution will always be used by Ethiris when requesting video from the camera regardless of the setting in the camera itself.



Address	Channel	Protocol	Resolution	Use
92.168.31.72		H.264 - RTSP over TCP	<Camera Setting>	root
92.168.31.83		H.264 - RTSP over TCP		servi

<Camera Setting>
1280x960 (4:3)
1024x768 (4:3)
800x600 (4:3)
640x480 (4:3)
480x360 (4:3)
320x240 (4:3)
240x180 (4:3)
160x120 (4:3)
1280x720 (16:9)
800x450 (16:9)
640x360 (16:9)
480x270 (16:9)
320x180 (16:9)
160x90 (16:9)
1024x640 (16:10)
800x500 (16:10)
640x400 (16:10)
480x300 (16:10)
320x200 (16:10)
160x100 (16:10)
176x144

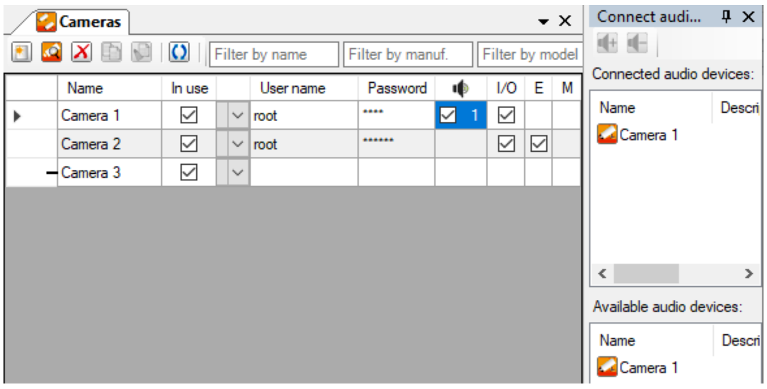
Figure 2.122 The Resolution list.

User name is the user name to use if the camera requires login.

Password is the password to use if the camera requires login.

🔈 A checkbox in this cell indicates that the camera supports audio out. If it is checked, the camera becomes available in the panel *Connect audio device to camera.* The panel is opened by clicking in the cell outside of the checkbox. **Note** that Ethiris, at the moment, only supports audio using the G.711 codec.

To the right of the checkbox, a number indicates how many audio devices are connected to this camera. Connected audio devices will get audio simultaneously when you activate the microphone for this camera. This way, it is possible to configure the camera to send audio to several different cameras at the same time, and hence it becomes effortless for the operator to speak to large areas at the same time.



Name	In use		User name	Password	🔈	I/O	E	M
Camera 1	✓		root	****	✓ 1	✓		
Camera 2	✓		root	******		✓	✓	
Camera 3	✓							

Connected audio devices:

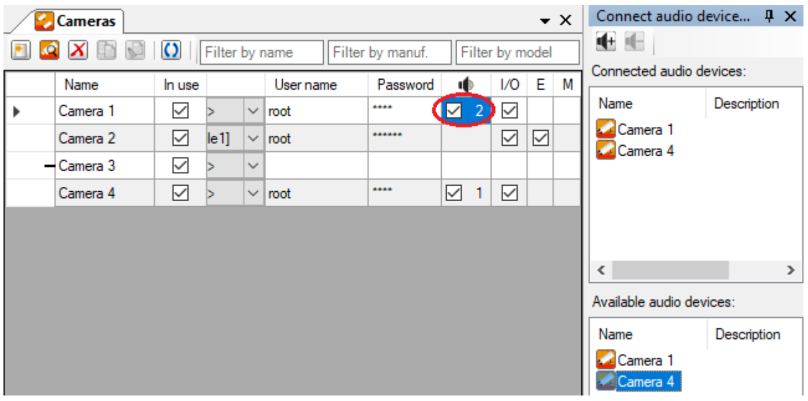
Name	Descri
Camera 1	

Available audio devices:

Name	Descri
Camera 1	

Figure 2.123 The panel Connect audio device to camera opened for Camera 1.

In the panel, connected audio devices are shown in the upper part and available audio devices in the lower part. By dragging and dropping cameras between the two parts, you select which audio devices should receive audio when the operator activates the microphone for the camera in Ethiris Client.



Name	In use		User name	Password	🔈	I/O	E	M
Camera 1	✓	>	root	****	✓ 2	✓		
Camera 2	✓	le1]	root	******		✓	✓	
Camera 3	✓	>						
Camera 4	✓	>	root	****	✓ 1	✓		

Connected audio devices:

Name	Description
Camera 1	
Camera 4	

Available audio devices:

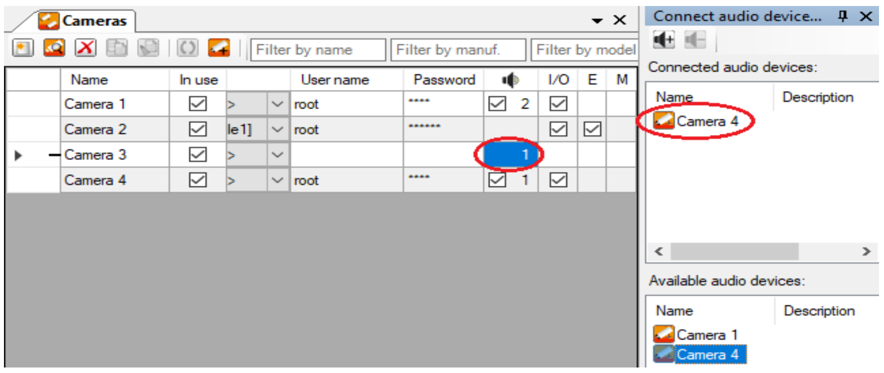
Name	Description
Camera 1	
Camera 4	

Figure 2.124 Audio devices for two cameras connected to Camera 1.

This means that when the operator activates the microphone for *Camera 1,* the audio will be sent in parallel to both *Camera 1* and *Camera 4.*

In this configuration, when the operator activates the microphone for *Camera 4, the* audio will only be sent to *Camera 4.*

Camera 3, in the example above, doesn't have audio support (you can tell as there is no checkbox in the column for *Camera 3*). It is still possible to connect audio devices from other cameras to this camera and hence get audio support on a camera that doesn't have that build in.



Name	In use		User name	Password	🔈	I/O	E	M
Camera 1	✓	>	root	****	✓ 2	✓		
Camera 2	✓	le1]	root	******		✓	✓	
— Camera 3	✓	>			1			
Camera 4	✓	>	root	****	✓ 1	✓		

Connected audio devices:

Name	Description
Camera 4	

Available audio devices:

Name	Description
Camera 1	
Camera 4	

Figure 2.125 Audio device connected to a camera that lacks audio support.

In this case, the audio will be sent to *Camera 4* regardless if the operator activates the microphone for *Camera 3* or *Camera 4.*

I/O should be ticked if you want to use the I/O port of the camera. As a default, I/O communication is enabled.

E should be ticked if you want to subscribe to events from the camera. Providing the camera has support for this. This can be a notification for when built-in analyzers in the camera sends data, status, etc. from the camera itself. Events from cameras can be handled in the script, see *3.23.1 Events in Cameras object* for more information.

M should be ticked if you want to receive metadata from the camera. Providing that the camera has support for this. Metadata can be seen as an extension of events since the events are also shown in the metadata stream. But with metadata, there comes more information, e.g., where the motion has been detected in the image, in the form of coordinates, and not just that there was a motion detected. Also, the classification of the object can be available, for instance, if it was a human, car, or animal that was detected depending on the capabilities of the camera. Metadata is handled in the script the same way as events, see *3.23.1 Events in Cameras object* for more information.

Multiupdate

With version 11 comes the ability to update the same settings on multiple items, in this case, cameras, at the same time. By first selecting multiple cameras in the far left column, either by left click and drag marking, or holding ctrl/shift to mark separate or nearby items.

Then when you select a setting, for instance, Resolution, and click the dropdown, all the marked cameras will get a dash,-, in the far left column. And when the change is made in that cameras dropdown, it will be made for all the marked cameras. Multiple changes can be made to the same selection.

This, together with filtering, makes it very easy to update passwords, change resolution, activate I/O, Events, and so on for multiple cameras.

The same functionality works in the Recording panel and the Camera Browser.

2.4.7 Recording node

Under each Ethiris Server in the treeview, there is a *Camera Recording* node. The purpose is to be able to configure automatic recording easily.

By double-clicking the node in the treeview, the corresponding panel is opened where all cameras connected to the Ethiris server are presented in a table. Both network cameras and analog cameras connected via video encoders are presented in the same table.

For each camera in the table, you can determine the settings for automatic recording. For event recording, you can select motion detection and/or a schedule as the condition, and for the continuous recording, you can select a schedule as a condition for the start of recording.

Via the table, you can automatically create motion detectors for each camera with three different settings for sensitivity. It is also possible to create a standard schedule directly from the table.

The configuration made in the table runs ”in parallel” with a possible script. This means that recording is started if either the conditions in this table are fulfilled or if logic in the script implies the start of recording. To avoid confusion, we recommend that you use either this simple recording table or script, not both.

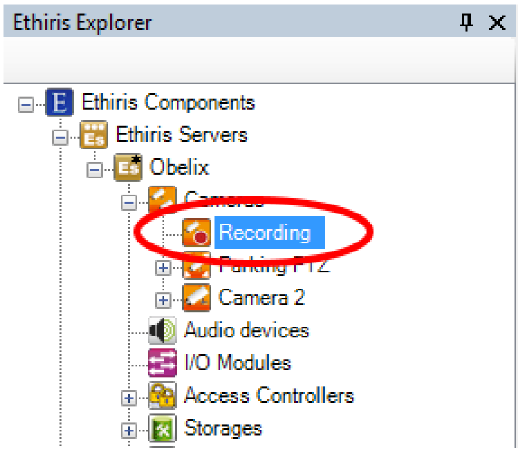


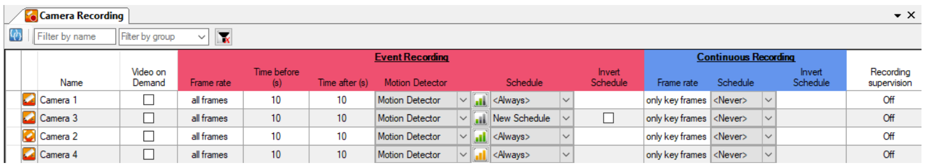
Figure 2.126 The node Camera Recording in Ethiris Explorer treeview.

Camera Recording popup menu

This node has no popup menu.

Camera Recording panel

Double-clicking the node *Camera Recording* in the treeview opens the corresponding panel.



		Event Recording						Continuous Recording			
Name	Video on Demand	Frame rate	Time before (s)	Time after (s)	Motion Detector	Schedule	Invert Schedule	Frame rate	Schedule	Invert Schedule	Recording supervision
Camera 1	☐	all frames	10	10	Motion Detector	<Always>		only key frames	<Never>		Off
Camera 3	☐	all frames	10	10	Motion Detector	New Schedule	☐	only key frames	<Never>		Off
Camera 2	☐	all frames	10	10	Motion Detector	<Always>		only key frames	<Never>		Off
Camera 4	☐	all frames	10	10	Motion Detector	<Always>		only key frames	<Never>		Off

Figure 2.127 The panel Camera Recording.

This panel consists of a list of all cameras that are currently part of the configuration.

At the top of the panel, there is a toolbar.

Recording panel toolbar

Figure 2.128 The toolbar in the Camera Recording panel.

Convert to script. Use this button to convert recording conditions to script code and reset all recording conditions defined in this panel after conversion. Use this only if you want to configure more advanced recording conditions than are possible to achieve in this panel. The script needs to be empty before invoking the convert function.

Recording panel camera list

The camera list consists of several columns.

Name is the name of each camera. The name cannot be changed in this context but is only used for presentation purposes. The icon to the left of the name indicates the current status of the camera. Orange indicates that the communication with the camera is OK, grey indicates that there is some kind of problem with the camera communication.

Video on demand is normally not checked. The normal procedure for Ethiris is to request a video stream from the camera, such as video is continuously sent from the camera to Ethiris Server over the network. There are advantages to this procedure, e.g., you can utilize the built-in motion detection feature in Ethiris, and you can also utilize the *Time before* parameter for event recording.

However, these features are not always beneficial, and perhaps you prefer to minimize the network traffic. Then you can check this option, which means that Ethiris Server only requests video from the camera when it is necessary. This happens when a client requests live video from the camera or when a recording condition for the camera becomes true.

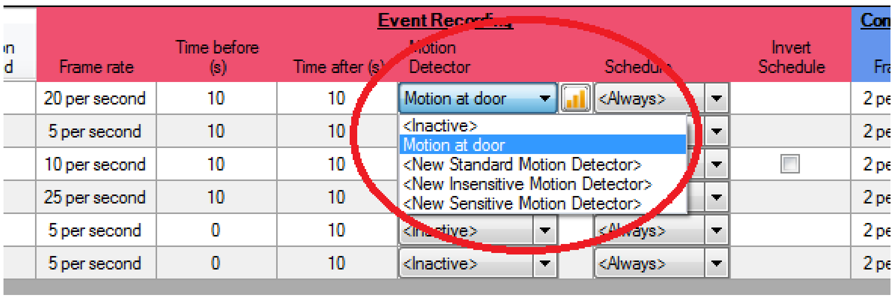
Event Recording

Frame rate determines the frame rate that will be used for event recording. It can be expressed per *seconds*, *minutes*, *hours,* or *all frames*.

Time before determines the time in seconds that recording will be performed before the condition for event recording is true. If, e.g. motion detection is used as a condition for event recording and time before is set to 10 seconds, the recording will comprise 10 seconds before the motion was detected. This setting also denotes maximal Instant replay time, i.e., how many seconds back in time Instant replay can display.

Time after determines the time in seconds that recording will be performed after the condition for event recording is no longer true. If motion detection is used as a condition for event recording and time after is set to 10 seconds, the recording will continue 10 seconds after motion is no longer detected.

Motion Detector determines which motion detector to use for starting event recording. In this list, all existing motion detectors are presented as well as the different alternatives for creating new detectors (*Standard*, *Sensitive,* and *Insensitive*). The choice *Inactive* means that you don’t want to use motion detection for starting event recording.



	Event Recording						Con
n d	Frame rate	Time before (s)	Time after (s)	Motion Detector	Schedule	Invert Schedule	Fr
	20 per second	10	10	Motion at door	<Always>		2 pe
	5 per second	10	10	<Inactive>			2 pe
	10 per second	10	10	Motion at door		☐	2 pe
	25 per second	10	10	<New Standard Motion Detector>			2 pe
				<New Insensitive Motion Detector>			
				<New Sensitive Motion Detector>			
	5 per second	0	10	<Inactive>	<Always>		2 pe
	5 per second	0	10	<Inactive>	<Always>		2 pe

Figure 2.129 The list of motion detectors.

Motion Detector icon indicates if one of the three standard settings is used or if another setting applies for the current motion detection. Click on the icon opens the definition for the currently selected motion detection in a separate panel.

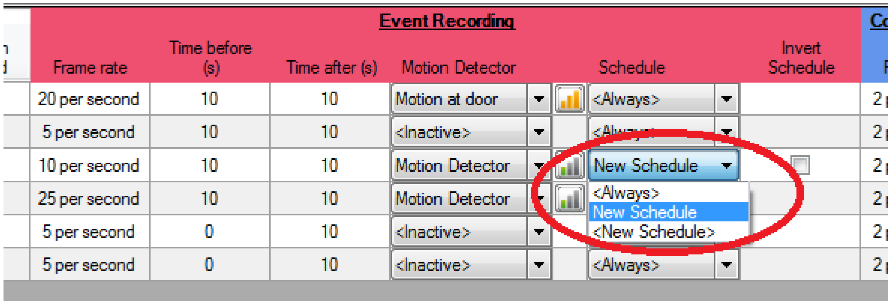
User defined The settings for the selected motion detector does not follow any of the three standard alternatives.

Sensitive The currently selected motion detector has settings that correspond to the alternative *Sensitive*. This means that *Background filtering* is activated, *Sensitivity* is set to 75, *Resolution* 1/3, *Frame rate* 1 per second, *Number of pictures for trigger* is set to 1, and *Triggering Level* is 2.00 %.

Standard The currently selected motion detector has settings that correspond to the alternative *Standard*. This means that *Background filtering* is inactivated, *Sensitivity* is set to 60, *Resolution* 1/3, *Frame rate* 2 per second, *Number of pictures for trigger* is set to 1, and *Triggering Level* is 5.00 %.

Insensitive The currently selected motion detector has settings that correspond to the alternative *Sensitive*. This means that *Background filtering* is inactivated, *Sensitivity* is set to 50, *Resolution* 1/5, *Frame rate* 2 per second, *Number of pictures for trigger* is set to 2, and *Triggering Level* is 10.00 %.

Schedule determines which schedule to use. In this list, all existing schedules, as well as the alternative for creating a new schedule (New Schedule), are presented. The choice *Always* means that you do not want to use a schedule for starting event recording. If both a motion detector and a schedule are selected, the schedule has to be active for recording to start.



	Event Recording						Co
	Frame rate	Time before (s)	Time after (s)	Motion Detector	Schedule	Invert Schedule	F
	20 per second	10	10	Motion at door	<Always>		2
	5 per second	10	10	<Inactive>			2
	10 per second	10	10	Motion Detector	New Schedule	☐	2
	25 per second	10	10	Motion Detector	<Always>		2
					New Schedule		
	5 per second	0	10	<Inactive>	<New Schedule>		2
	5 per second	0	10	<Inactive>	<Always>		2

Figure 2.130 The list of schedules.

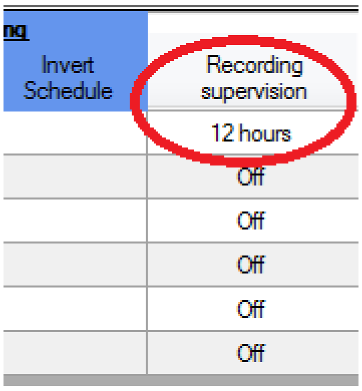
Invert schedule determines if the recording will occur when the schedule is active or inactive. If the checkbox is ticked, the recording will occur when the schedule is inactive.

Continuous Recording

Frame rate determines the frame rate that will be used for event recording. It can be expressed per *seconds*, *minutes*, *hours,* or *all frames*.

Schedule determines which schedule to use. The function is the same as for event recording described above except for the additional alternative *Never,* which means that continuous recording never will occur.

Invert schedule determines if the recording will occur when the schedule is active or inactive. The function is the same as for event recording described above.



Invert Schedule	Recording supervision
	12 hours
	Off
	Off
	Off
	Off
	Off

Figure 2.131 Recording supervision set to 12 hours.

Recording supervision is normally set to *Off*. But, by selecting the field, you can scroll the mouse wheel or click the arrows to select a time between one hour and seven days. If the recording supervision is active, it means that Ethiris Server supervises if a recording is done within the specified time. If not, an alarm named *RecordingError* is activated.

Note that one or more fields for a camera can be grayed out and that it is not possible to edit the value in this panel. That is because the camera is a member of a camera group and that you have not made any overrides on the camera for these settings.

If you want to edit the grayed out settings, go to the camera group the camera is a member of, and edit the storage settings there.

2.4.8 Camera group node

Under the *Cameras* node in the treeview, there can be one or many *Camera group* nodes. The meaning with camera groups is to make the configuration of cameras faster by grouping cameras that one way or another is linked together. It can be that the cameras are all mounted on the same building or at a specific customer and that you want to handle these cameras separately from other cameras in the system.

There are settings for storage and privilege on the group level. All cameras that are added to the group will automatically inherit all these settings. If you want to change the settings, change them on the group level, and all the cameras in the group will automatically use the changed settings.

You can also change settings for each camera separately, if you want, by overriding the group settings for storage or privilege and that way, you can have different settings on cameras in a camera group.

The camera groups can also be used from script to easily start recording on all cameras in the group simultaneously or activate all cameras in the group based on a schedule. Read more about this in chapter 3.

When one or more camera groups are created, it is possible to drag and drop cameras to/from groups directly in the treeview. It is also possible to right-click the cameras and select in the popup menu to move the camera to or from a group.

To change the name of the camera group, click the already selected camera group in the treeview to start to edit the name. When you are finished, press <*Return*> on your keyboard.

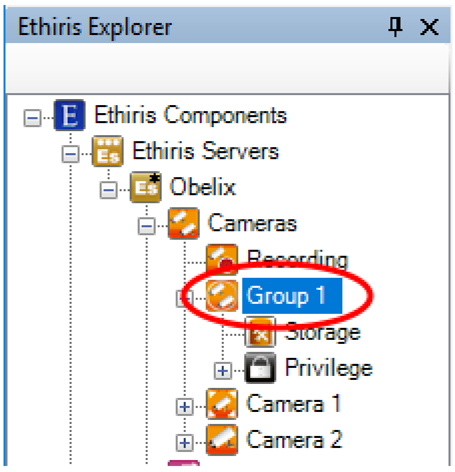


Figure 2.132 A camera group node in Ethiris Explorer treeview.

Camera group popup menu

Right-clicking such a node brings up a context menu.

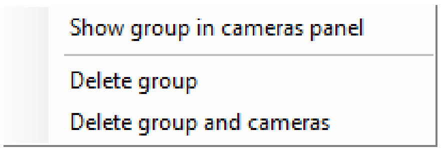


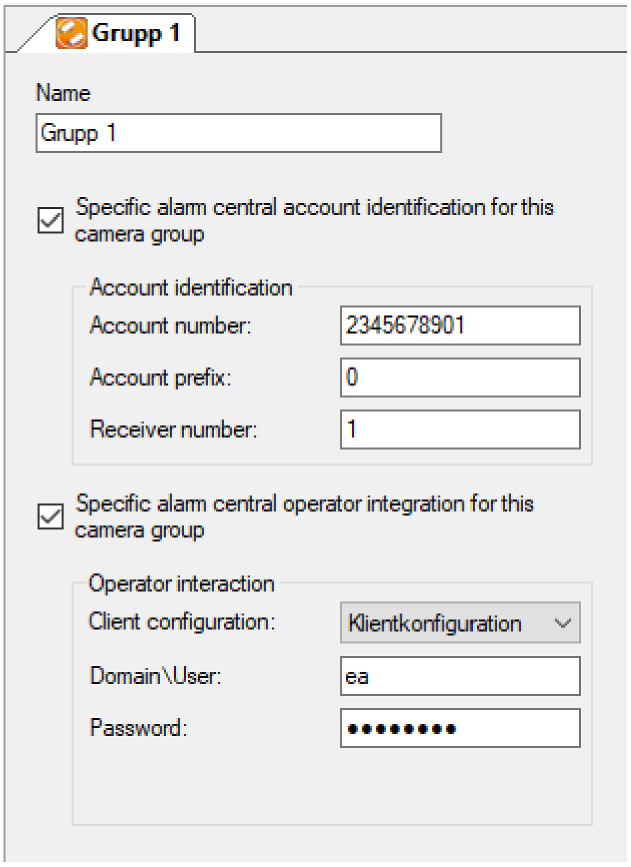
Figure 2.133 The popup menu for a Camera group node.

Delete group will move all cameras from the group and then delete the group. The cameras will no longer be part of any group.

Delete group and cameras will delete all cameras in the group and then the group itself.

Camera group panel

Double-clicking the node *Camera group* in the treeview opens the corresponding panel.



Name: Grupp 1

☑ Specific alarm central account identification for this camera group

Account identification

Account number:	2345678901
Account prefix:	0
Receiver number:	1

☑ Specific alarm central operator integration for this camera group

Operator interaction

Client configuration:	Klientkonfiguration
Domain\User:	ea
Password:	●●●●●●●●

Figure 2.134 The panel Camera group.

In this panel, you can enter settings for the camera group. The alarm central settings will be used by all cameras in the group, i.e., when an alarm or event that references cameras in this camera group is to be sent to the alarm central, account identification and/or operator integration settings defined here will be used if the corresponding checkbox is checked. This can be useful if more than one customer shares the same server, and they have different account identifications at the alarm central.

Name indicates the name of the camera group. In the case of several different customers share the same server, you can enter the name of the customer here or just group the cameras according to whatever you want.

The following settings will only be visible if you have a license that allows for alarm central settings, and alarm central is enabled in the alarm central tab of the server panel.

Specific alarm central account identification for this camera group denotes if cameras in this camera group will use different account identification settings than are defined in the alarm central tab of the server panel.

Account identification

Account number Enter your account number at the alarm central here.

Account prefix Enter account prefix if used by the alarm central. Otherwise, enter 0.

Receiver number Enter receiver number here if used by the alarm central. Otherwise, enter 1.

Specific alarm central operator integration for this camera group denotes if cameras in this camera group will use different operator integration settings than are defined in the alarm central tab of the server panel.

Operator integration

Client configuration If the alarm central has Ethiris Client installed locally and you have an agreement that they should connect to your system when an alarm is sent, enter the client configuration they will use. The client configuration must be handled by this server, and you must have entered the public IP-address and port in the fields *External address* and *External port.* Your router must be configured to allow connections to Ethiris from outside of your local network.

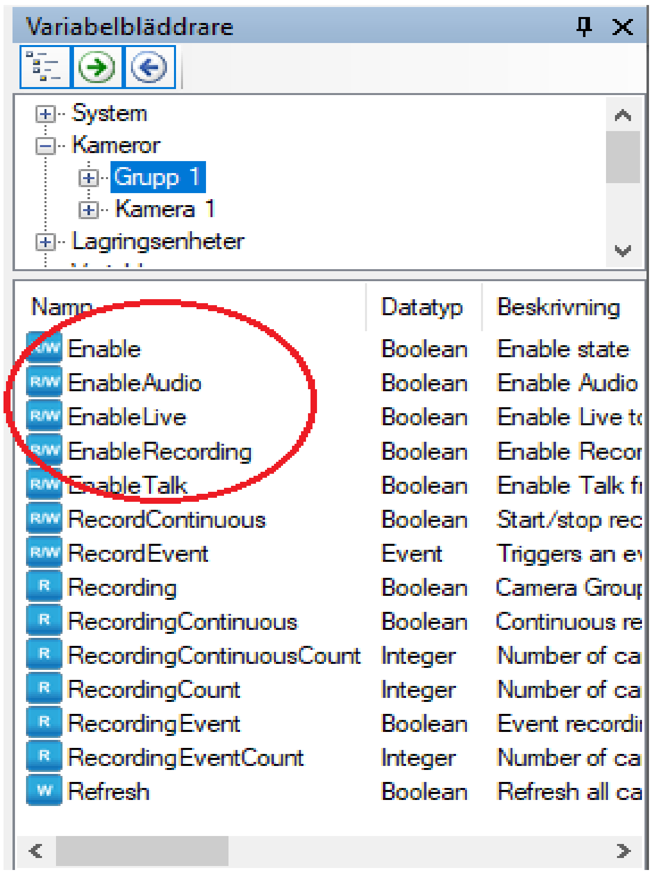
Domain\User Enter username the alarm central should use when connecting to your system. If no login is required, leave the field blank.

Password Enter the password the alarm central should use when connecting to your system. If no login is required, leave the field blank.

Camera group variables

When defining a *Camera group*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The variable browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.135* for an example where a *Camera group* is selected, and the variables belonging to the camera group are presented in the lower pane (encircled).



Namn	Datatyp	Beskrivning
Enable	Boolean	Enable state
EnableAudio	Boolean	Enable Audio
EnableLive	Boolean	Enable Live t
EnableRecording	Boolean	Enable Recor
EnableTalk	Boolean	Enable Talk f
RecordContinuous	Boolean	Start/stop rec
RecordEvent	Event	Triggers an e
Recording	Boolean	Camera Grou
RecordingContinuous	Boolean	Continuous re
RecordingContinuousCount	Integer	Number of ca
RecordingCount	Integer	Number of ca
RecordingEvent	Boolean	Event recordi
RecordingEventCount	Integer	Number of ca
Refresh	Boolean	Refresh all ca

Figure 2.135 Variables for a Camera group.

We will discuss script and variables later on in this manual, but it seems like a good idea to shortly describe some of the variables right now.

There are a whole bunch of variables for a camera group. We will briefly look at some of them. All variables on the group level affect all cameras in the group.

Enable is a read and writable variable that enables all cameras in the group. It can, for example, be connected to a schedule to easily enable/disable all cameras in the group based on the schedule.

EnableLive is a read and writable variable that allow clients to display the live video of the cameras in the group. When this set to *false,* live video cannot be displayed in any client for the cameras in the group.

EnableRecording is a read and writable variable that allows recording to be done for the cameras in the group. If this set to *false,* recording is prevented for all cameras in the group.

2.4.9 Storage node

Under the Camera group node in the treeview is the *Storage* node.

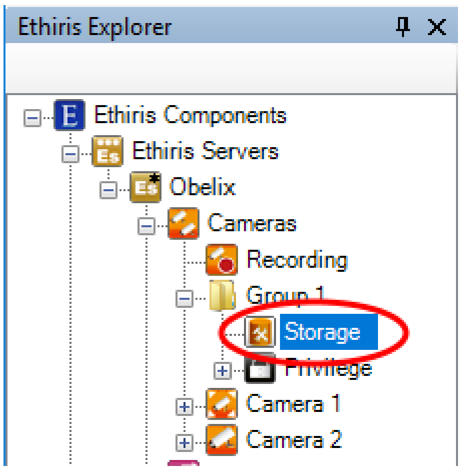


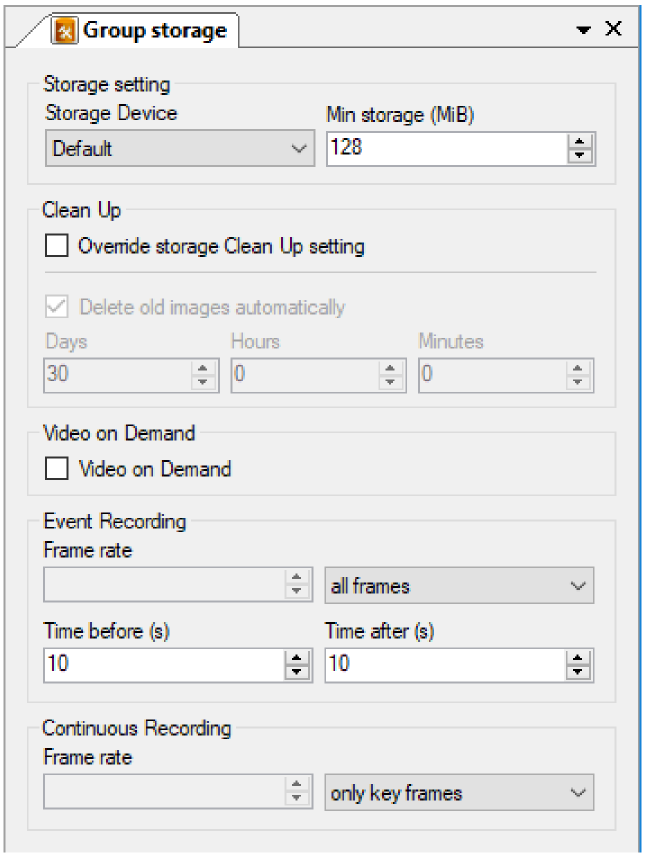
Figure 2.136 The Storage node for a Camera group in the Ethiris Explorer treeview.

Storage popup menu

There is no popup menu for this node.

Storage panel

Double-clicking a *Storage* node for a Camera group in the treeview opens the corresponding panel.


Group storage

Storage setting
Storage Device: Default
Min storage (MiB): 128

Clean Up
☐ Override storage Clean Up setting
✓ Delete old images automatically
Days: 30
Hours: 0
Minutes: 0

Video on Demand
☐ Video on Demand

Event Recording
Frame rate: all frames
Time before (s): 10
Time after (s): 10

Continuous Recording
Frame rate: only key frames

Figure 2.137 The Camera group Storage panel for a camera group.

In this panel, you can enter recording settings for the camera group. The settings will be inherited by all cameras in the group as long as they're not overridden in the camera storage panel.

Storage setting

Storage Device indicates which of the defined storage devices the camera will use to save video. Storage devices are defined in the *Storages* panel for the Ethiris Server.

Min storage (MiB) indicates how much hard disk space must always be protected for this camera's video. When the storage space for video is full, Ethiris automatically removes the oldest video to make room for new video. If one camera in the system records a high volume of video, there is a risk that video from other cameras will be deleted. Using this setting, you can protect part of the video for each camera.

Clean Up

Override storage Clean Up setting should be checked if you want to change the clean-up settings for this camera group. Default is to obey the general settings entered in the common panel for the whole Ethiris Server, *Storages*.

Delete old images automatically. Check this box if you want recorded video to be removed from the hard disk automatically when it gets too old.

Days, Hours *&* **Minutes** defines which video is to be removed. All recorded video older than specified here are removed from the hard disk.

Video on demand

Video on demand should be checked if you don't want Ethiris Server to retrieve images from the camera all the time, but only when it is required. I.e., when an Ethiris Client wants to show live images from the camera or when a recording condition is true.

Note that if this choice is made, the *Time before* is set to 0 for Event Recording.

Event recording

Frame Rate defines the frame rate in frames per second (or what time unit is selected) when video is stored for an event. Ensure that this value does not conflict with the frame rate required from the camera entered in the *Picture Settings* panel.

Time before defines a period of time in seconds for which video is to be recorded before an event has occurred. This setting also denotes maximal Instant replay time, i.e., how many seconds back in time Instant replay can display.

Time after defines the number of seconds for which video is to be recorded after an event has occurred (after the condition for event recording is no longer *true*).

Continuous Recording

Frame rate indicates the frame rate for the storage of continuously recorded video. Ensure that this value does not conflict with the frame rate required from the camera entered in the *Picture Settings* panel.

2.4.10 Privilege node

Under the Camera group node in the treeview is also the *Privilege* node.

A number of specific user operations are defined in Ethiris. For each such user operation, you can, if you want, specify that the user who performs the operation must be a member of a specific user group. This may be either an Ethiris user group, a local user group on the computer running Ethiris Server, or a global user group in a domain or the Active Directory if the computers and users involved are members of a domain. To be able to specify a user group in the domain, it is necessary for both the computer running Ethiris Server to be a member of the domain and the logon entered by the user to be for an account in the same domain.

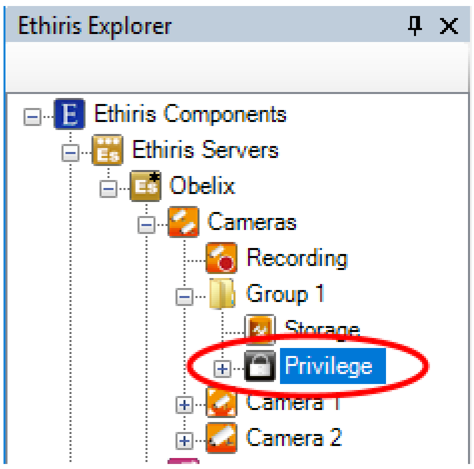


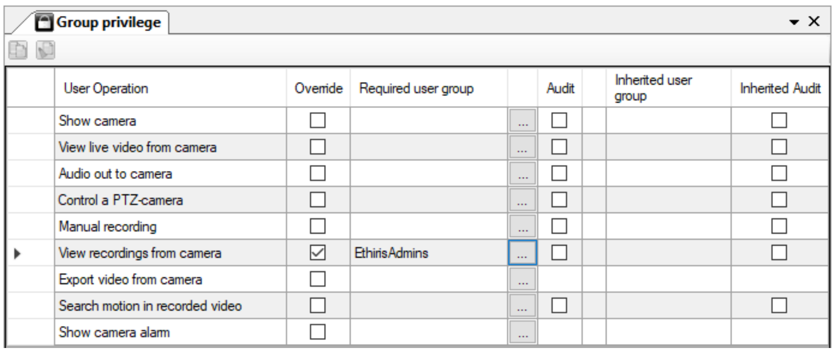
Figure 2.138 The Privilege node for a Camera group in Ethiris Explorer treeview.

Privilege popup menu

There is no popup menu for this node.

Privilege panel

Double-clicking a *Privilege* node for a camera group in the treeview opens the corresponding panel.



	User Operation	Override	Required user group		Audit	Inherited user group	Inherited Audit
	Show camera	☐		...	☐		☐
	View live video from camera	☐		...	☐		☐
	Audio out to camera	☐		...	☐		☐
	Control a PTZ-camera	☐		...	☐		☐
	Manual recording	☐		...	☐		☐
▶	View recordings from camera	☑	EthirisAdmins	...	☐		☐
	Export video from camera	☐		...			
	Search motion in recorded video	☐		...	☐		☐
	Show camera alarm	☐		...			

Figure 2.139 The Camera group Privilege panel.

In this panel, you can enter settings for the camera group regarding access control. The user operations listed here can be set at the Ethiris Server level (valid for all cameras) at the camera group level (valid for all cameras in the group) and specifically for each camera. In this panel, you set access control parameters for this camera group explicitly.

As a default, no login is required. Every user has access to all functions. If you want to restrict access to a particular user operation, you have to specify which user group the user has to be a member of to get access to the operation. To gain access, the user has to log in as a user that is a member of the required user group.

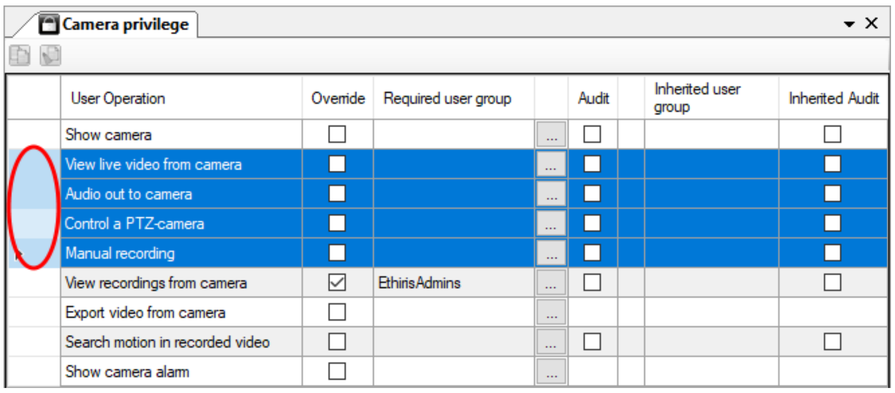
At the top of the panel, there is a toolbar.

Privilege panel toolbar

Figure 2.140 The toolbar in the Group privilege panel.

Copy. Use this button to copy the content of a specific row in the *User Operations* list.

Paste Use this button to paste the settings from the copied operation to one or several other operations. Select one or several operations by clicking in the area to the left of the *User Operation* column. See *Figure 2.141* below. Use the *Ctrl* or *Shift* keys on the keyboard together with the mouse to select several rows. You can also click and drag over several rows to select them.



User Operation	Override	Required user group		Audit	Inherited user group	Inherited Audit
Show camera	☐		...	☐		☐
View live video from camera	☐		...	☐		☐
Audio out to camera	☐		...	☐		☐
Control a PTZ-camera	☐		...	☐		☐
Manual recording	☐		...	☐		☐
View recordings from camera	☑	EthirisAdmins	...	☐		☐
Export video from camera	☐		...			
Search motion in recorded video	☐		...	☐		☐
Show camera alarm	☐		...			

Figure 2.141 Four rows selected.

The following columns are part of the *Operations* list in the Group privilege panel:

User Operation lists the available operations that you can set access control requirements for on this camera. These are:

Show camera – Makes the camera disappear completely in Ethiris Client unless a user that belongs to the specified user group is logged in.

View live video from camera – Restricts live video from this camera. In Ethiris Client, the message *Not authorized!* is displayed in the camera view if not, a user that is a member of the required group is logged in. See *Figure 2.161*.

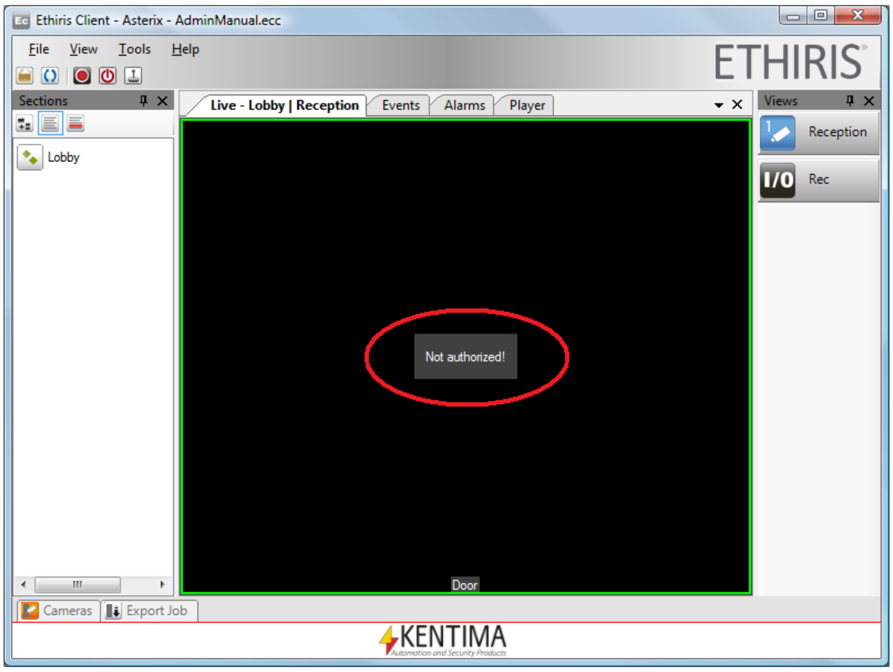


Figure 2.142 No one logged in when View live video from camera requires log in.

Audio out to camera – Gives access to the audio out on all cameras connected to this camera in the panel *Connect audio device to camera.*

Control a PTZ-camera – Restricts all optical PTZ operations, i.e., you cannot move or zoom the camera if not logged in as a member of the required group. Digital PTZ still works.

Manual recording – Restricts manual recording via the recording button in Ethiris Client. Clicking the recording button when not logged in simply does nothing. No recording is done.

View recordings from camera - Restricts recorded video from this camera. In Ethiris Client, the message *Not authorized!* is displayed in the camera view in the Player if not a user that is a member of the required group is logged in.

Export video from camera – Restricts all kind of export. If not logged in as a member of the required group, there will be no *Export* menu items in the various popup menus. If you click the Export button in the Player, only authorized cameras will appear in the camera list in the *Export dialog*.

Search motion in recorded video – Restricts motion search in the Client's Player. If not logged in as a member of the required group, the message *Can not search motion right now!* will be displayed when you select the menu item *Search next Motion* in the popup menu for the camera in the Player.

Show camera alarm – Restricts visibility of alarms and alarm events related to alarms on this camera unless you are logged in as a member of the required group.

OK, back to the columns in the *Camera Privilege* operations list:

Override has to be explicitly checked to be able to change the *Required user group*. A privilege setting can be set at the Ethiris Server level unless they are overridden explicitly in the *Group privilege* panel by checking *Override*.

Required user group specifies what user group the user has to be a member of to access this function. A blank field means that no log in is required. You can browse for available user groups by clicking the browse button to the right of this column.

Audit can be checked if you want to log when a particular operation is executed. For each operation for which *Audit* is specified, the system will save information on the time, the operation performed, who performed it, the client computer from which it was performed, and any other available information, depending on the type of operation performed.

It is permitted to specify that the system must log an operation without, at the same time, making any requirement that the user must be a member of a specific group. The operation will be logged regardless of this, but in these cases, there may be no information on who performed the operation if the user has not logged on. Other available information is logged as usual.

The audit log can be viewed in the *Events* panel in Ethiris Client.

Inherited user group is just information on the current setting on the Ethiris Server level (or camera group level if the camera is a member of a camera group). To override this setting, check the *Override* checkbox.

Inherited Audit is just information on the current setting on the Ethiris Server level. To override this setting, check the *Override* checkbox.

2.4.11 Client type privilege node

Under the node Camera group\Privilege in the treeview, there is a *Client type privilege* node.

The purpose of this node is to provide the opportunity to enter specific privilege settings for different types of clients. Today you can enter settings for *Ethiris Client, Ethiris Admin, Ethiris Mobile,* and *WideQuick EthirisView,* respectively.

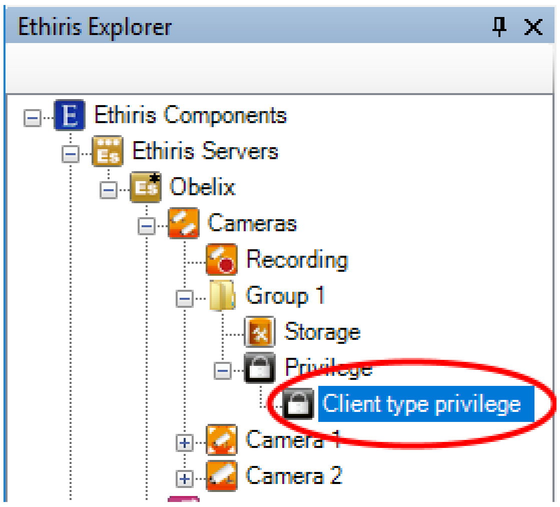


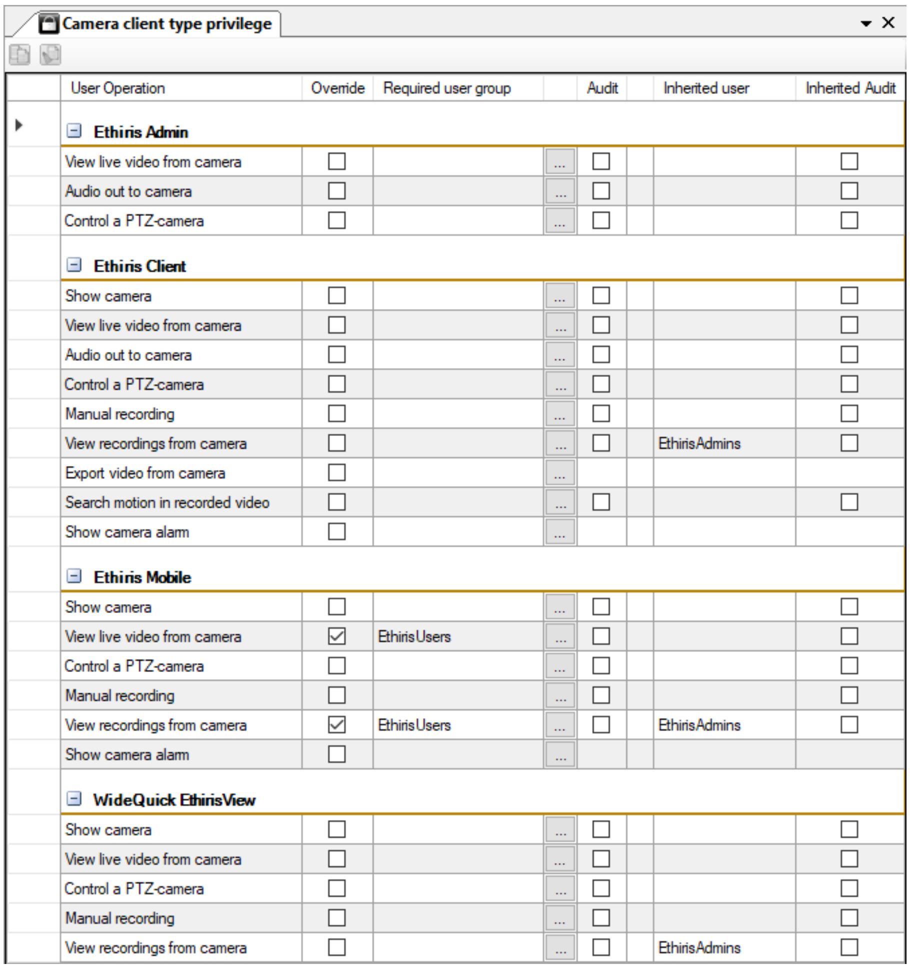
Figure 2.143 The node Client type privilege for a camera group in the Ethiris Explorer treeview.

Client type privilege popup menu

There is no popup menu for this node.

Client type privilege panel

Double-click on a *Client type privilege* node for a camera group in the treeview will open the corresponding panel.



Camera client type privilege

User Operation	Override	Required user group		Audit	Inherited user	Inherited Audit
Ethiris Admin						
View live video from camera	☐		...	☐		☐
Audio out to camera	☐		...	☐		☐
Control a PTZ-camera	☐		...	☐		☐
Ethiris Client						
Show camera	☐		...	☐		☐
View live video from camera	☐		...	☐		☐
Audio out to camera	☐		...	☐		☐
Control a PTZ-camera	☐		...	☐		☐
Manual recording	☐		...	☐		☐
View recordings from camera	☐		...	☐	EthirisAdmins	☐
Export video from camera	☐		...			
Search motion in recorded video	☐		...	☐		☐
Show camera alarm	☐		...			
Ethiris Mobile						
Show camera	☐		...	☐		☐
View live video from camera	☑	EthirisUsers	...	☐		☐
Control a PTZ-camera	☐		...	☐		☐
Manual recording	☐		...	☐		☐
View recordings from camera	☑	EthirisUsers	...	☐	EthirisAdmins	☐
Show camera alarm	☐		...			
WideQuick EthirisView						
Show camera	☐		...	☐		☐
View live video from camera	☐		...	☐		☐
Control a PTZ-camera	☐		...	☐		☐
Manual recording	☐		...	☐		☐
View recordings from camera	☐		...	☐	EthirisAdmins	☐

Figure 2.144 The panel Client type privilege for a camera group.

In the list, there are the same columns and the same operations as for camera group privilege. The difference is that the operations are divided between the different types of clients; *Ethiris Admin*, *Ethiris Client (regular Ethiris Client)*, *Ethiris Mobile (the mobile app Ethiris Mobile),* and *WideQuick EthirisView*. See the last section for an explanation of the various columns and operations.

For each type of client, the relevant operations are listed.

2.4.12 Camera node

Under the Cameras node in the treeview, there is one node for each network camera or analog camera connected to a video encoder that is part of the server configuration.

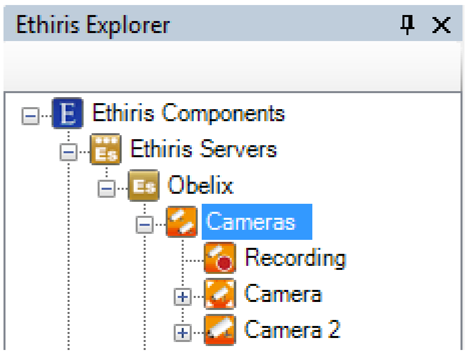


Figure 2.145 A camera node in the Ethiris Explorer treeview.

Each camera in the configuration is represented by an icon in the treeview. The icon indicates whether it is a fixed camera or a PTZ camera, and it also indicates the current status of the camera communication with Ethiris Server.

Type of camera

Fixed camera	The camera is fixed, I.e., not a PTZ camera,
PTZ camera	The camera is a PTZ camera with some possibility to control Pan, Tile, and/or Zoom.

Camera communication status

Communication OK	The camera is saved to Ethiris Server, is enabled, and the communication with the camera is working.
Inactive / Communication error	The camera is saved to Ethiris Server, but no communication exists with the camera. It may be caused by manually setting it to disabled or the reason maybe there is a communication error. On communication error, there is an alarm in each connected Ethiris Client.
Not saved	The camera is not yet saved to Ethiris Server, hence no communication. Notice the small diskette symbol in the lower right corner of the icon. Save the server configuration to establish communication with the camera.

Camera popup menu

Right-clicking such a node brings up a context menu.

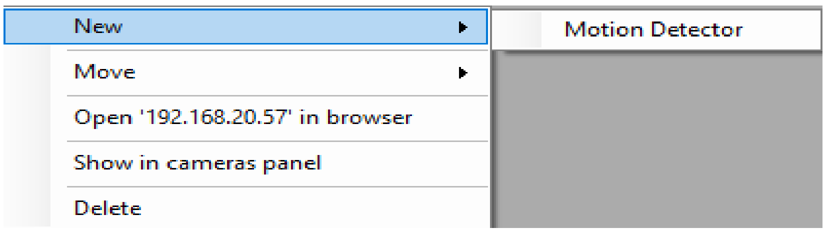


Figure 2.146 The popup menu for a Camera node.

New->Motion Detector adds a new *Motion detection* definition for this camera. A new node in the treeview is immediately created.

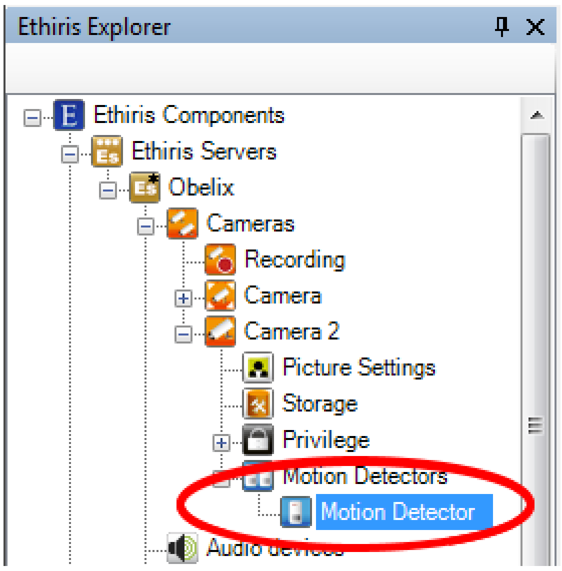


Figure 2.147 New Motion Detector added.

For more information about Motion detection, please see the section *Security* node on page *2:255*.

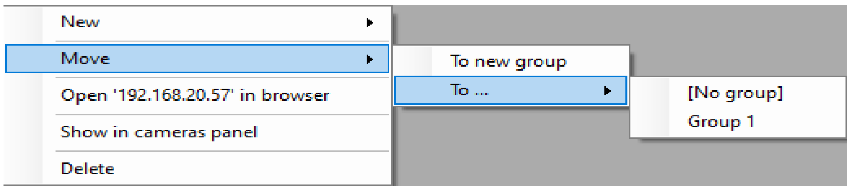


Figure 2.148 The popup menu for a Camera node – Move submenu.

Move->To new group Will create a new camera group and move the camera to that group. Note that any overridden settings in panels *Storage* and *Privilege* on the camera will be reset when the camera is added to a camera group.

Move->To ...->[No group] Will move the camera out of the current group. The camera will no longer belong to any camera group.

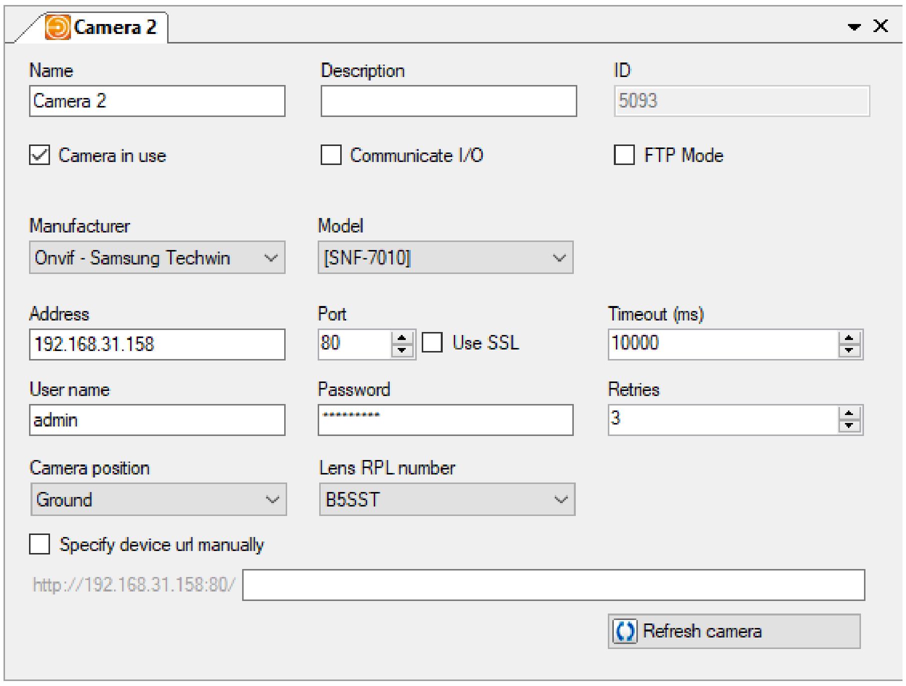
Move->To ...-><Group name> Will move the camera to the specified camera group. A camera can only belong to one group at the time. Note that any overridden settings in panels *Storage* and *Privilege* on the camera will be reset when the camera is added to a camera group.

Open '<IP address>' in browser opens the current IP address in the standard web browser. This way, you can fast and easily enter camera-specific settings that are only accessible via the camera native web interface.

Delete removes the camera from the server configuration. The corresponding node in the treeview is immediately removed. The same goes for the corresponding row in the *Cameras* panel.

Camera panel

Double-clicking a *Camera* node in the treeview opens the corresponding panel.



Name: Camera 2	Description:	ID: 5093
☑ Camera in use	☐ Communicate I/O	☐ FTP Mode
Manufacturer: Onvif - Samsung Techwin	Model: [SNF-7010]	
Address: 192.168.31.158	Port: 80 ☐ Use SSL	Timeout (ms): 10000
User name: admin	Password: *********	Retries: 3
Camera position: Ground	Lens RPL number: B5SST	
☐ Specify device url manually		
http://192.168.31.158:80/		

Figure 2.149 The Camera panel.

In this panel, you can enter general settings for the camera. Some of the fields in this panel are the same as those that are present in the list of cameras in the *Cameras* panel. Depending on the type of camera, more or fewer fields may be visible.

Name is the name of the camera. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the right of the name field indicating the problem.

Description describes the camera. This description is visible in Ethiris Client in the *Cameras* tool window and the *Player*.

ID is automatically generated and cannot be changed. Most of the time, you have no reason to know the ID of the camera. The only time you need to know is if you want to send video via FTP from the camera to Ethiris Server. Read more about this in the *Getting Started with Ethiris* manual.

Camera in use is ticked as default, meaning that the camera is enabled, and Ethiris Server retrieves video from the camera. If not ticked, Ethiris Server will not request video from the camera. In this case, of course, neither live video nor the recording of video is available.

Communicate I/O should be ticked if you want to use the I/O port of the camera. As a default I/O communication is enabled. The setting is only visible if the camera supports I/O.

FTP Mode. When checked, indicates whether the camera is to send video via FTP on its own initiative when an event is detected locally in the camera. In this mode, you can reduce the load on the network during normal use as Ethiris does not retrieve video from the camera if you do not want to see live video from it. However, the camera must be able to detect events in some way when video is to be saved. This mode can also be useful in the situation in which the camera has to connect via a dialed connection.

Delay Transfer is only available for a camera in FTP mode (see above). Indicates how the camera acts when it sends video after having detected an event. It either starts sending video immediately when the event occurs, or it first collects all the video to be saved and then sends them (delayed transfer). This setting only affects the timestamp of the event created in Ethiris when the video start to arrive.

Manufacturer is the manufacturer of the camera. Each manufacturer has a specific set of models available. This column consists of a list of available manufacturers.

Model is the model of the camera. This column consists of a list of available camera models.

Rotation is the rotation of the camera, I.e., how the camera is mounted. This column consists of a list of supported camera rotations. If the camera supports it (i.e., Ethiris can ask the camera for the current rotation setting), the list may contain the item <*Camera Setting*>. The setting is only visible if the camera and Ethiris support the rotation setting.

Address is the IP address for the camera address (or DNS name if this was set up). This address is the base for all communication with the camera.

Port is the port number Ethiris uses for communication with the camera. The port that is usually used is port 80, the default port for HTTP. In some types of cameras, the port number can be changed, but not in all. This setting is used when probing cameras that support that and for PTZ- and I/O-commands. Normally, this setting is not used for retrieving video from the camera.

Timeout indicates how long Ethiris Server is to wait for a response from the camera before we consider that a communication error has occurred and a new connection attempt is made.

User name is the user name to use if the camera requires login.

Password is the password to use if the camera requires login.

Retries indicate how many attempts to connect to the camera are to be made before an error signal is generated.

Camera position denotes a 360-camera position. The setting is a list with three alternatives, *Ceiling, Wall* and *Ground*. Select the setting that corresponds to the mounting of the camera. The setting is only visible for ONVIF-cameras, Generic, Kentima simulator, Axis-cameras, and Samsung 360-cameras that are supported by Ethiris. If you don’t have a 360-camera, the setting will be ignored.

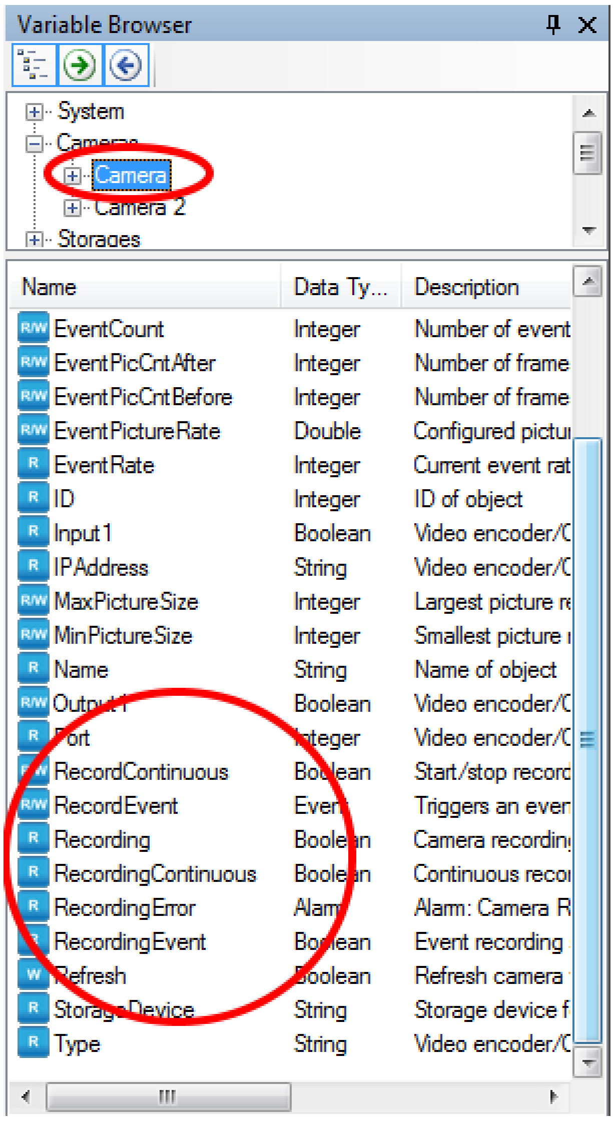
Lens RPL number denotes the type of 360-camera you have. RPL stands for *Registered Panomorph Lens* and tells Immervision which type of lens the camera has. Using this setting, Ethiris will be able to dewarp the image from the camera and also pan, tilt, and zoom the image. If you don’t have a 360-camera, select *Not panomorph* in the list. The setting is only visible for ONVIF-cameras, Generic, Kentima Simulator, and Axis-cameras. Note that it is only 64-bit versions of Ethiris Admin and Ethiris Client that can dewarp the image.

Specify Device url manually This setting will be displayed if the camera is an ONVIF camera. This possibility can be used if discovery does not work for finding the camera. This may happen if the camera is located on another subnet than the computer where Ethiris Server runs.

Camera variables

When defining a *Camera*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See Figure 2.150 for an example where a *Camera* is selected, and the variables belonging to the camera are presented in the lower pane (encircled).



	Name	Data Ty...	Description
R/W	EventCount	Integer	Number of event
R/W	EventPicCntAfter	Integer	Number of frame
R/W	EventPicCntBefore	Integer	Number of frame
R/W	EventPictureRate	Double	Configured pictu
R	EventRate	Integer	Current event rat
R	ID	Integer	ID of object
R	Input1	Boolean	Video encoder/C
R	IPAddress	String	Video encoder/C
R/W	MaxPictureSize	Integer	Largest picture re
R/W	MinPictureSize	Integer	Smallest picture i
R	Name	String	Name of object
R/W	Output1	Boolean	Video encoder/C
R	Port	Integer	Video encoder/C
R/W	RecordContinuous	Boolean	Start/stop record
R/W	RecordEvent	Event	Triggers an ever
R	Recording	Boolean	Camera recordin
R	RecordingContinuous	Boolean	Continuous recor
R	RecordingError	Alarm	Alarm: Camera R
R	RecordingEvent	Boolean	Event recording
W	Refresh	Boolean	Refresh camera
R	StorageDevice	String	Storage device f
R	Type	String	Video encoder/C

Figure 2.150 Variables for a Camera.

We will discuss script and variables later on in this manual, but it seems like a good idea to shortly describe some of the variables right now.

There are a whole bunch of variables for a camera, but some are used more often than others. We will briefly look at the three most important ones.

RecordContinuous is a writable variable, meaning it can be activated via script or e.g., a button in Ethiris Client. When *true,* the camera starts continuous recording and when *false* recording stops (unless the settings in the panel *Camera recording* say that recording should take place).

RecordEvent is a writable variable, meaning it can be activated via script or e.g., a button in Ethiris Client. When *true,* the camera starts event recording. Event recording includes a pre-alarm time and a post-alarm time. For a deeper discussion about recording, see section *Storage* node on *page 2:109*. When RecordEvent is *false,* recording stops (unless the settings in the panel *Camera recording* say that recording should take place).

Recording is a read-only variable that is *true* when recording takes place, both event recording and continuous recording is indicated here. This information can, e.g., be used in Ethiris Client in an LED for displaying when recording for a camera occurs.

RecordingContinuous is a read-only variable that is *true* when continuous recording takes place. This information can be used if you need to make a distinction between continuous and event recording.

RecordingError is a read-only variable that is *true* if the alarm for recording missing is activated. In the panel *Camera Recording,* you can configure Ethiris such as if recording doesn't occur within the time specified, an alarm is automatically activated.

RecordingEvent is a read-only variable that is *true* when event recording takes place. This information can be used if you need to make a distinction between continuous and event recording.

2.4.13 Picture Settings node

Under the *Camera* node in the treeview, there are several nodes for specific settings for a camera. Depending on the type of camera, the type of nodes may vary. There is always a *Picture Settings* node.

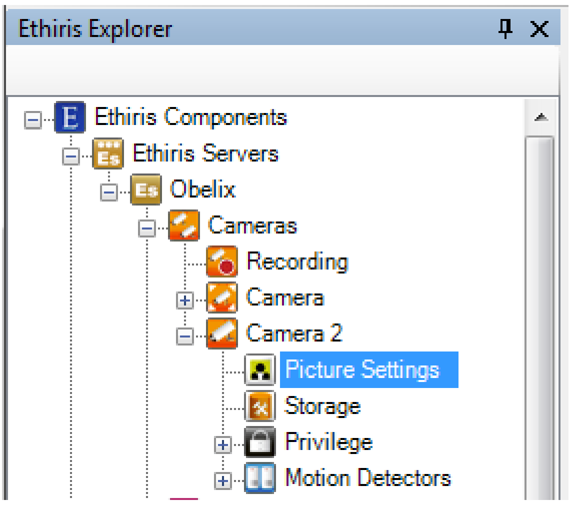


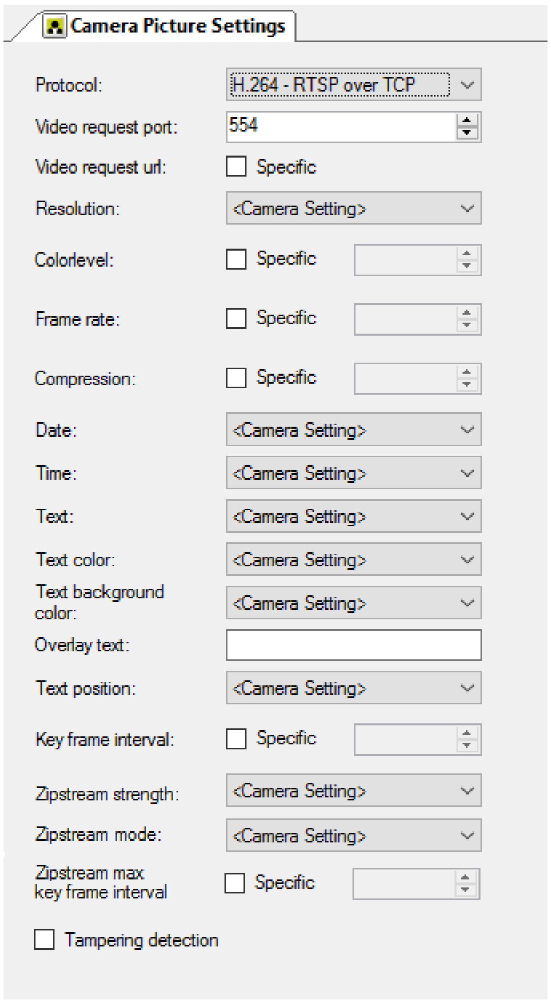
Figure 2.151 The Picture Settings node for a Camera in Ethiris Explorer treeview.

Picture Settings popup menu

There is no popup menu for this node.

Picture Settings panel

Double-clicking a *Picture Settings* node for a camera in the treeview opens the corresponding panel.


Camera Picture Settings

Protocol:	H.264 - RTSP over TCP
Video request port:	554
Video request url:	☐ Specific
Resolution:	<Camera Setting>
Colorlevel:	☐ Specific
Frame rate:	☐ Specific
Compression:	☐ Specific
Date:	<Camera Setting>
Time:	<Camera Setting>
Text:	<Camera Setting>
Text color:	<Camera Setting>
Text background color:	<Camera Setting>
Overlay text:	
Text position:	<Camera Setting>
Key frame interval:	☐ Specific
Zipstream strength:	<Camera Setting>
Zipstream mode:	<Camera Setting>
Zipstream max key frame interval	☐ Specific

☐ Tampering detection

Figure 2.152 The Camera Picture Settings panel.

In this panel, you can enter settings regarding the video from the camera. Some of the fields in this panel are the same as those that are present in the list of cameras in the *Cameras* panel. Depending on the type of camera, more or fewer fields may be visible.

Protocol determines what video compression and video transport methods to use. Different camera models may have different available protocols. This column consists of a list of available combinations of video compression methods and transport protocols.

Video request port. When communicating over certain protocols, the video request port might differ from the standard command port. The camera's default video request port is automatically entered when the protocol is selected, but this value can be exchanged to another port. In most cases, it is not necessary to change the port number.

Receive port. This setting is only available if UDP is used as the transport protocol. If *Specific* is not checked, Ethiris Server selects a random port number for receiving the video. This works in most cases. However, if a firewall has to be set up to enable incoming connections, the port number may have to be specified. The number has to be an even number between 50002 – 65534.

Video request url Specific should be checked if you want to manually, instead of letting Ethiris Server automatically, create the string to send to the camera for requesting video. If this box is not checked, Ethiris Server will automatically create the string based on the settings entered for the camera. If it is checked, the other settings for the camera will be ignored and the string entered here will be used by Ethiris Server when requesting video from the camera.

Resolution determines what resolution to use. Different camera models may have different available resolutions. This field consists of a list of available resolutions. If you select *<Camera Setting>*, the resolution is determined by the settings directly in the camera. In this case, if you change the resolution in the camera, you will also affect the video sent to Ethiris. If you, on the other hand, specifically select a resolution in the Ethiris server configuration, this resolution will always be used by Ethiris when requesting video from the camera regardless of the setting in the camera itself.

Color level indicates how much color information there should be in the video retrieved from the camera. If, for example, you enter 0, you get a black and white video that takes up very little space, while higher values produce a video with richer shades of color that require more storage space. If you click *Specific*, you can select a suitable value. Otherwise, the settings entered locally in the camera apply.

Frame rate indicates the speed of the flow of frames retrieved from the camera. If you click *Specific*, you can select a suitable value. Otherwise, the settings entered locally in the camera apply. Please note that you have to ensure that this value is not lower than the frame rate required for the recording settings selected in the *Storage* panel for the camera.

Compression indicates the compression you want for the frame information in the video retrieved from the camera. Higher compression produces poorer video quality, but the video takes up less storage space. If you click *Specific*, you can select a suitable value. Otherwise, the settings entered locally in the camera apply.

Quality. For some camera models, you can select a setting for quality instead of compression. The camera model determines which values are available. Higher values produce better video quality.

Date indicates whether the current date (from the camera) is to be displayed in the frame. You can select *Off*, *On,* or *<Camera setting>*, in which case the settings entered locally in the camera are used.

Time indicates whether the current time (from the camera) is to be displayed in the frame. You can select *Off*, *On,* or *<Camera setting>*, in which case the settings entered locally in the camera are used.

Text indicates whether any text string defined locally in the camera is to be displayed in the frame. You can select *Off* or *On***.** You can also select *<Camera setting>*, in which case the settings entered locally in the camera are used.

Text color specifies the color of the overlay text in the camera image. The available choices are specified by the camera model.

Text background color specifies the background color of the overlay text in the camera image. The available choices are specified by the camera model.

Text size specifies the size of the overlay text in the camera image. The available choices are specified by the camera model.

Overlay text specifies the actual text you want the camera to overlay on the image, possibly together with *Date* and *Time*. For the text to be displayed, the setting *Text* should be set to *On.*

Text position specifies the position in the picture where the overlay text should be located. The available choices are specified by the camera model.

Key frame interval specifies how often the camera should send key-frames in the video stream. This setting only affects H264 or MPEG4 video streams. Key-frames require much more storage space/bandwidth than other frames. Therefore you don't want to send them more often than necessary. If you check Specific you can enter a suitable value.

Zipstream strength specifies how hard the Axis Zipstream algorithm should work to reduce the bandwidth from the camera. The available choices are specified by the camera model. Only available for certain Axis cameras.

Zipstream mode specifies if the camera may use dynamic gop-length or not. Dynamic gop-length can reduce bandwidth further. Only available for certain Axis cameras.

Zipstream max key frame interval specifies the maximum gop-length if the parameter *Zipstream mode* is set to *Dynamic*. Only available for certain Axis cameras.

Tampering detection can be used to detect possible tampering attempts to the camera and also if the video sent from the camera is too bright or dark to be useful. This is a built-in function in Ethiris that can be activated for desired cameras. The detection is executed every 5th second. Several signals are automatically generated in the Ethiris data store and can be used in Script. The four signals that are most likely used are *Tampering*, *FullFrameMovement*, *BrightImage,* and *DarkImage*. These signals can be monitored and used for activating other functions in Ethiris, like sending an SMS to notify an operator of tampering.

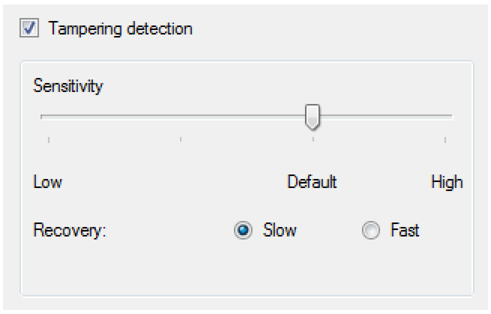
Tampering is activated if the camera is covered with, e.g., a plastic bag or if it loses focus.

FullFrameMovement is activated if the camera is moved out of position, i.e., if the whole image is moved.

BrightImage is activated if the video from the camera is so bright that it can't be used for tampering analysis.

DarkImage is activated if the video from the camera is so dark that it can't be used for tampering analysis.

When *Tampering detection* is activated, some additional settings appear in the panel.


[✓] Tampering detection

Sensitivity: Default (Low – Default – High)

Recovery: [✓ Slow] Fast

Figure 2.153 Tampering settings in the Camera Picture Settings panel.

Sensitivity can be set to four different values. The most sensitive (High) value means that only a small portion of the camera has to be covered to activate the *Tampering* signal. When set to *Low*, most of the camera has to be covered for activating the *Tampering* signal. The *Sensitivity* signal itself can be altered via Script as well as in this panel. You can, for example, change the sensitivity value based on a schedule.

Recovery. When the *Tampering* signal has been activated, you can choose to let the new circumstances be the normal situation, or you can choose to keep the circumstances before Tampering was detected to be the normal. This adjustment can be set by selecting *Slow* or *Fast* recovery. *Slow* recovery means that it takes between 30 seconds and an hour before the new circumstances are regarded as normal, while *Fast* recovery means that it takes max 20 seconds. The time for slow recovery varies depending on the amount of tampering.

2.4.14 Storage node

Another node under the Camera node in the treeview is the *Storage* node.

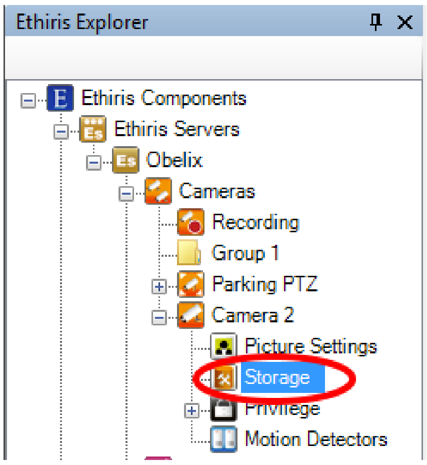


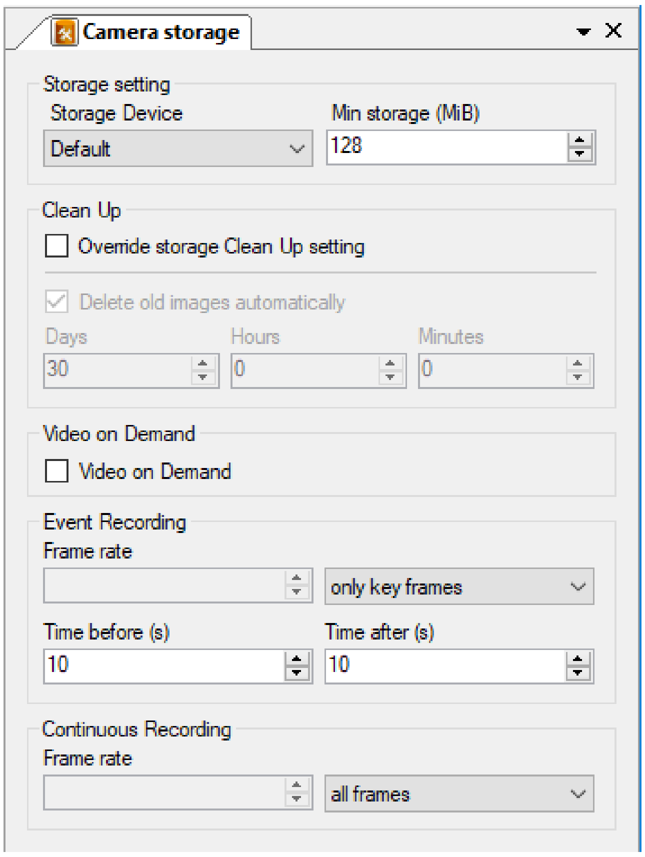
Figure 2.154 The Storage node for a Camera in Ethiris Explorer treeview.

Storage popup menu

There is no popup menu for this node.

Storage panel

Double-clicking a *Storage* node for a Camera in the treeview opens the corresponding panel.



Camera storage

Storage setting
Storage Device: Default
Min storage (MiB): 128

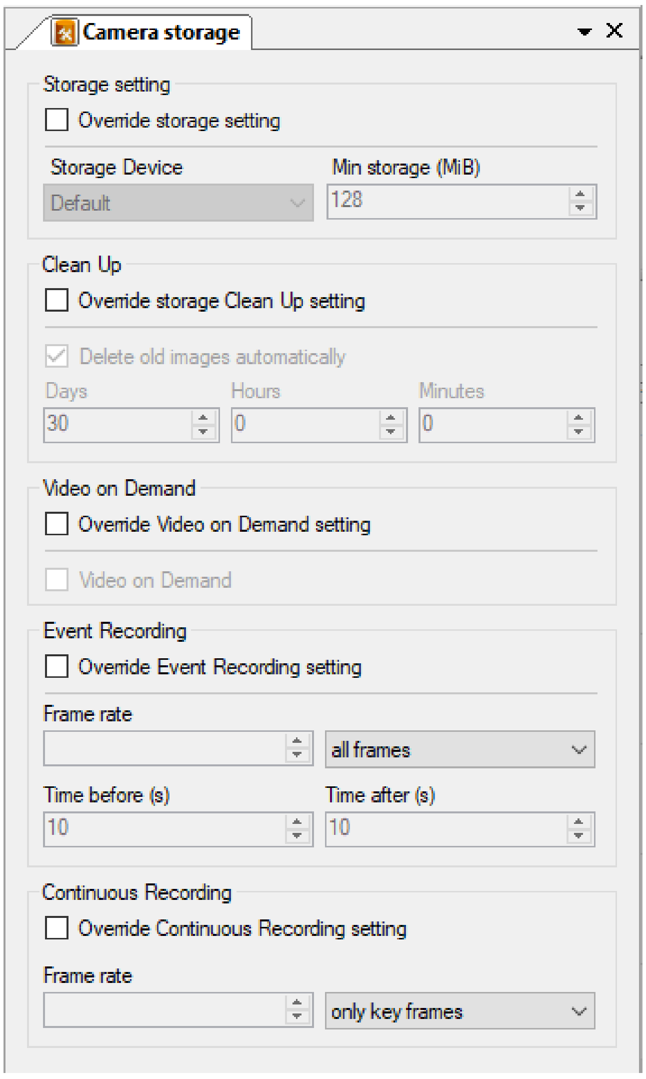
Clean Up
☐ Override storage Clean Up setting
✓ Delete old images automatically
Days: 30
Hours: 0
Minutes: 0

Video on Demand
☐ Video on Demand

Event Recording
Frame rate: only key frames
Time before (s): 10
Time after (s): 10

Continuous Recording
Frame rate: all frames

Figure 2.155 The Camera Storage panel for a camera, not a member of a group.



Camera storage

Storage setting

☐ Override storage setting

Storage Device	Min storage (MiB)
Default	128

Clean Up

☐ Override storage Clean Up setting

✓ Delete old images automatically

Days	Hours	Minutes
30	0	0

Video on Demand

☐ Override Video on Demand setting

☐ Video on Demand

Event Recording

☐ Override Event Recording setting

Frame rate: all frames

Time before (s)	Time after (s)
10	10

Continuous Recording

☐ Override Continuous Recording setting

Frame rate: only key frames

Figure 2.156 The Camera Storage panel for a camera member of a group.

In this panel, you can enter recording settings for the camera.

Storage setting

Override storage setting is only visible if the camera is a member of a camera group. Check if you want to override group storage settings for this camera.

Storage Device indicates which of the defined storage devices the camera will use to save video. Storage devices are defined in the *Storages* panel for the Ethiris Server.

Min storage (MiB) indicates how much hard disk space must always be protected for this camera’s video. When the storage space for video is full, Ethiris automatically removes the oldest video to make room for new video. If one camera in the system records a high volume of video, there is a risk that video from other cameras will be deleted. Using this setting, you can protect part of the video for each camera.

Clean Up

Override storage Clean Up setting should be checked if you want to change the clean-up settings for this camera. Default, for cameras not a member of a camera group, is to obey the general settings entered in the common panel for the whole Ethiris Server, *Storages*. For cameras that are members of a camera group, default is to follow group clean-up settings.

Delete old images automatically. Check this box if you want the recorded video to be removed from the hard disk automatically when it gets too old.

Days, Hours *&* **Minutes** defines which video is to be removed. All recorded video older than specified here are removed from the hard disk.

Video on demand should be checked if you don't want Ethiris Server to retrieve images from the camera all the time, but only when it is required. I.e., when an Ethiris Client wants to show live images from the camera or when a recording condition is true.

Note that if this choice is made, the *Time before* is set to 0 for Event Recording.

Event recording

Override Event Recording setting is only visible if the camera is a member of a camera group. Check if you want to override group settings for event recording for this camera*.*

Frame Rate defines the frame rate in frames per second (or what time unit is selected) when video is stored for an event. Ensure that this value does not conflict with the frame rate required from the camera entered in the *Picture Settings* panel.

Time before defines a period of time in seconds for which video is to be recorded before an event has occurred. This setting also denotes maximal Instant replay time, i.e., how many seconds back in time Instant replay can display.

Time after defines the number of seconds for which video is to be recorded after an event has occurred (after the condition for event recording is no longer *true*).

Continuous Recording

Override Continuous Recording setting is only visible if the camera is a member of a camera group. Check if you want to override group settings for continuous recording for this camera*.*

Frame rate indicates the frame rate for the storage of continuously recorded video. Ensure that this value does not conflict with the frame rate required from the camera entered in the *Picture Settings* panel.

2.4.15 Privilege node

Under the Camera node in the treeview is also the *Privilege* node.

A number of specific user operations are defined in Ethiris. For each such user operation, you can, if you want, specify that the user who performs the operation must be a member of a certain user group in the Windows user system. This may be either a local user group on the computer running Ethiris Server or a global user group in a domain or the Active Directory if the computers and users involved are members of a domain. To be able to specify a user group in the domain, it is necessary for both the computer running Ethiris Server to be a member of the domain and the logon entered by the user to be for an account in the same domain.

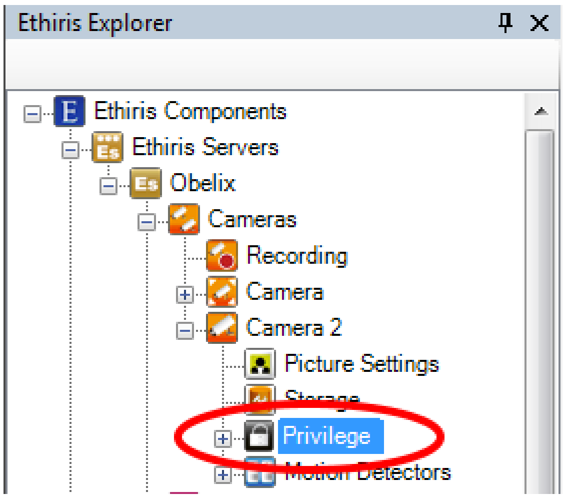


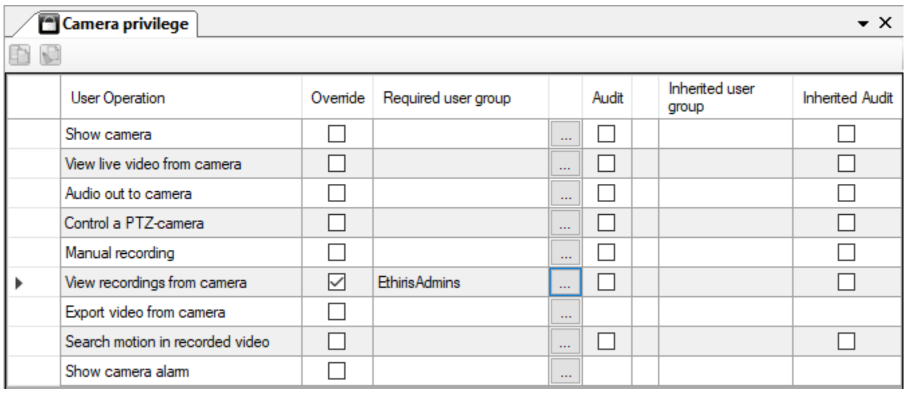
Figure 2.157 The Privilege node for a Camera in Ethiris Explorer treeview.

Privilege popup menu

There is no popup menu for this node.

Privilege panel

Double-clicking a *Privilege* node for a Camera in the treeview opens the corresponding panel.



Camera privilege

	User Operation	Override	Required user group		Audit	Inherited user group	Inherited Audit
	Show camera	☐		...	☐		☐
	View live video from camera	☐		...	☐		☐
	Audio out to camera	☐		...	☐		☐
	Control a PTZ-camera	☐		...	☐		☐
	Manual recording	☐		...	☐		☐
▶	View recordings from camera	☑	EthirisAdmins	...	☐		☐
	Export video from camera	☐		...			
	Search motion in recorded video	☐		...	☐		☐
	Show camera alarm	☐		...			

Figure 2.158 The Camera Privilege panel.

In this panel, you can enter settings for the camera regarding access control. The eight user operations listed here can be set at the Ethiris Server level (valid for all cameras) at the camera group level (valid for all cameras in the group) and specifically for each camera. In this panel, you set access control parameters for this camera specifically.

As a default, no log in is required. Every user has access to all functions. If you want to restrict access to a certain user operation, you have to specify which Windows user group the user has to be a member of to get access to the operation. To gain access, the user has to log in as a Windows user that is a member of the required Windows user group.

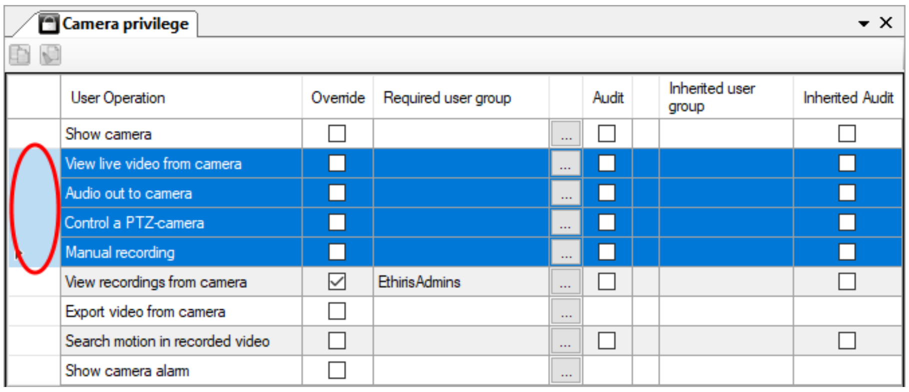
At the top of the panel, there is a toolbar.

Privilege panel toolbar

Figure 2.159 The toolbar in the Cameras Privilege panel.

Copy. Use this button to copy the content of a specific row in the *User Operations* list.

Paste Use this button to paste the settings from the copied operation to one or several other operations. Select one or several operations by clicking in the area to the left of the *User Operation* column. See *Figure 2.160* below. Use the Ctrl or Shift keys on the keyboard together with the mouse to select several rows. You can also click and drag over several rows to select them.



	User Operation	Override	Required user group		Audit	Inherited user group	Inherited Audit
	Show camera	☐		...	☐		☐
	View live video from camera	☐		...	☐		☐
	Audio out to camera	☐		...	☐		☐
	Control a PTZ-camera	☐		...	☐		☐
	Manual recording	☐		...	☐		☐
	View recordings from camera	☑	EthirisAdmins	...	☐		☐
	Export video from camera	☐		...			
	Search motion in recorded video	☐		...	☐		☐
	Show camera alarm	☐		...			

Figure 2.160 Four rows selected.

The following columns are part of the *Operations* list in the Camera Privilege panel:

User Operation lists the available operations that you can set access control requirements for on a camera. These are:

Show camera – Makes the camera disappear completely in Ethiris Client unless a user that belongs to the specified user group is logged in.

View live video from camera – Restricts live video from this camera. In Ethiris Client, the message *Not authorized!* is displayed in the camera view if not, a user that is a member of the required group is logged in. See *Figure 2.161*.

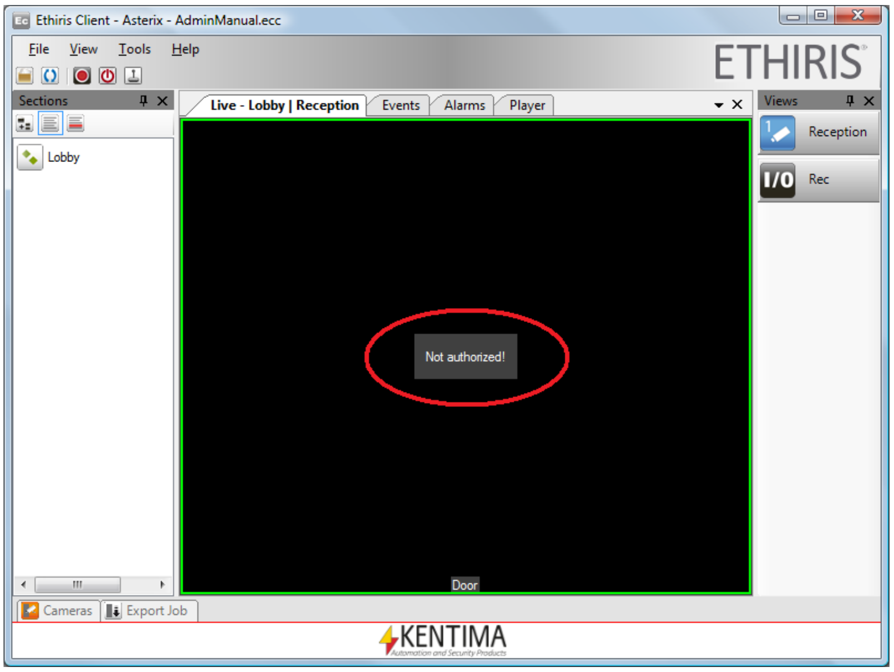


Figure 2.161 No one logged in when View live video from camera requires log in.

Audio out to camera – Gives access to the audio out on all cameras connected to this camera in the panel *Connect audio device to camera.*

Control a PTZ-camera – Restricts all optical PTZ operations, i.e., you cannot move or zoom the camera if not logged in as a member of the required group. Digital PTZ still works.

Manual recording – Restricts manual recording via the recording button in Ethiris Client. Clicking the recording button when not logged in simply has no effect. No recording is done.

View recordings from camera - Restricts recorded video from this camera. In Ethiris Client, the message *Not authorized!* is displayed in the camera view in the Player if not a user that is a member of the required group is logged in.

Export video from camera – Restricts all kind of export. If not logged in as a member of the required group, there will be no *Export* menu items in the various popup menus. If you click the Export button in the Player, only authorized cameras will appear in the camera list in the *Export dialog*.

Search motion in recorded video – Restricts motion search in the Client's Player. If not logged in as a member of the required group, the message *Can not search motion right now!* will be displayed when you select the menu item *Search next Motion* in the popup menu for the camera in the Player.

Show camera alarm – Restricts visibility of alarms and alarm events related to alarms on this camera unless you are logged in as a member of the required group.

OK, back to the columns in the *Camera Privilege* operations list:

Override has to be explicitly checked to be able to change the *Required User Group*. A privilege setting can be set at the Ethiris Server level or camera group level (if the camera is a member of a camera group), meaning that all cameras have the same security settings unless they are specifically overridden in the *Camera Security* panel by checking *Override*.

Required user group specifies what user group the user has to be a member of to access this function. A blank field means that no log in is required. You can browse for available user groups by clicking the browse button to the right of this column.

Audit can be checked if you want to log when a certain operation is executed. For each operation for which *Audit* is specified, the system will save information on the time, the operation performed, who performed it, the client computer from which it was performed, and any other available information, depending on the type of operation performed.

It is permitted to specify that the system must log an operation without, at the same time, making any requirement that the user must be a member of a specific group. The operation will be logged regardless of this, but in these cases, there may be no information on who performed the operation if the user has not logged on. Other available information is logged as usual.

The audit log can be viewed in the *Events* panel in Ethiris Client.

Inherited user group is just information on the current setting on the Ethiris Server level (or camera group level if the camera is a member of a camera group). To override this setting, check the *Override* checkbox.

Inherited Audit is just information on the current setting on the Ethiris Server level. To override this setting, check the *Override* checkbox.

Inherited user group and *Inherited Audit* works according to the following rules for this panel:

- If the camera is a member of a camera group, the camera inherits the Required user group and Audit from the *Privilege* of the group.
- Alternatively, and if the camera is not a member of a camera group, the camera inherits the corresponding properties from the *Privilege* on the Ethiris Server level.

2.4.16 Client type Privilege node

Under the node Security in the treeview, there is a *Client type Privilege* node.

The purpose of this node is to provide the opportunity to enter specific security settings for different types of clients. Today you can enter settings for *Ethiris Client, Ethiris Admin, Ethiris Mobile,* and *WideQuick EthirisView,* respectively.

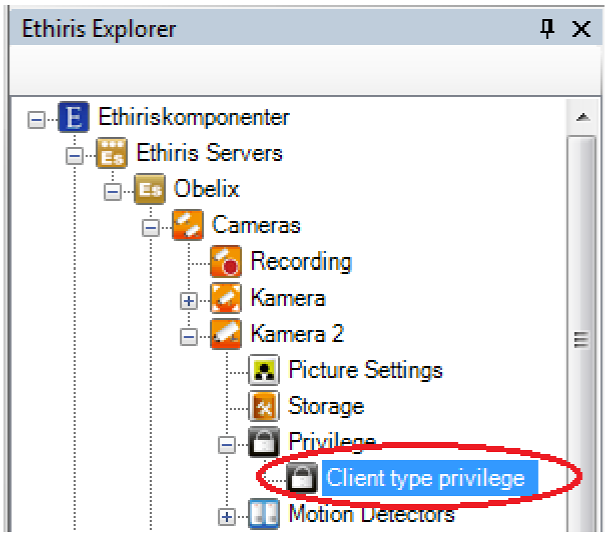


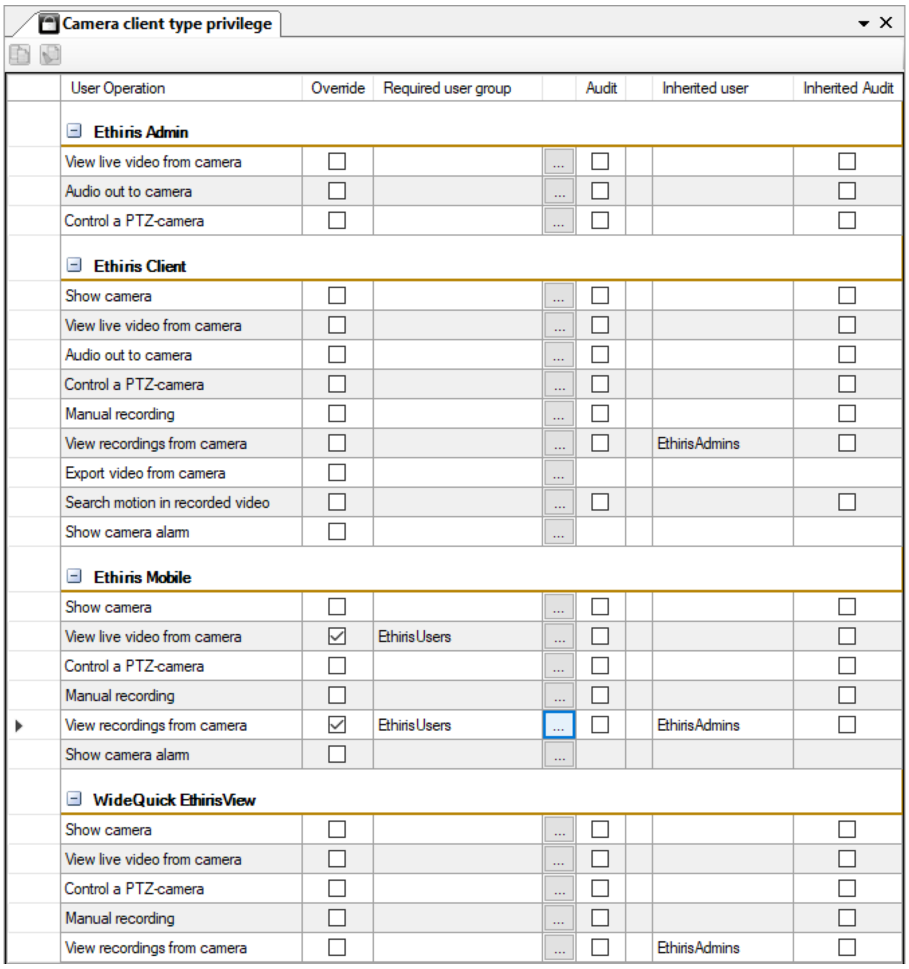
Figure 2.162 The node Client type Privilege for a camera in the Ethiris Explorer treeview.

Camera Client type Privilege popup menu

There is no popup menu for this node.

Camera Client type Privilege panel

Double-click on a *Client type Privilege* node for a camera in the treeview will open the corresponding panel.



User Operation	Override	Required user group	Audit	Inherited user	Inherited Audit
Ethiris Admin					
View live video from camera	☐		☐		☐
Audio out to camera	☐		☐		☐
Control a PTZ-camera	☐		☐		☐
Ethiris Client					
Show camera	☐		☐		☐
View live video from camera	☐		☐		☐
Audio out to camera	☐		☐		☐
Control a PTZ-camera	☐		☐		☐
Manual recording	☐		☐		☐
View recordings from camera	☐		☐	EthirisAdmins	☐
Export video from camera	☐				
Search motion in recorded video	☐		☐		☐
Show camera alarm	☐				
Ethiris Mobile					
Show camera	☐		☐		☐
View live video from camera	☑	EthirisUsers	☐		☐
Control a PTZ-camera	☐		☐		☐
Manual recording	☐		☐		☐
View recordings from camera	☑	EthirisUsers	☐	EthirisAdmins	☐
Show camera alarm	☐				
WideQuick EthirisView					
Show camera	☐		☐		☐
View live video from camera	☐		☐		☐
Control a PTZ-camera	☐		☐		☐
Manual recording	☐		☐		☐
View recordings from camera	☐		☐	EthirisAdmins	☐

Figure 2.163 The panel Client type Privilege for a camera.

In the list, there are the same columns and the same operations as for camera security. The difference is that the operations are divided between the different types of clients; *Ethiris Admin*, *Ethiris Client (regular Ethiris Client)*, *Ethiris Mobile (the mobile app Ethiris Mobile),* and *WideQuick EthirisView*. See the last section for an explanation of the various columns and operations.

For each type of client, the relevant operations are listed.

Inherited user group and *Inherited Audit* works according to the following rules for this panel:

- In the first place, the camera will inherit the required user group and audit defined under *Privilege* on the camera itself.
- In the second place, if the camera is a member of a camera group, the corresponding properties are inherited from *Group client type privilege* on the group.
- In third place, if the camera is a member of a camera group, the corresponding properties are inherited from *Privilege* on the group.
- In fourth place, or if the camera is not a member of a camera group, the corresponding properties are inherited from *Client type privilege* on the Ethiris Server level.
- In the last place, the corresponding properties are inherited from *Privilege* on the Ethiris Server level.

2.4.17 PTZ node

For so-called PTZ cameras, there is a *PTZ* node under the Camera node in the treeview.

The purpose of this node is to limit the PTZ camera’s ability to move. Perhaps you are currently permitted only to film part of the camera’s range.

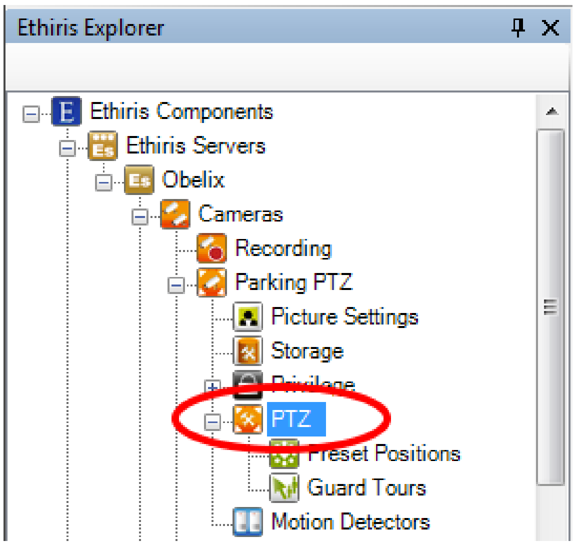


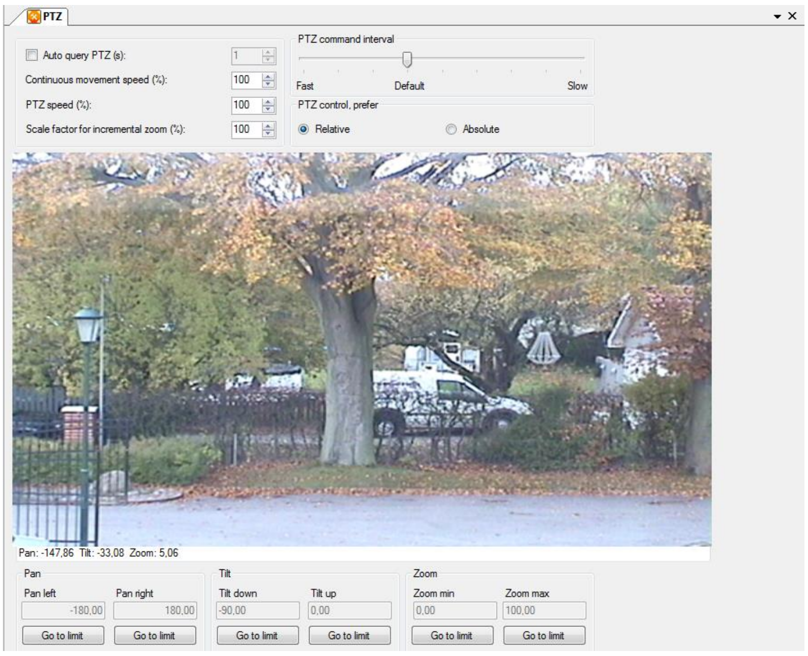
Figure 2.164 The PTZ node for a camera in the Ethiris Explorer treeview.

PTZ popup menu

There is no popup menu for this node.

PTZ panel

Double-clicking a *PTZ* node for a camera in the treeview opens the corresponding panel.



Auto query PTZ (s): 1
Continuous movement speed (%): 100
PTZ speed (%): 100
Scale factor for incremental zoom (%): 100

PTZ command interval: Fast — Default — Slow

PTZ control, prefer: [✓ Relative] Absolute

Pan: -147,86 Tilt: -33,08 Zoom: 5,06

Pan		Tilt		Zoom	
Pan left	Pan right	Tilt down	Tilt up	Zoom min	Zoom max
-180,00	180,00	-90,00	0,00	0,00	100,00

Figure 2.165 The camera PTZ panel.

In this panel, you can limit the range of the camera. You can control the camera in the normal manner with the mouse.

The following settings can be found in this panel:

Auto query PTZ indicates in seconds how often the system is to query the camera for its current position. This function is that disabled since Ethiris automatically fetches the current position after moving the camera.

Continuous movement speed indicates in % how fast the camera is to move in relation to its maximum speed for continuous control. 100 indicate full speed.

PTZ speed indicates in % how fast the PTZ camera is to move to a new position. This may be a pre-set position or just a command from the client concerning, for example, centering the frame.

Scale factor for incremental zoom determines in % how large steps the camera will zoom relative to the normal zoom step. There are two ways to zoom stepwise in Ethiris. Imagine that the whole zoom range is between
0 – 10 000 where 0 is completely zoomed out, and 10 000 is completely zoomed in. The two ways for zooming have the following normal zoom steps:

One notch on the mouse scroll wheel: *240*.

+/- keys on the keyboard: *60*.

This means that for zooming in from completely zoomed out to completely zoomed in via the *Zoom in* button in the control panel, you have to click 20 times (10 000/500).

Normally the scale factor for incremental zoom is set to 100, which means that the zoom steps above are used. But you can both increase and decrease the scale factor. E.g., a scale factor of *20* means that, e.g., a step of the mouse wheel corresponds to a zoom step of *48*.

PTZ command interval specifies how often Ethiris may send PTZ commands to the camera. Some cameras can only handle a few PTZ commands per second, while others can handle 10-20 or more commands per second. The more commands per second Ethiris can send, the more responsive the camera will appear.

PTZ control, prefer relative or **absolute** positioning of the camera. The standard setting is to prefer relative positioning. Some cameras can not handle relative positioning very well, so here we can instruct Ethiris to use absolute positioning instead.

These settings are common for the camera. Now we have some settings for Pan, Tilt & Zoom limits. Depending on the capabilities of the camera, some buttons may be disabled.

Go to limit simply means the camera is moved to the corresponding limit position. This can be used to verify that the PTZ control of the camera is working as expected.

2.4.18 Preset Positions node

Under the PTZ node of a camera in the treeview is a *Preset Positions* node.

The purpose of this node is to create *Preset positions*. An unlimited number can be created for each camera. The positions can then be used in tour lists or as they are for manual and automatic movement to individual positions.

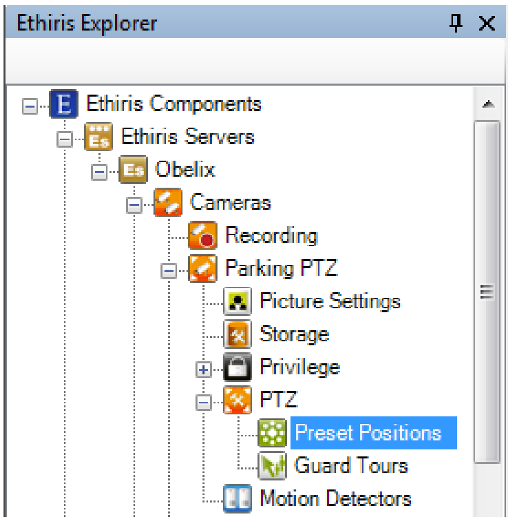


Figure 2.166 The Preset Positions node for a camera in Ethiris Explorer treeview.

Preset Positions popup menu

Right-clicking such a node brings up a context menu.

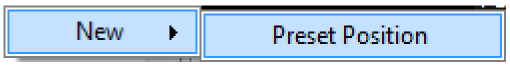
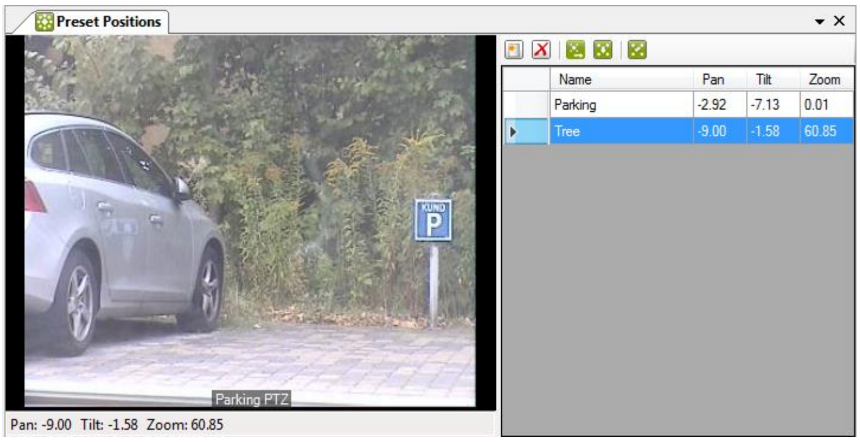


Figure 2.167 The popup menu for a Preset Positions node.

New->Preset Position adds a new *Preset position* definition for this camera. It has the same effect as clicking the *Add a new Preset Position* button in the toolbar. A new row is created in the preset positions list in the Preset positions panel.

Preset Positions panel

Double-clicking a *Preset Positions* node for a Camera in the treeview opens the corresponding panel.



Name	Pan	Tilt	Zoom
Parking	-2.92	-7.13	0.01
Tree	-9.00	-1.58	60.85

Figure 2.168 The Camera PTZ Preset Positions panel.

In this panel, you can add, delete, and change existing *preset positions*. In the camera view to the left, you can operate the PTZ camera via the mouse.

At the top of the panel to the right, there is a toolbar.

Preset Positions panel toolbar

Figure 2.169 The toolbar in the Camera PTZ Preset Positions panel.

Add new preset position.

Use this button to create a new preset position. A new row is added to the list.

Delete preset position

Use this button to delete the selected preset position(s). Mark (un)desired positions by clicking in the column to the left of the *Name* column.

Go to preset position

Mark a preset position in the list and then click this button to move the camera to the position.

Use current position

Click this button to copy the current Pan, Tilt & Zoom values from the camera to the selected preset position.

Toggle use of the unit's internal management for all preset positions

As a default, Ethiris Server handles preset positions and store Pan, Tilt & Zoom values for each preset position. When a preset position is activated, Ethiris Server sends the Pan, Tilt & Zoom values to the camera/video encoder.

However, some cameras and video encoders handle preset positions internally in the unit itself. In this case, Ethiris Server only sends the name of the preset position to the camera/video encoder, which then knows how to position the camera.

When preset positions should be handled by the unit itself, the Pan, Tilt & Zoom columns disappear from the list. In this case, the names of the preset positions have to correspond between the list in Ethiris Admin, and the preset positions entered directly in the camera/video encoder.

The following columns are part of the *Preset positions* list in the Camera Preset positions panel:

Name is the desired name of the preset position. This name is used to identify the position in various contexts. E g in Ethiris Client, when right-clicking a camera view, a list of available preset positions are presented in the popup menu.

Pan is the pan value stored for this position. It can be changed manually directly in the cell, but most often it is automatically read from the current camera PTZ position

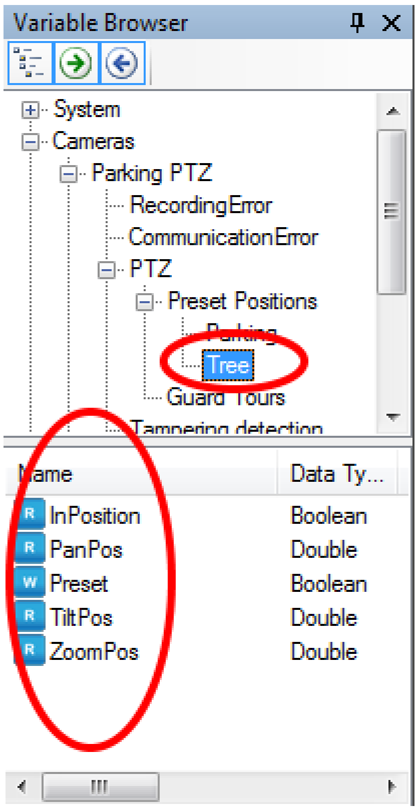
Tilt is the tilt value stored for this position. It can be changed manually directly in the cell, but most often, it is automatically read from the current camera PTZ position.

Zoom is the zoom value stored for this position. It can be changed manually directly in the cell, but most often, it is automatically read from the current camera PTZ position.

Camera PTZ Preset Positions variables

When defining a *Preset position* for a camera, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See Figure 2.170 for an example where a *Preset position* is selected, and the variables belonging to the preset position are presented in the lower pane (ringed in).



Name	Data Ty...
InPosition	Boolean
PanPos	Double
Preset	Boolean
TiltPos	Double
ZoomPos	Double

Figure 2.170 Variables for a Preset position.

We will discuss script and variables later on in this manual, but it seems like a good idea to shortly describe the variables right now.

InPosition is a read-only variable. It is *true* when the camera is in the preset position.

PanPos is a read-only variable. It is the current pan position of the camera.

Preset is a writable variable, meaning it can be activated via script or e.g., a button in Ethiris Client. When *true,* the camera goes to the preset position.

TiltPos is a read-only variable. It is the current tilt position of the camera.

ZoomPos is a read-only variable. It is the current zoom position of the camera.

2.4.19 Guard Tours node

Under the PTZ node of a camera in the treeview is a *Guard Tours* node.

The purpose of this node is to create *Guard Tours*, i.e., lists of preset positions. An unlimited number can be created for each camera. The lists can then be activated either automatically via script or, e.g. by a button in Ethiris Client.

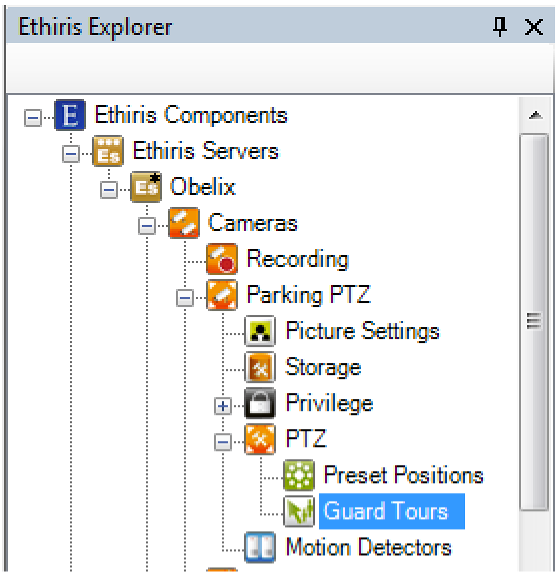


Figure 2.171 The Guard Tours node for a camera in Ethiris Explorer treeview.

Guard Tours popup menu

Right-clicking such a node brings up a context menu.

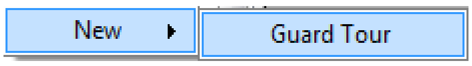


Figure 2.172 The popup menu for a Guard Tours node.

New->Guard Tour adds a new *Guard tour* definition for this camera. It has the same effect as clicking the *Add a new Guard Tour* button in the toolbar in the Guard Tours panel. A new row is created in the guard tours list in the Guard Tours panel.

Guard Tours panel

Double-clicking a *Guard Tours* node for a camera in the treeview opens the corresponding panel.

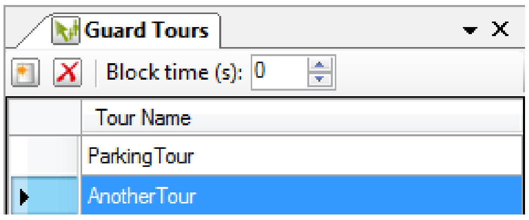


Figure 2.173 The Camera PTZ Guard Tours panel.

In this panel, you can add, delete, and change the name of existing *guard tours*.

At the top of the panel, there is a toolbar.

Guard Tours panel toolbar

Figure 2.174 The toolbar in the Guard Tours panel.

Add new guard tour.

Use this button to create a new guard tour. A new row is added to the list, and a new node is created in the treeview. To enter preset positions in the list, you have to double-click the new guard tour node in the treeview to open the guard tour panel.

Delete guard tour	Use this button to delete the selected guard tour(s). Mark (un)desired tours by clicking in the column to the left of the *Name* column.
Block time	As default *Block time* is set to 0. This means that active tours run even if someone tries to control the camera manually. If *Block time* is > 0, it means that if someone controls the camera manually from Ethiris Client, the tour is paused for the number of seconds specified in Block time.

The following columns are part of the *Guard Tours* list in the Camera Guard Tours panel:

Tour Name is the desired name of the guard tour. This name is used to identify the tour in various contexts.

2.4.20 Guard Tour node

Under the PTZ Guard Tours node of a camera in the treeview, there are possibly one or several *Guard Tour* nodes.

The purpose of these nodes is to enter *Preset positions* for the Guard Tours and possibly set the time for each position.

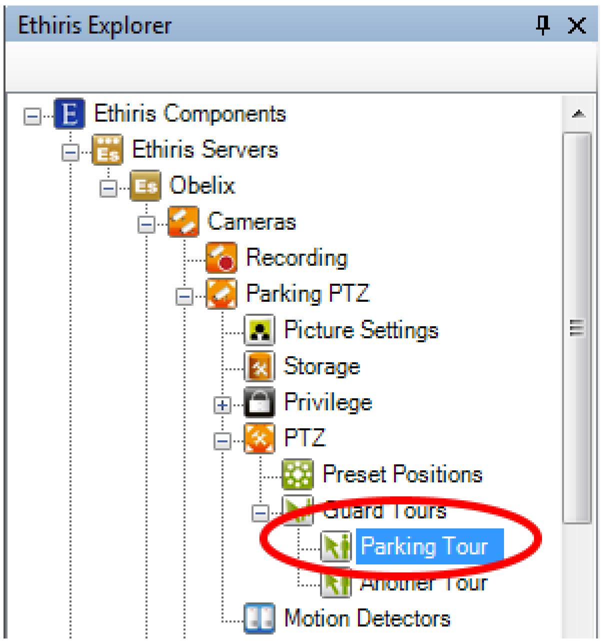


Figure 2.175 The Guard Tour node for a camera in Ethiris Explorer treeview.

Guard Tour popup menu

Right-clicking such a node brings up a context menu.

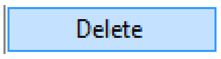
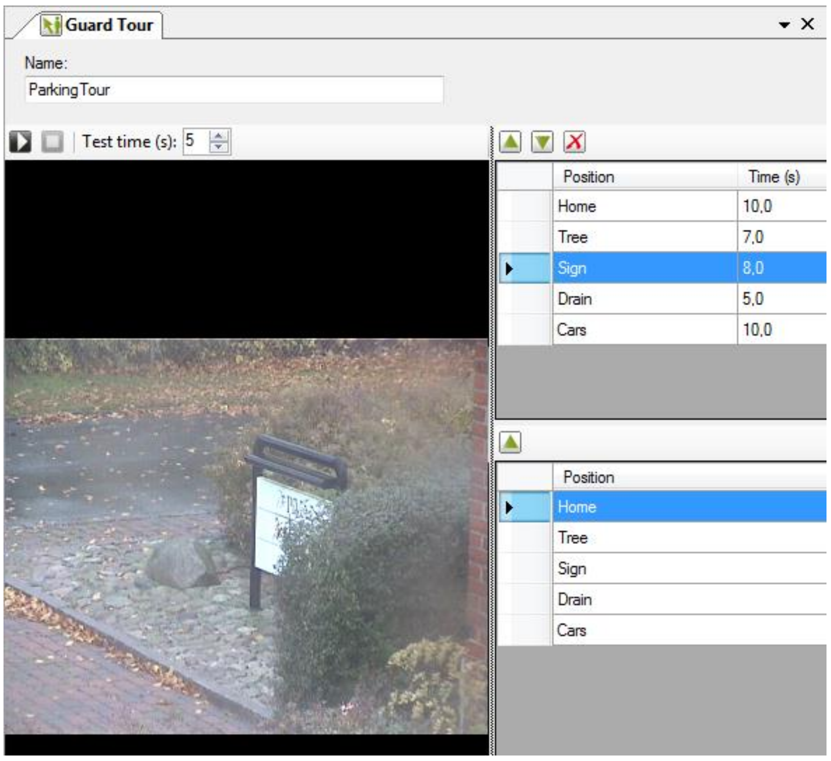


Figure 2.176 The popup menu for a Guard Tour node.

Delete removes the guard tour from the server's configuration. Selecting this menu item has the same effect as clicking the *Delete guard tour* button in the toolbar of the *Guard Tours* panel described above.

Camera PTZ Guard Tour panel

Double-clicking a *Guard Tour* node for a camera in the treeview opens the corresponding panel.



Name:
ParkingTour

	Position	Time (s)
	Home	10,0
	Tree	7,0
▶	Sign	8,0
	Drain	5,0
	Cars	10,0

Figure 2.177 The Camera PTZ Guard Tour panel.

In this panel, you can add, delete, and change existing *preset positions* in the guard tour list. In the camera view to the left, you can operate the PTZ camera via the mouse, but the reason why it's there is to be able to test the tour and see what is happening.

To the right in the panel, there are two lists. The upper one is the actual preset position list of the tour. The lower list is a list of available preset positions for this camera.

In this panel, there are several toolbars.

Guard Tour panel toolbars

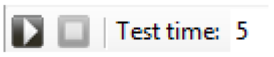


Figure 2.178 The main toolbar in the Guard Tour panel.

Start demo tour	Use this button to test the tour.
Stop demo tour	Use this button to stop testing the tour.
Test time	Test time is the number of seconds the camera stands in each position during the test tour.

Figure 2.179 The toolbar in the Cameras PTZ Guard Tour panel for managing the preset list.

Move up	Use this button to move the selected preset position(s) up in the list.
Move down	Use this button to move the selected preset position(s) down in the list.
Delete	Use this button to delete the selected preset position(s) from the list.

Figure 2.180 The toolbar in the Cameras PTZ Guard Tour panel for the list of available preset positions.

Add position

Use this button to add the selected preset position to the preset position tour list. The added preset position is inserted after the currently selected preset position in the tour list.

Note that you can have the same position several times in the tour list.

The following fields are part of the *Guard Tour* panel:

Name is the desired name of the guard tour. This name is used to identify the tour.

In the tour list of preset positions, there are two columns:

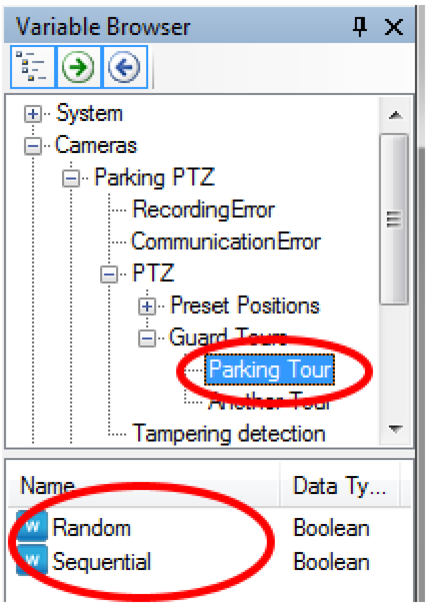
Position is the name of the preset position. This value is read-only.

Time is by default 10 seconds. You can change it to the desired value for each position.

Guard Tour variables

When defining a *Guard Tour* for a camera, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See Figure 2.181 for an example where a *Guard tour* is selected, and the variables belonging to the tour are presented in the lower pane (ringed in).



Name	Data Ty...
Random	Boolean
Sequential	Boolean

Figure 2.181 Variables for a Guard tour.

We will discuss script and variables later on in this manual, but it seems like a good idea to shortly describe the variables right now.

Random is a writable variable, meaning it can be activated via script or e.g., a button in Ethiris Client. When *true,* the camera starts the tour, and the preset positions are visited in random order.

Sequential is a writable variable, meaning it can be activated via script or e.g., a button in Ethiris Client. When *true* the camera starts the tour and the preset positions are visited in the order they are defined in the tour list.

2.4.21 Motion Detectors node

Under each Camera in the treeview, there is a *Motion Detectors* node. This is a container node for all motion detection definitions for the camera.

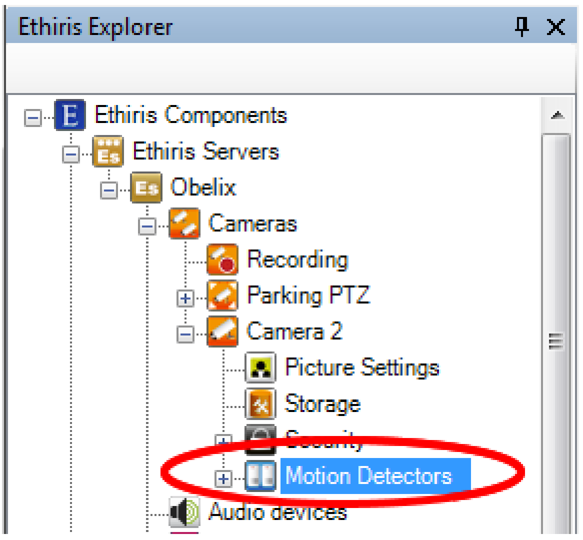


Figure 2.182 A Motion detectors node in Ethiris Explorer treeview.

Motion detectors popup menu

Right-clicking this node brings up a context menu.

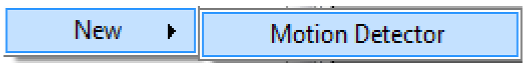


Figure 2.183 The popup menu for the Motion detectors node.

New->Motion Detector adds a new motion detector definition to the camera. It is immediately visible in the treeview as a new motion detector node. Should you have opened the *Motion Detectors* panel, the new motion detection would be added there too.

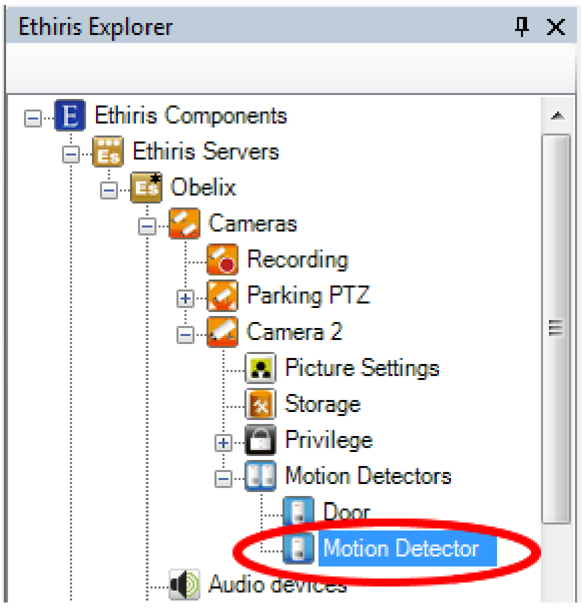
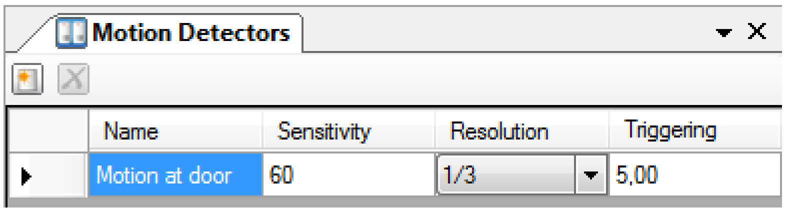


Figure 2.184 New Motion detector added.

Motion Detectors panel

Double-clicking the *Motion detectors* node in the treeview opens the corresponding panel.



	Name	Sensitivity	Resolution	Triggering
▶	Motion at door	60	1/3	5,00

Figure 2.185 The Motion detectors panel.

This panel consists of a list of all motion detectors currently defined for the camera.

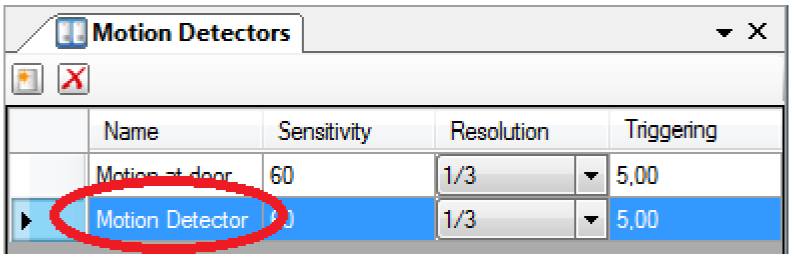
At the top of the panel, there is a toolbar.

Motion detectors panel toolbar

Figure 2.186 The toolbar in the Motion detectors panel.

New Motion detector. Use this button to create a new motion detector for the camera. This is the same as *New->Motion detector* in the popup menu described above. A new motion detector is immediately added to the server configuration. See *Figure 2.187* for an example of how it looks in the motion detectors list.

Delete Motion detector Use this button to delete the selected motion detector(s) from the configuration. You can select more than one motion detector by using the *Ctrl*-button and/or the *Shift*-button.



	Name	Sensitivity	Resolution	Triggering
	Motion at door	60	1/3	5,00
▶	Motion Detector	60	1/3	5,00

Figure 2.187 Motion detector added in the list.

Motion detectors panel motion detector list

The motion detector list consists of several columns.

Name is the name of the motion detector. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the Name column.

Sensitivity. Each pixel in a frame consists of three color components, red, green, and blue (RGB). Each color component can assume values between 0 and 255. Each color component's value in the current frame is compared with the value of the corresponding color component in the previous frame, pixel by pixel. If the difference is higher than the sensitivity set (0-100%), this pixel is regarded as having motion. A sensitivity near the maximum requires a small difference in color value, while a sensitivity near 0 requires a large difference in color value. In the preview window (which is displayed when opening the

Motion detector panel), pixels with motion detected are indicated in green, yellow, or red.

Green pixels indicate motion in individual pixels. They are not included in the calculation of the overall motion level. They are considered to be noise in the frame.

Adjacent pixels with motion are indicated in yellow. If the overall motion detection exceeds the set trigger level, adjacent pixels in motion are indicated in red.

Resolution indicates the proportion of pixels in the frame that are to be checked for motion. If you enter the maximum value (1/1), all pixels are checked. To reduce the load on the computer, you can choose not to check all the pixels in the frames. If, for example, you reduce the value to ½ (1 out of 2), only every other pixel is checked, both horizontally and vertically, i.e., every fourth pixel on average. If you reduce the value by one further step to 1/3, only every third pixel is checked in each direction, i.e., every ninth pixel, etc. The lowest resolution, 1/10, therefore, means that every tenth pixel in each direction is checked, i.e., every 100th pixel. In the preview window, you can continuously see the pixels with motion detected indicated in green, yellow, or red (see Sensitivity above).

Triggering indicates what proportion of the pixels (0.01-100.00%) in the area checked must indicate motion to start recording frames. It can be entered in hundredths of %.

In the preview window, the current level of motion is displayed to the left in a meter. The motion level in % is indicated below the meter. In the center of the meter is a bar that symbolizes the current trigger limit. The bar in the meter is green while the motion level is below the limit. It becomes red when the limit is exceeded.

2.4.22 Motion Detector node

Under the Motion Detectors node of a camera in the treeview, there are possibly one or several *Motion detector* nodes.

The purpose of these nodes is to enter detailed settings for a motion detector if necessary.

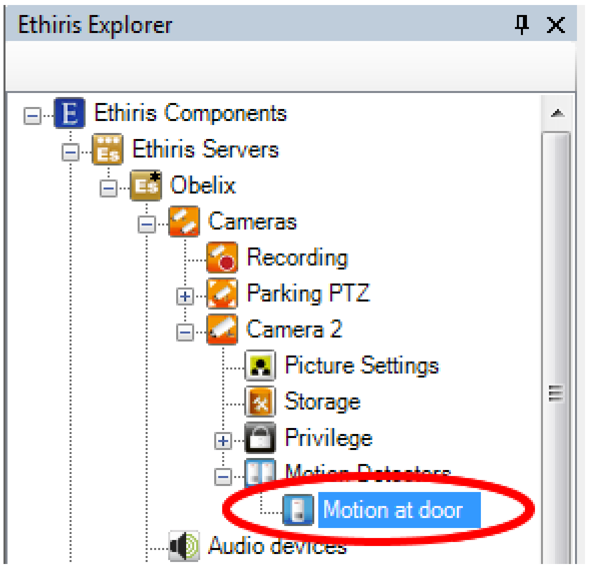


Figure 2.188 A Motion detector node for a Camera in the Ethiris Explorer treeview.

Motion detector popup menu

Right-clicking such a node brings up a context menu.

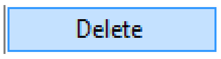
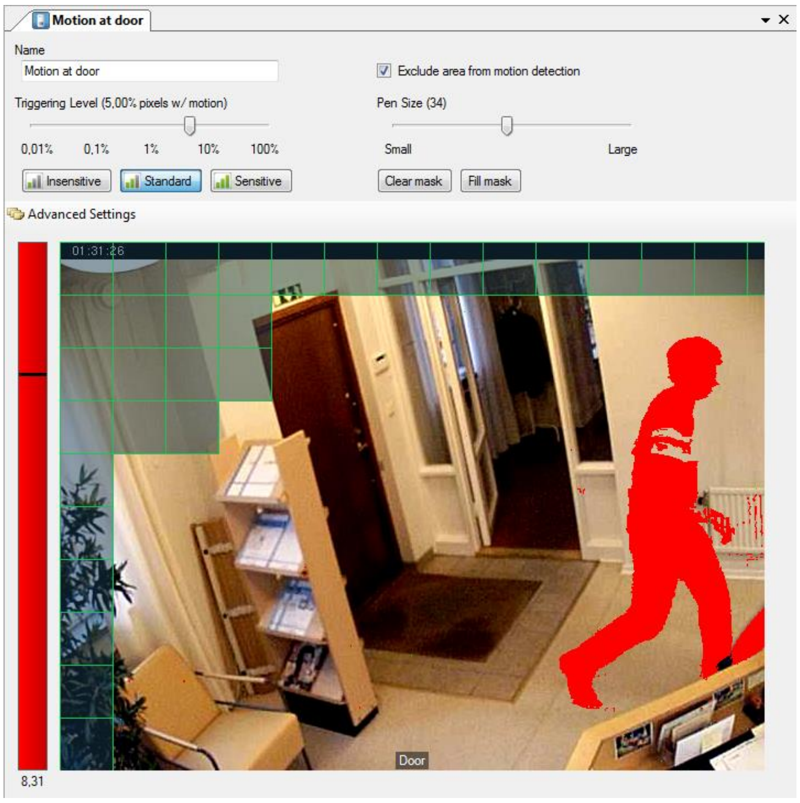


Figure 2.189 The popup menu for a Motion detector node.

Delete removes the motion detector from the server's configuration. Selecting this menu item has the same effect as clicking the *Delete Motion detector* button in the toolbar of the *Motion detectors* panel described above.

Motion detector panel

Double-clicking a *Motion detector* node for a camera in the treeview opens the corresponding panel.



Motion at door

Name: Motion at door

☑ Exclude area from motion detection

Triggering Level (5,00% pixels w/ motion)

Pen Size (34)

Figure 2.190 The Camera Motion detector panel.

In this panel, you can fine-tune the settings for motion detection. To assist you in determining the appropriate trigger level, there is a motion meter to the left, indicating the current motion level in the video. In the example above, the current motion level is 8.31%, which in this case, exceeds the *Trigger level* that is set to 5.00%. When the trigger level is exceeded, the motion meter has a red color, and the motion pixels in the video frame are also colored in red.

The following fields are part of the *Motion detection* panel:

Name is the desired name of the motion detection. This name is used to identify the motion detector.

Triggering level is the same as described above in the section *Motion detectors panel motion detector list* on *page 2:129*.

In this panel, there are three buttons with pre-defined settings; *Insensitive*, *Standard,* and *Sensitive*. When you create a new motion detector, the *Standard* setting is pre-selected. The following settings are pre-defined for the three different settings:

	Insensitive	**Standard**	**Sensitive**
Triggering level	10	5	2
Sensitivity	50	60	75
Resolution	1/5	1/3	1/3
Frame rate	2	2	1
Number of pictures for trigger	2	1	1
Background filtering	Off	Off	On (5 frames/10s)

By clicking any of the buttons, the parameters are changed according to the table above.

If you change the value for any of these 6 parameters, the settings will be considered user-defined, and as a consequence, no one of the three buttons will be selected (if you do not happen to enter values that exactly match any of the three pre-defined settings).

Exclude area from motion detection can be checked to be able to mask parts of the image. Sometimes you do not want to monitor the entire frame. There may be elements in the frame that continuously move, for example, a bush moving in the wind. Some parts of the frame may also not be interesting to monitor. You can then mask parts of the frame you do not want by painting over them.

Check the checkbox, set the pen width you want, and paint with the left mouse button in the frame. You can erase the mask with the right mouse button.

The **Clear mask button** removes any mask completely, i.e., the entire frame is monitored.

The **Fill mask button** masks the entire frame. The consequence of this is that no part of the frame is monitored. Obviously, this is not a very sensible setting. But if only a small part of the frame is to be monitored, this can be a quick way of achieving this.

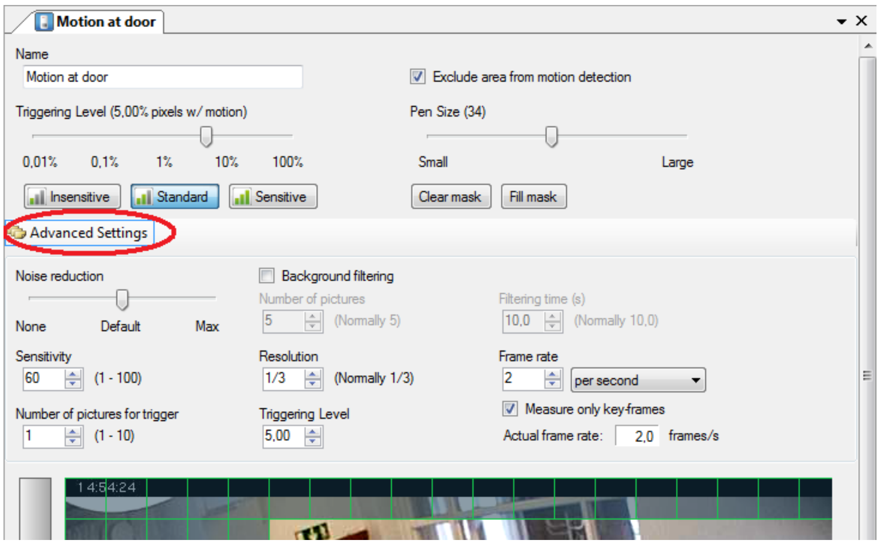
In the example above, the top left part of the frame is masked in respect of monitoring.

The smaller the area Ethiris needs to monitor, the smaller the load on the computer. So get into the habit of masking unnecessary parts of the frame.

Note that the mask does not in any way affect the recorded material. The complete image is recorded; it's only the motion detection that is performed on a smaller part of the image.

Advanced Settings

There is a button in the panel, which, when clicked, reveals additional fields that are used for advanced settings.


Motion at door

Name: Motion at door

[✓] Exclude area from motion detection

Triggering Level (5,00% pixels w/ motion)

Pen Size (34)

Noise reduction

[ ] Background filtering

Number of pictures: 5 (Normally 5)

Filtering time (s): 10,0 (Normally 10,0)

Sensitivity: 60 (1 - 100)

Resolution: 1/3 (Normally 1/3)

Frame rate: 2 per second

Number of pictures for trigger: 1 (1 - 10)

Triggering Level: 5,00

[✓] Measure only key-frames

Actual frame rate: 2,0 frames/s

Figure 2.191 Advanced settings in the Motion detector panel.

The following fields are part of the *Advanced settings* in the Motion detection panel:

Noise reduction is used to filter out noise from the image. There are 9 levels of noise reduction, reaching from *None* to *Max.* The default value will remove small and medium-size noise. To obtain the same function as in previous versions of Ethiris, set noise reduction to 1, i.e., one step to the right of *None.* Filtered out noise will be indicated in yellow, so it is easy to see what is filtered out when you change the setting.

Background filtering is used to obtain greater significance in the motion. Without background filtering, each new frame is compared with the previous frame. With background filtering, the new frame is instead compared with a synthetically produced background frame, which produces greater differences in motion.

Number of pictures. *An average frame of several frames back in time is used* as the background frame to compare new frames with. You enter here how many frames the background frame is to be calculated from.

Filtering time. You enter here how far back in time, the background frame should be calculated.

If the number of frames is 4 and the filter time is 12 seconds, this means that a frame is selected for average calculation of the background frame every 3 seconds, and the 4 most recent frames are included in the calculation.

Sensitivity is the same as described above in the section *Motion detectors panel motion detector list* on *page 2:129*.

Resolution is the same as described above in the section *Motion detectors panel motion detector list* on *page 2:129*.

Frame rate in this context has nothing to do with the recording frame rate!

Frame rate indicates how often motion is to be checked in the picture. Please note that this has nothing at all to do with what frame rate is used when recording video from this camera! This setting only determines how often motion is to be checked. A value of 2 frames per second usually is sufficient.

The fewer frames per second, the less strain on the server computer. Also, the difference between comparisons is greater (since more things have had time to change in the video). However, too long time between comparisons may result in failure to detect motion since, e.g., a person might have time to slip by the camera without being noticed.

Again, a value of 2 frames per second usually works fine.

Measure only key frames. We strongly recommend checking this checkbox. It has a huge impact on performance if the video format is MPEG-4, H.264, or H.265. In these cases, Ethiris Server doesn’t have to decode the so-called *P-frames* in the video stream. A MPEG-4/H.264/H.265 video stream consists of both keyframes and P frames. Keyframes (Complete images) arrive rather seldom, maybe 1 – 2 per second, and after a keyframe, a number of P-frames (changes only) arrive.

Number of pictures for trigger. Here you can enter how many comparisons in succession must be over the trigger limit before a proper alarm is produced.

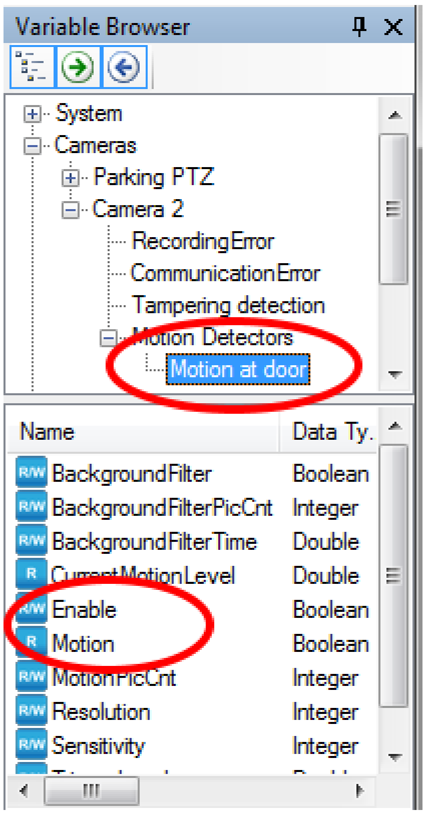
This may be useful to filter out false alarms because the camera changes brightness, or someone switches on/off a light that affects the camera.

Triggering level. This is just another way of setting the triggering level. In this field, it is easier to fine-tune the level since you can set it in 1/100 of a percent.

Motion detector variables

When defining a *Motion detection* for a camera, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.192* for an example where a *Motion detection* is selected, and the variables belonging to the motion detection are presented in the lower pane (ringed in).



Name	Data Ty.
BackgroundFilter	Boolean
BackgroundFilterPicCnt	Integer
BackgroundFilterTime	Double
CurrentMotionLevel	Double
Enable	Boolean
Motion	Boolean
MotionPicCnt	Integer
Resolution	Integer
Sensitivity	Integer

Figure 2.192 Variables for a Motion detector.

We will discuss script and variables later on in this manual, but it seems like a good idea to shortly describe the most important variables right now. As you can see in the screenshot above, there are quite a few variables for a motion detector. Most of them are intended to be able to adjust the settings via script or via OPC.

The two ringed in variables are those that are most often used, especially *Motion*.

Motion is a read-only variable that is *true* when the motion level in the video is exceeding the triggering level. This is by far the most often used motion detection variable. For more information about how to use it in Ethiris Script, please see the *Getting Started with Ethiris* manual.

Enable is a writeable variable, meaning that you can activate it via script or e.g., a button in Ethiris Client. The purpose of this variable is to turn the motion detection *on* or *off*. When *true,* the motion detection runs. When *false,* it doesn't. The *Enable* variable is a good way to control when motion detection is active by, e.g., a schedule. Note that when *Enable* is *false,* the motion detection will not run, meaning that Ethiris Server has less to do, and resources are preserved.

2.4.23 Audio devices node

Under each Ethiris Server in the treeview, there is an *Audio devices* node. This is a container node for all audio devices connected to the server. Currently, only audio devices from Axis are supported. As a bonus, you can also connect Axis cameras as audio devices. This requires the firmware in the camera to be 4.40 or later.

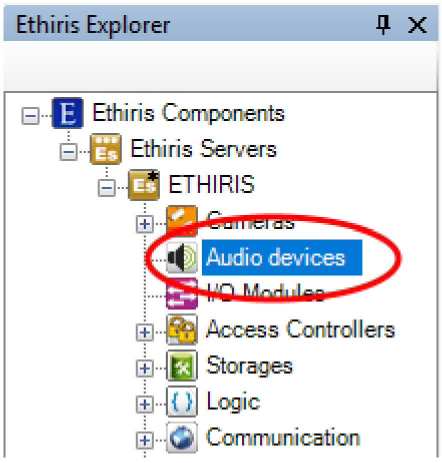


Figure 2.193 The Audio devices node in Ethiris Explorer.

Audio devices popup menu

Right-click on this node opens a menu.

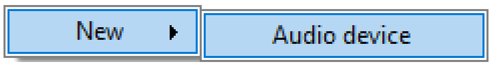


Figure 2.194 The popup menu for the Audio devices node.

New->Audio device adds a new audio device to the server configuration. It will be immediately visible in the treeview as a new nod. If the panel *Audio devices* is open, it will be displayed there as well.

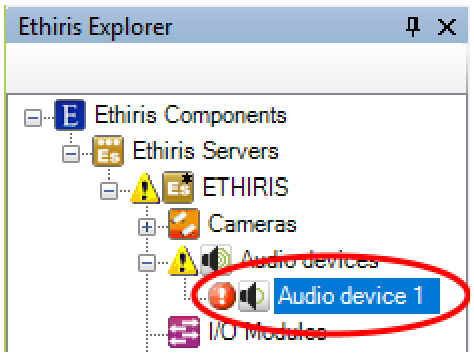
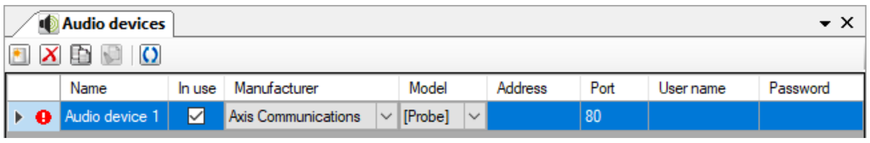


Figure 2.195 A new audio device has been added to the configuration.

Note the icon that indicates an error to the left of the new audio device. This is because the new audio device has no IP-address yet, and it also has not been probed so that Ethiris can build the configuration from the audio device configuration. There are also warning icons at higher levels in the treeview to warn that there are errors at lower levels in the configuration tree.

Audio devices panel

Double click on the *Audio devices* treeview opens the corresponding panel.



	Name	In use	Manufacturer	Model	Address	Port	User name	Password
	Audio device 1	✓	Axis Communications	[Probe]		80		

Figure 2.196 The Audio devices panel.

This panel consists of a list of all audio devices currently in the server's configuration.

At the top of the panel, there is a toolbar.

Audio devices panel toolbar

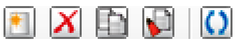


Figure 2.197 The toolbar in the panel Audio devices.

Add a new Audio device.	Use this button to create a new audio device. This is the same as selecting the menu *New->Audio device* in the popup menu described above. A new audio device will be added to the server configuration immediately. See *Figure 2.196* for an example of what it looks like.
Delete Selected Audio device(s)	Use this button to remove selected audio devices from the configuration. You can select more than one audio device by using the *Ctrl*-key and/or the *Shift*-key.
Copy	This button is used to copy the selected audio device(s) from the configuration. Use the *Paste* button later to create a new audio device with the same settings as the one you copied.
Paste as new	Creates a new audio device with the same settings as the one you copied earlier with the *Copy* button. You can paste several times to create several audio devices with the same settings.
Refresh Audio device	Use this button to update all selected audio devices. Ethiris will ask all selected audio devices (that have an IP-address) for information like *model, configuration, etc.* This must be done once on each audio device after adding them to the configuration.

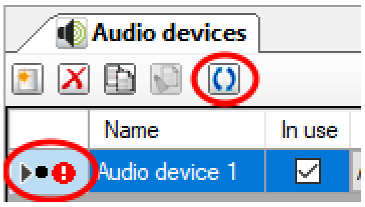
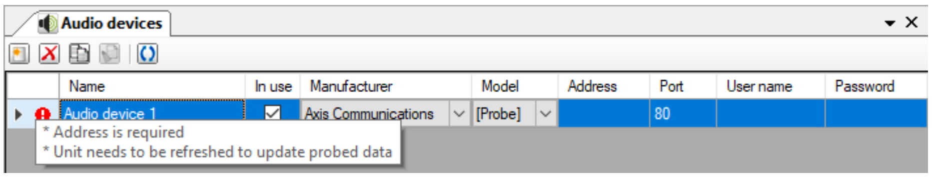


Figure 2.198 Indication that probing is going on for this audio device.

The small black dot indicates that probing is taking place for the audio device.

New Audio device



* Address is required
* Unit needs to be refreshed to update probed data

Figure 2.199 Audio device added to the audio devices list.

When a new audio device is added to the configuration, initially, it has no IP-address, this must be specified manually. After specifying an IP-address and possibly a user name and password, the audio device must be updated so Ethiris Server can get the current configuration of the audio device.

Audio devices panel Audio device list

The audio device list has several columns.

Name is the name of the audio device. The name must be unique in the configuration. If you enter an invalid name an error icon is displayed to the left of the audio device in the list.

In use is checked by default, this means the module should be used, and Ethiris should communicate with it. If it’s not checked, Ethiris will not communicate with the module.

Manufacturer specifies who produced the audio device. Each manufacturer has a specific set of audio devices that are supported by Ethiris. This column is a list of available manufacturers.

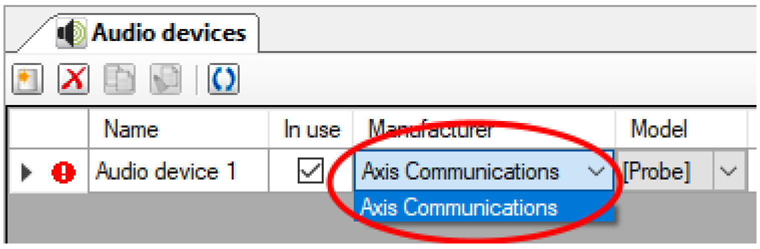


Figure 2.200 List of manufacturers.

Model is the model of the audio device. This will be *[Probe]* (or the model name inside[ ]). This means that Ethiris can probe the audio device to find out the model and how it is configured.

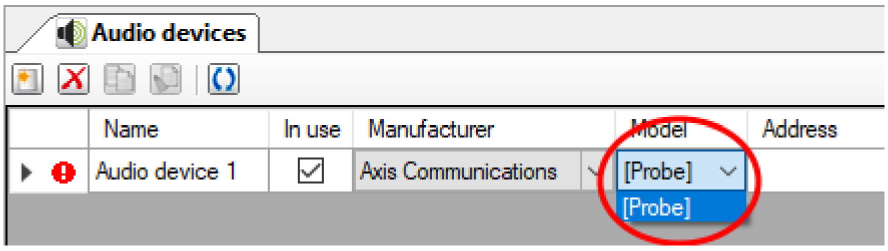


Figure 2.201 The list of models.

Address is the IP-address of the audio device (or DNS-name if DNS is available).

Port is the port Ehiris should connect to on the device, usually port 80.

User name is the login name if the audio device requires login.

Password is the password if the audio device requires login.

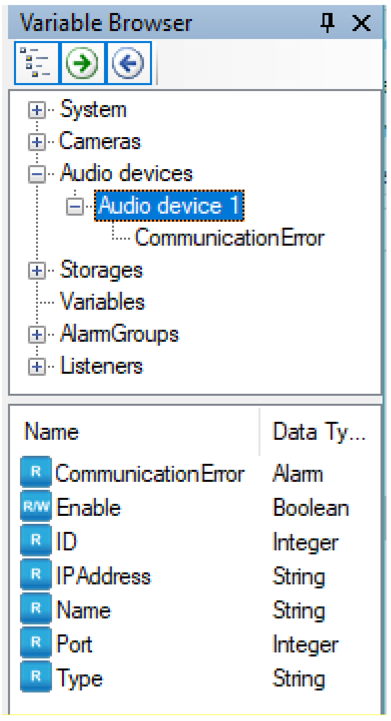
Audio device panel

There is no panel at this level. The *Audio devices* panel will open.

Audio device variables

When defining an *Audio device, some variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store.


Variable Browser

- System
- Cameras
- Audio devices
  - Audio device 1
    - CommunicationError
- Storages
- Variables
- AlarmGroups
- Listeners

Name	Data Ty...
R CommunicationError	Alarm
R/W Enable	Boolean
R ID	Integer
R IPAddress	String
R Name	String
R Port	Integer
R Type	String

Figure 2.202 Variables for an audio device.

We will discuss the script and variables later on in this manual.

2.4.24 I/O modules node

Under each Ethiris Server in the treeview, there is an *I/O Modules* node. This is a container node for all I/O modules connected to the server. Currently, only I/O modules from Axis are supported. As a bonus, you can also connect Axis cameras as I/O modules. This requires the firmware in the camera to be 4.40 or later.

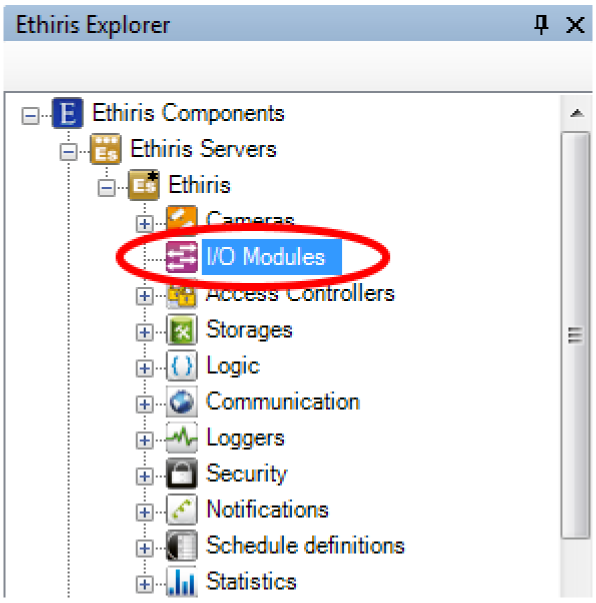


Figure 2.203 The I/O modules node in Ethiris Explorer.

I/O modules popup menu

Right-click on this node opens a menu.

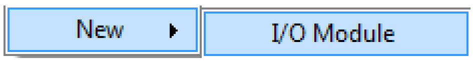


Figure 2.204 The popup menu for the I/O modules node.

New->I/O module adds a new I/O module to the server configuration. It will be immediately visible in the treeview as a new nod. If the panel *I/O modules* is open, it will be displayed there also.

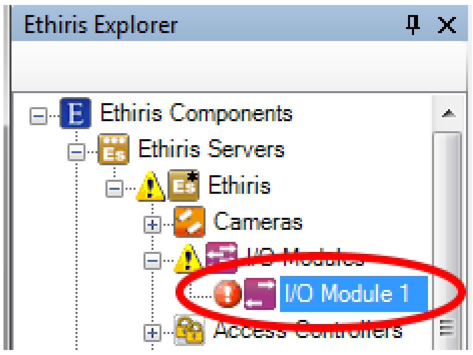
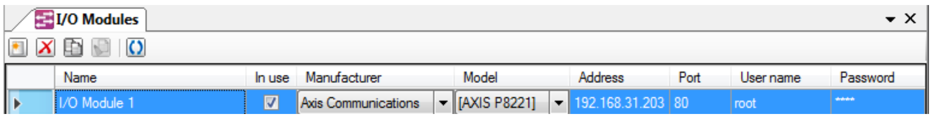


Figure 2.205 A new I/O module has been added to the configuration.

Note the icon that indicates an error to the left of the new I/O module. This is because the new I/O module has no IP-address yet, and it also has not been probed so that Ethiris can build the configuration from the I/O module configuration. There are also warning icons at higher levels in the treeview to warn that there are errors at lower levels in the configuration tree.

I/O modules panel

Double click on the *I/O modules* treeview opens the corresponding panel.



I/O Modules

	Name	In use	Manufacturer	Model	Address	Port	User name	Password
▸	I/O Module 1	✓	Axis Communications	[AXIS P8221]	192.168.31.203	80	root	****

Figure 2.206 The I/O modules panel.

This panel consists of a list of all I/O modules currently in the server's configuration.

At the top of the panel, there is a toolbar.

I/O modules panel toolbar

Figure 2.207 The toolbar in the panel IO-modules.

Add a new I/O Module

Use this button to create a new I/O module. This is the same as selecting the menu *New->I/O module* in the popup menu described above. A new I/O module will be added to the server configuration immediately. See *Figure 2.206* for an example of what it looks like.

Delete selected I/O Module(s)

Use this button to remove selected I/O modules from the configuration. You can select more than one I/O module by using the *Ctrl*-key and/or the *Shift*-key.

Copy

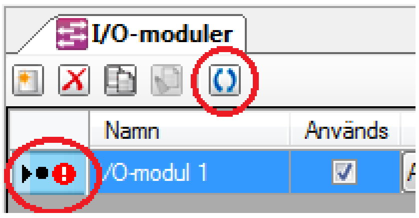
This button is used to copy selected I/O module(s) from the configuration. Use the *Paste* button later to create a new I/O module with the same settings as the one you copied.

Paste as new

Creates a new I/O module with the same settings as the one you copied earlier with the *Copy* button. You can paste several times to create several I/O modules with the same settings.

Refresh I/O Module

Use this button to update all selected I/O modules. Ethiris will ask all selected I/O modules (that has an IP-address) for information like *model, I/O-configuration, etc.* This must be done once on each I/O module after adding them to the configuration.



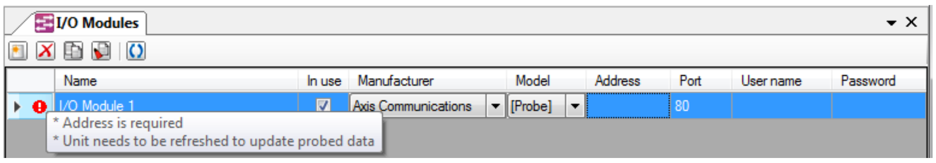
I/O-moduler

	Namn	Används
▸●	/O-modul 1	✓

Figure 2.208 Indication that probing is going on for this I/O module.

The small black dot indicates that probing is taking place for the I/O-module.

New I/O module



* Address is required
* Unit needs to be refreshed to update probed data

Figure 2.209 I/O module added to the I/O module list.

When a new I/O module is added to the configuration, initially, it has no IP-address. This must be specified manually. After specifying an IP-address and possibly a user name and password, the I/O module must be updated so Ethiris Server can get the current configuration of the I/O module.

I/O module panel I/O module list

The I/O module list has several columns.

Name is the name of the I/O module. The name must be unique in the configuration. If you enter an invalid name an error icon is displayed to the left of the I/O module in the list.

In use is checked by default, this means the module should be used, and Ethiris should communicate with it. If it's not checked, Ethiris will not communicate with the module, and the inputs- and outputs located on this module will not function.

Manufacturer specifies who produced the I/O module. Each manufacturer has a specific set of I/O modules that are supported by Ethiris. This column is a list of available manufacturers.

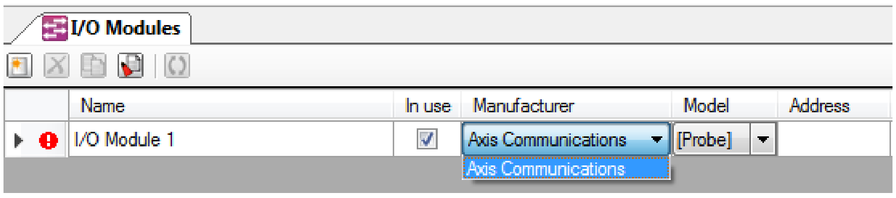


Figure 2.210 List of manufacturers.

Model is the model of the I/O module. This will be *[Probe]* (or the model name inside[ ]). This means that Ethiris can probe the I/O module to find out the model and how it is configured.

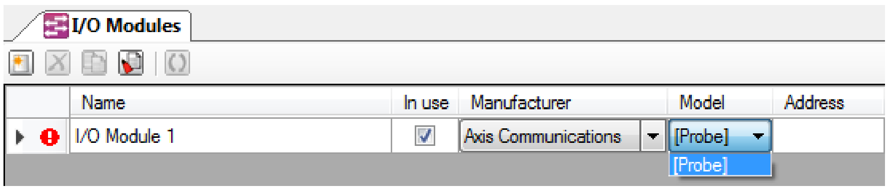


Figure 2.211 The list of models.

Address is the IP-address of the I/O-module (or DNS-name if DNS is available).

Port is the port Ehiris should connect to on the module, usually port 80.

User name is the login name if the I/O module requires login.

Password is the password if the I/O module requires login.

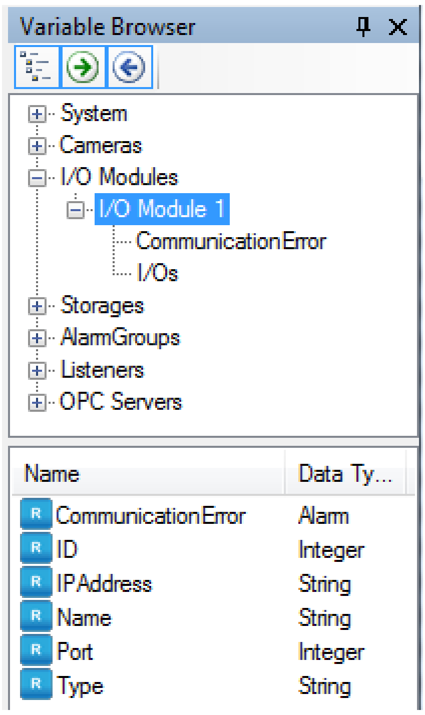
I/O module panel

There is no panel at this level. The *I/O modules* panel will open.

I/O module variables

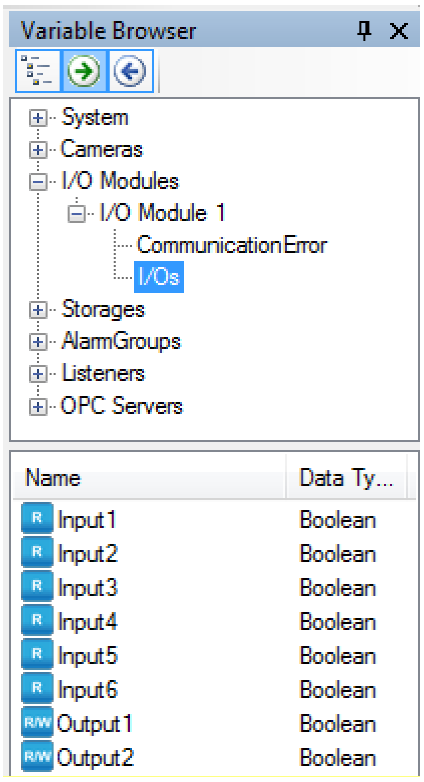
When defining an *I/O module*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store.



Name	Data Ty...
CommunicationError	Alarm
ID	Integer
IPAddress	String
Name	String
Port	Integer
Type	String

Figure 2.212 Variables for an I/O module.



Name	Data Ty...
Input1	Boolean
Input2	Boolean
Input3	Boolean
Input4	Boolean
Input5	Boolean
Input6	Boolean
Output1	Boolean
Output2	Boolean

Figure 2.213 I/O-variables for an I/O module.

We will discuss script and variables later on in this manual, but it seems like a good idea to shortly describe the most important variables right now.

There are a number of variables for an I/O module. There are also variables representing the I/O signals on the I/O module. The number of input and output signals depends on the I/O module, and so it can vary.

Input<n> is a readable variable representing the status of a physical input on the I/O module.

Output<n> is a variable that is both readable and writable. This means that even if it's an output, it is possible to read the value on it from a script. Depending on the support in the module, it might be so that the value read is the actual value in the module, even if this has been changed outside of the control of Ethiris. Otherwise, the value read is simply the last value written to the Ethiris Server data store.

2.4.25 Access Controllers node

Below each Ethiris Server in the treeview, there is an *Access Controllers* node. This is a collection node for the Access Controllers that are configured in the Ethiris Server. Currently, only Access Controllers from Axis are supported, more specifically, the A1001.

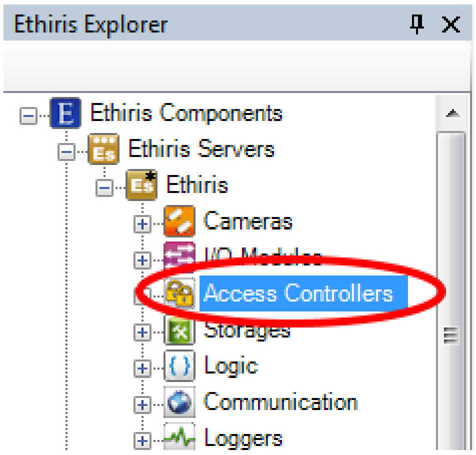


Figure 2.214 The node Access Controllers in Ethiris Explorer.

Access Controllers popup menu

A right-click on this node brings up this menu.

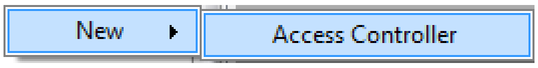


Figure 2.215 Popup menu for the nod Access Controllers.

New->Access Controller adds a new Access Controller to the server configuration. It will be displayed immediately in the treeview as a new node. If the panel *Access Controllers* is open, it will be displayed there as well.

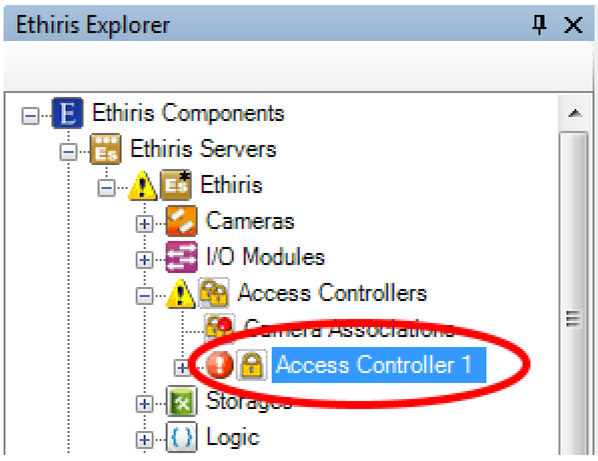
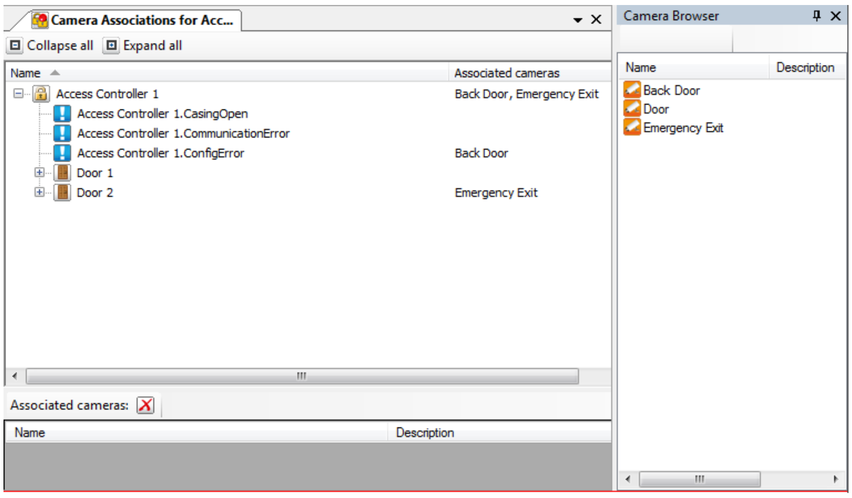


Figure 2.216 A new Access Controller has been added to the configuration.

Note the icon that indicates an error to the left of the new Access Controller. This is because the new Access Controller has no IP-address yet, and it also has not been probed so that Ethiris can build the configuration from the Access Controller configuration. There are also warning icons at higher levels in the treeview to warn that there are errors at lower levels in the configuration tree.

Camera Associations panel

Double click on the node Camera Associations opens the corresponding panel.



Name	Associated cameras
Access Controller 1	Back Door, Emergency Exit
Access Controller 1.CasingOpen	
Access Controller 1.CommunicationError	
Access Controller 1.ConfigError	Back Door
Door 1	
Door 2	Emergency Exit

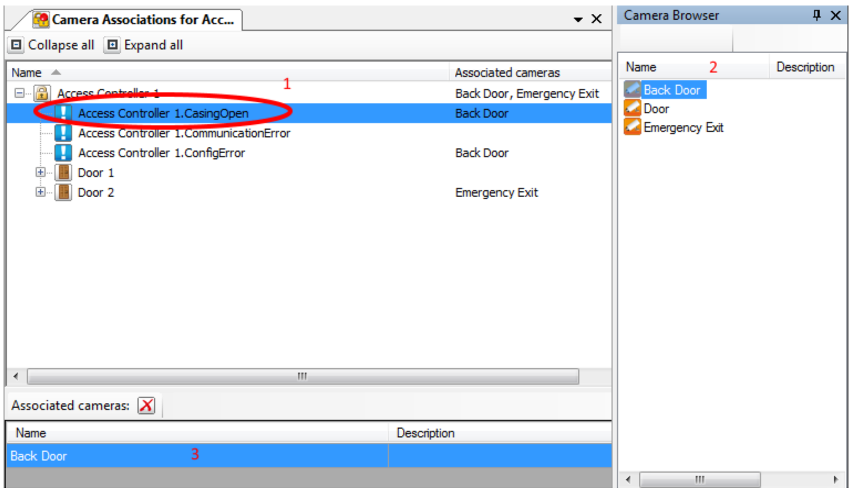
Name	Description
Back Door	
Door	
Emergency Exit	

Figure 2.217 The panel Camera Associations.

This panel has three parts. Up to the left, you can see a combined treeview/list with all configured access controllers. The treeview shows the configuration of access controllers regarding what doors are configured, and for each door which Access Points are available. For each Access Controller, Door, and Access Point, one or more events and/or alarms are generated. In this list, you can associate cameras with these events/alarms so that event recording will be triggered automatically when the event/alarm is activated by the Access Controller.

Up to the right, there is a list of all cameras in the configuration. Below is a list where you define the associations between camera and event/alarm.

To associate the camera *Back Door* with the alarm *CasingOpen*, you select the row with the alarm (1) that you wish to associate with a camera (2), then drag and drop the camera(s) on the gray area (3) in the list of associated cameras.



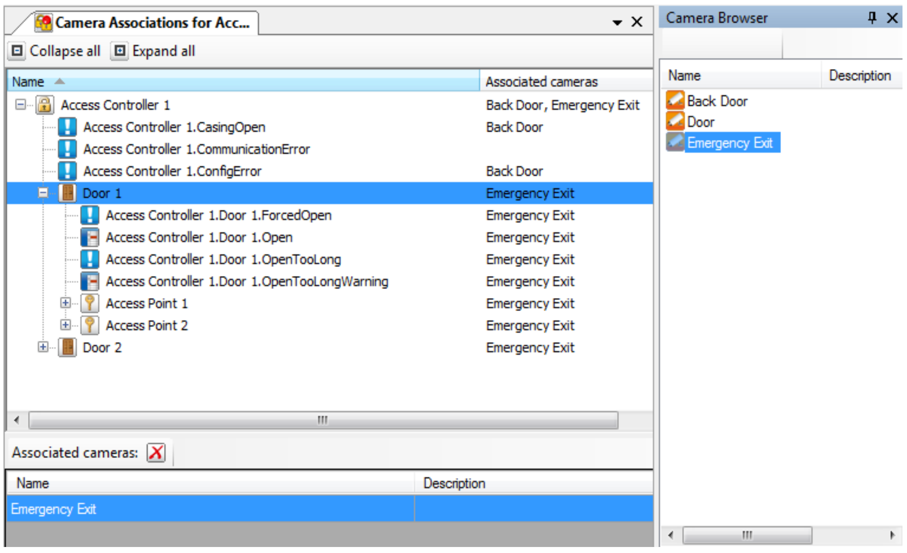
Name	Associated cameras
Access Controller 1	Back Door, Emergency Exit
Access Controller 1.CasingOpen	Back Door
Access Controller 1.CommunicationError	
Access Controller 1.ConfigError	Back Door
Door 1	
Door 2	Emergency Exit

Name	Description
Back Door	
Door	
Emergency Exit	

Name	Description
Back Door	

Figure 2.218 Camera associated with an alarm.

You can select several alarms/events in the upper left list and then drag and drop a camera into the list below. The camera will then be associated with all the selected events/alarms. You can also select, for example, a door in the list, like *Dorr 1*. The camera you drag and drop will then be associated with all events/alarms on that door.



Name	Associated cameras
Access Controller 1	Back Door, Emergency Exit
Access Controller 1.CasingOpen	Back Door
Access Controller 1.CommunicationError	
Access Controller 1.ConfigError	Back Door
Door 1	Emergency Exit
Access Controller 1.Door 1.ForcedOpen	Emergency Exit
Access Controller 1.Door 1.Open	Emergency Exit
Access Controller 1.Door 1.OpenTooLong	Emergency Exit
Access Controller 1.Door 1.OpenTooLongWarning	Emergency Exit
Access Point 1	Emergency Exit
Access Point 2	Emergency Exit
Door 2	Emergency Exit

Associated cameras:

Name	Description
Emergency Exit	

Name	Description
Back Door	
Door	
Emergency Exit	

Figure 2.219 Camera associated with the door.

To remove an association, you select the alarm/event in the upper left list and the association in the list below, then click the button *Remove camera association*.

At the top of the panel, there is a toolbar.

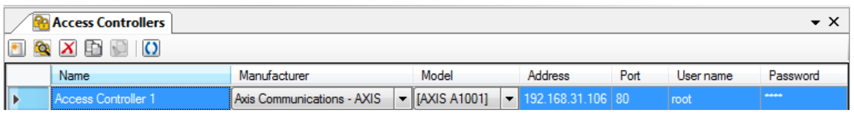
Camera associations toolbar

Figure 2.220 Toolbar in the panel Camera Associations.

Collapse all	Use this button to close all nodes in the tree.
Expand all	Use this button to open all nodes in the tree.

Access Controllers panel

A double click on the node *Access Controllers* in the treeview will open the corresponding panel.



Name	Manufacturer	Model	Address	Port	User name	Password
Access Controller 1	Axis Communications - AXIS	[AXIS A1001]	192.168.31.106	80	root	****

Figure 2.221 The panel Access Controllers.

This panel holds a list of all Access Controllers in the configuration.

At the top of the panel, there is a toolbar.

Access Controllers panel toolbar

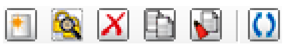


Figure 2.222 Toolbar in the Access Controllers panel.

Add a new Access Controller

Use this button to create a new Access Controller. This is the same as selecting the menu *New->Access Controller* in the popup menu described above. A new Access Controller is added to the configuration immediately. See Figure 2.224 for an example of what it looks like in the list.

Browse for Access Controllers

Use this button to search for Access Controllers on the network. See *Figure 2.118* for the view of the search dialog.

Delete Selected Access Controller(s)

Use this button to remove selected Access Controllers from the configuration. You can select more than one Access Controller by using the *Ctrl*-key and/or the *Shift*-key.

Copy

This button is used to copy the selected Access Controller(s) from the configuration. Use the *Paste* button later to create a new Access Controller with the same settings as the one you copied.

Paste as new

Creates a new Access Controller with the same settings as the one you copied earlier with the *Copy* button. You can paste several times to create several Access Controllers with the same settings.

Refresh Access Controller

Use this button to update all selected access controllers. Ethiris will ask all selected access controllers (that has an IP-address) for information like *model, configuration, etc.* This must be done once on each Access Controller after adding them to the configuration.

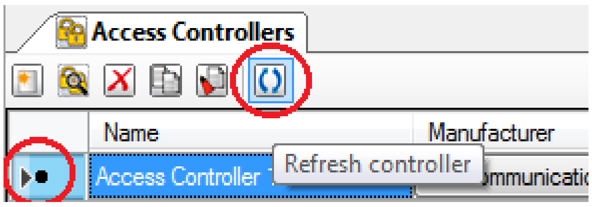
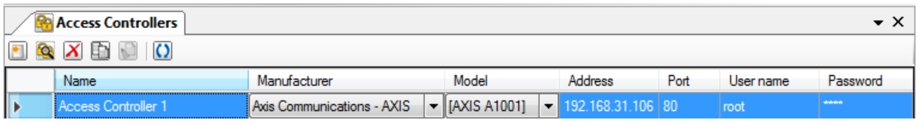


Figure 2.223 Indication of probing of this Access Controller.

The small black dot indicates that probing is taking place on the Access Controller.

New Access Controller



Access Controllers

Name	Manufacturer	Model	Address	Port	User name	Password
Access Controller 1	Axis Communications - AXIS	[AXIS A1001]	192.168.31.106	80	root	****

Figure 2.224 Access Controller added to the configuration.

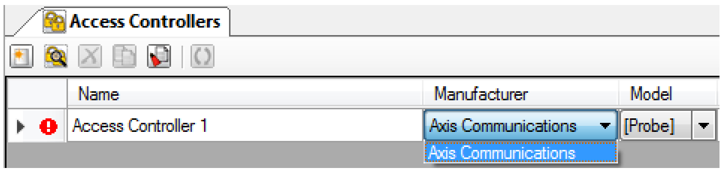
When a new Access Controller is added to the configuration, initially, it has no IP-address, this must be specified manually. After specifying an IP-address and possibly a user name and password, the Access Controller must be updated so Ethiris Server can get the current configuration of the Access Controller.

Access Controller panel Access Controller list

The Access Controller list has several columns.

Name is the name of the Access Controller. The name must be unique in the configuration. If you enter an invalid name an error icon is displayed to the left of the Access Controller in the list

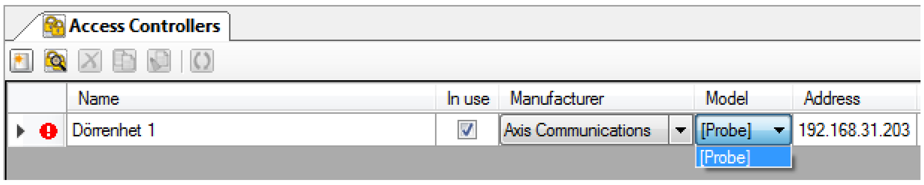
Manufacturer specifies who produced the Access Controller. Each manufacturer has a specific set of Access Controllers that are supported by Ethiris. This column is a list of available manufacturers.


Access Controllers

	Name	Manufacturer	Model
▶ ●	Access Controller 1	Axis Communications	[Probe]

Figure 2.225 List of manufacturers.

Model is the model of the Access Controller. This will be *[Probe]* (or the model name inside[ ]). This means that Ethiris can probe the Access Controller to find out the model and how it is configured


Access Controllers

	Name	In use	Manufacturer	Model	Address
▶ ●	Dörrenhet 1	✓	Axis Communications	[Probe]	192.168.31.203

Figure 2.226 List of models.

Address is the IP-address of the Access Controller (or DNS-name if DNS is available).

Port is the port Ehiris should connect to on the Access Controller, usually port 80.

User name is the login name if the Access Controller requires login.

Password is the password if the Access Controller requires login.

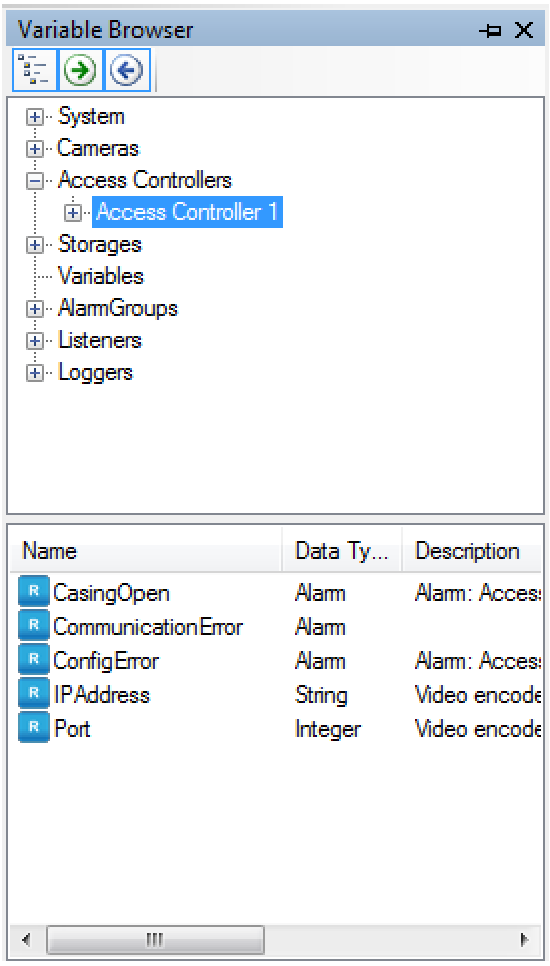
Access Controller panel

There is no panel at this level.

Access Controller variables

When defining an *Access Controller*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See Figure 2.227 for an example where an *Access Controller* is selected and the corresponding variables are shown in the lower panel.



Name	Data Ty...	Description
CasingOpen	Alarm	Alarm: Acces
CommunicationError	Alarm	
ConfigError	Alarm	Alarm: Acces
IPAddress	String	Video encode
Port	Integer	Video encode

Figure 2.227 Variables for an Access Controller.

We will discuss the script and variables later on in this manual.

Doors panel

There is no panel at this level.

2.4.26 Door node

Below an Access Controller in the treeview, there is a node for each door that is configured in the Access Controller.

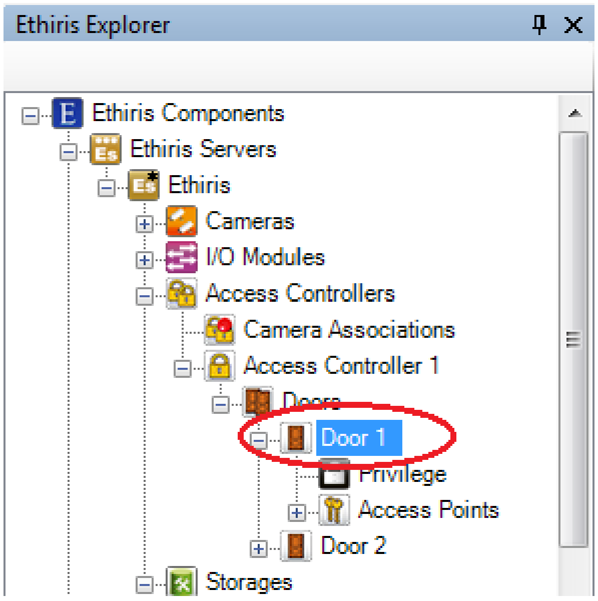


Figure 2.228 A door in Ethiris Explorer.

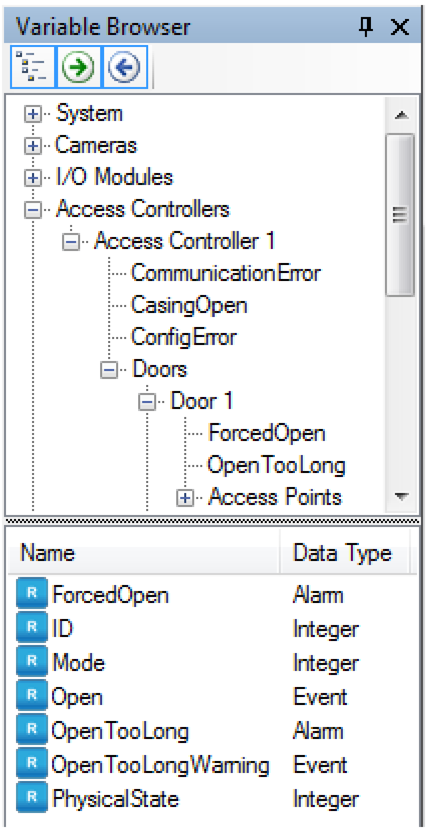
Door panel

There is no panel at this level.

Door variables

When defining an *Access Controller*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See Figure 2.229 for an example where a *Door* is selected and the corresponding variables are shown in the lower panel.



Name	Data Type
ForcedOpen	Alarm
ID	Integer
Mode	Integer
Open	Event
OpenTooLong	Alarm
OpenTooLongWarning	Event
PhysicalState	Integer

Figure 2.229 Variables for a door on an Access Controller.

We will discuss the script and variables later on in this manual.

2.4.27 Privilege node

Under the Door node in the treeview is also the *Privilege* node.

A number of specific user operations are defined in Ethiris. For each such user operation, you can, if you want, specify that the user who performs the operation must be a member of a certain user group in the Windows user system. This may be either an Ethiris user group, a local user group on the computer running Ethiris Server, or a global user group in a domain or the Active Directory if the computers and users involved are members of a domain. To be able to specify a user group in the domain, it is necessary for both the computer running Ethiris Server to be a member of the domain and the logon entered by the user to be for an account in the same domain.

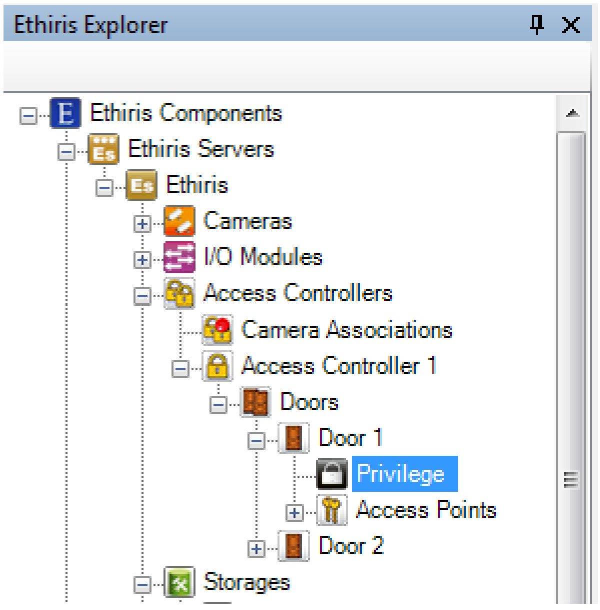


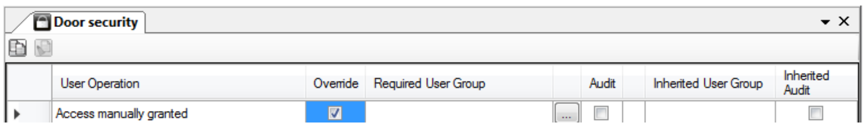
Figure 2.230 The node Security for a Door in Ethiris Explorer.

Privilege popup menu

There is no popup menu for this node.

Privilege panel

Double-clicking a *Security* node for a Door in the treeview opens the corresponding panel.



User Operation	Override	Required User Group	Audit	Inherited User Group	Inherited Audit
Access manually granted	☑	...	☐		☐

Figure 2.231 The panel Security for a door.

In this panel, you can enter settings for the camera regarding access control. The user operations listed here can be set both at the Ethiris Server level and specifically for each camera. In this panel, you set access control parameters for this camera specifically.

As a default, no log in is required. Every user has access to all functions. If you want to restrict access to a certain user operation, you have to specify which Windows user group the user has to be a member of to get access to the operation. To gain access the user has to log in as a Windows user that is a member of the required Windows user group.

At the top of the panel, there is a toolbar.

Privilege panel toolbar

Figure 2.232 The toolbar in the panel Privilege for a door.

Copy	Use this button to copy the content of a specific row in the *User Operations* list.
Paste	Use this button to paste the settings from the copied operation to one or several other operations. Select one or several by clicking in the area to the left of the *User Operation* column. Use the Ctrl or Shift keys on the keyboard together with the mouse to select several rows. You can also click and drag over several rows to select them.

The following columns are part of the *Operations* list in the Door Privilege panel:

User operation lists the operations that you can set login requirement on for a door. These are:

Access manually granted is to allow the user to manually open a door via a connected Door.

Access Point panel

There is no panel at this level.

2.4.28 Access point node

Below a door, in the treeview, there is a node for each Access Point that is configured for the door in the Access Controller.

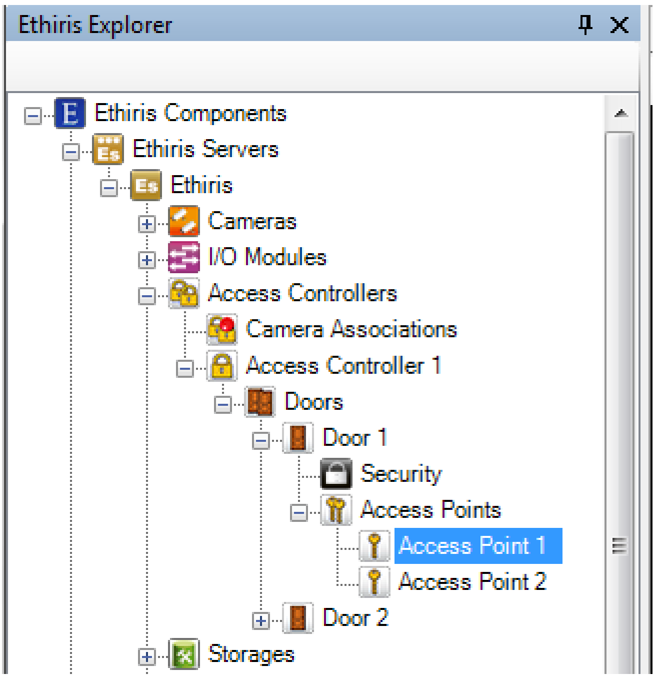


Figure 2.233 An Access Point on a door in Ethiris Explorer.

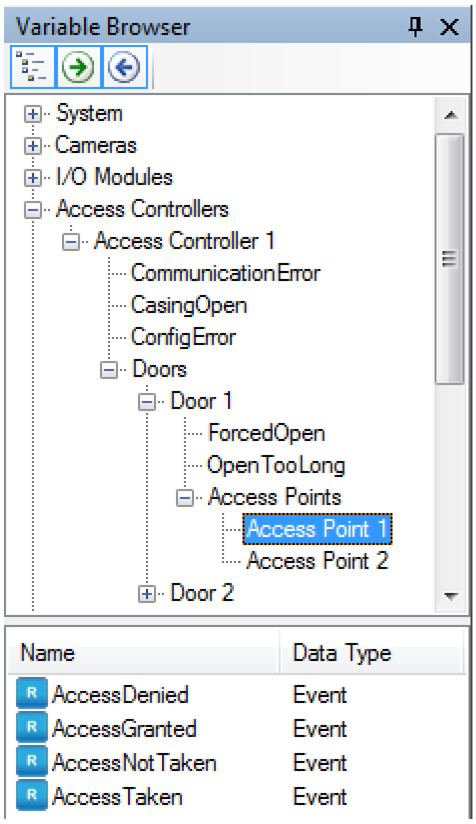
Access Point panel

There is no panel at this level.

Access Point variables

When defining an Access Controller, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See Figure 2.234 for an example where an *Access Point* is selected and the corresponding variables are shown in the lower panel.



Name	Data Type
AccessDenied	Event
AccessGranted	Event
AccessNotTaken	Event
AccessTaken	Event

Figure 2.234 Variables belonging to a door on an Access Controller.

We will discuss the script and variables later on in this manual.

2.4.29 Storages node

Under each Ethiris Server in the treeview, there is a *Storages* node. This is a container node for all storages that are defined in the server.

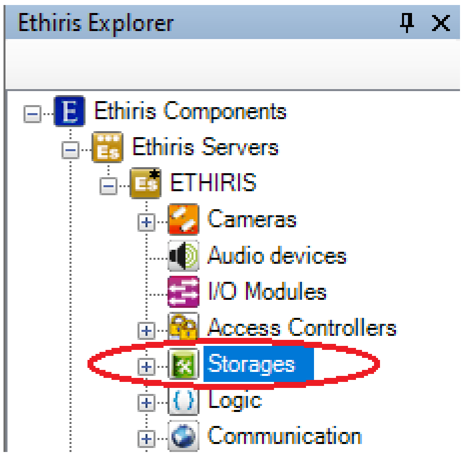


Figure 2.235 The Storages node in Ethiris Explorer treeview.

Storages popup menu

Right-clicking this node brings up a context menu.

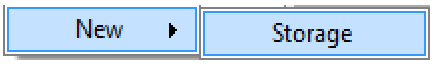


Figure 2.236 The popup menu for the Storages node.

New->Storage adds a new storage to the server configuration. It is immediately visible in the treeview as a new storage node. Should you have opened the *Storages* panel, the new storage would be added there too.

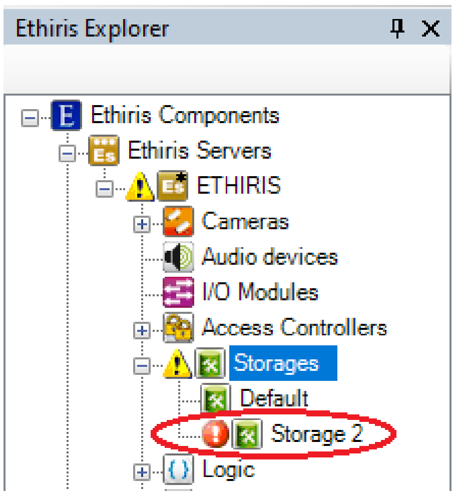
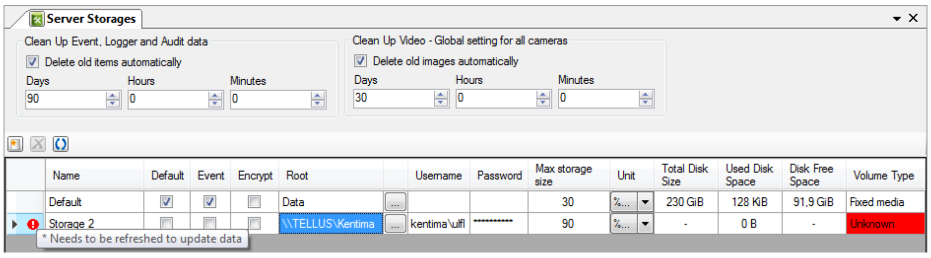


Figure 2.237 New storage added.

Notice the error icon to the left of the new storage node. This is because the new storage has no valid path. There are warning icons further up in the treeview indicating an error somewhere in the configuration.

Storages panel

Double-clicking the *Storages* node in the treeview opens the corresponding panel.



Server Storages

Clean Up Event, Logger and Audit data
☑ Delete old items automatically

Days	Hours	Minutes
90	0	0

Clean Up Video - Global setting for all cameras
☑ Delete old images automatically

Days	Hours	Minutes
30	0	0

	Name	Default	Event	Encrypt	Root		Username	Password	Max storage size	Unit	Total Disk Size	Used Disk Space	Disk Free Space	Volume Type
	Default	✓	✓	☐	Data	...			30	%...	230 GiB	128 KiB	91,9 GiB	Fixed media
▸ ❗	Storage 2	☐	☐	☐	\\TELLUS\Kentima	...	kentima\ulfl	***********	90	%...	-	0 B	-	Unknown

* Needs to be refreshed to update data

Figure 2.238 The Storages panel.

This panel consists of a list of all storages currently part of the server configuration.

At the top of the panel, there is a toolbar.

Storages panel toolbar

Figure 2.239 The toolbar in the Storages panel.

Add a new storage device.

Use this button to create a new storage. This is the same as *New->Storage* in the popup menu described above. A new storage is immediately added to the server configuration.

Delete selected storage device(s)

Use this button to delete the selected storage(s) from the configuration. You can select more than one storage by using the *Ctrl*-button and/or the *Shift*-button.

Refresh storage information

Use this button to refresh the information about the storage devices in the list. *Used disk space* and *Disk free space* may be changed while the panel is open, then you can click this button to retrieve current information. Ethiris also validates the path and checks that it can be written to. If any error occurs, an exclamation mark will be shown to the left on that row in the list. Hoover the exclamation mark with the mouse to get a message explaining the problem.

Storages panel general settings

At the top of the panel, there are general settings for automatic deletion of old data that apply for all storage devices.

Clean Up Event, Logger and Audit data

Event data are system events and possible custom-defined events that are presented in the Event tab in Ethiris Client.

Logger data applies to data that has been stored with the new function for historical variable data via *Loggers*.

Audit data are data that have been stored as audit trail information and are presented in the Event tab in Ethiris Client.

Delete old items automatically should be ticked if you want automatic deletion.

Days, **Hours** & **Minutes** are used for setting how old data can be before it is automatically deleted.

Clean Up video – Global setting for all cameras

This part concerns recorded video for all cameras regardless of which storage device the cameras used for storing video. Under each camera, you can override these global settings and make a specific clean up setting for each camera.

Delete old images automatically should be ticked if you want automatic deletion.

Days, **Hours** & **Minutes** are used for setting how old data can be before it is automatically deleted.

Storages panel storage list

The storage list consists of several columns.

Name is the desired name of the storage. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the storage in the list indicating the problem.

Default determines which storage is the default storage for recorded video for all cameras. Each camera can have specific storage for recorded video, but if you do not specify storage, the default storage will be used.

Event determines which storage is used for storing events. This is usually the same as *Default*. If you change which storage that is used for storing events, all old events in the system will disappear.

Encrypt determines if the files in the storage will be encrypted. If you choose to encrypt, that means that only Ethiris Server can read the video files. Not even an administrator can read or copy the files manually.

Ethiris Server may not see all disks that are visible in Ethiris Admin.

Root defines where the storage is located. Note that Ethiris Server may not ”see” the same disks as you do in Ethiris Admin. E.g., mapped disks (to a letter, like Z:\) will only be visible to a specific interactive user logged in to the computer.

Note! Ethiris Server normally runs under the *Local System* account, which **does not** have the permissions to access files or folders on the network. In the service's properties in Windows, you can change which account the service logs in as. This has to be configured in order to access a root directory, which is located on another computer. See below for an alternative method.

You can also specify a root directory by using UNC names, like \\GALATEA, for a computer named GALATEA. Anyway, the path is validated by Ethiris Server immediately. Any invalid path results in an error icon with information regarding the problem.

Note! You should only have one storage on each physical disk!

You should, of course, avoid having other types of files in an Ethiris Storage directory. An Ethiris storage is intended for recorded video only.

Username defines the username used to log on to the remote resource defined in the **Root** field. Normally this field is only used if **Root** points to a remote resource on another computer. The username and password should have both read, write, and delete permissions on the remote location.

Password defines the password used to log on to the remote resource defined in the **Root** field. Normally this field is only used if **Root** points to a remote resource on another computer. The username and password should have both read, write, and delete permissions on the remote location.

Max storage size defines how much storage space this storage is allowed to use. Depending on the *Unit,* the value has different meanings. The default value is *90 % of the total disk size*.

Unit is either *GiB* or *% of total disk size*.

Total Disk Size is a read-only value that presents the total size of the disk where the storage is located.

Used Disk Space is a read-only value that presents how much of the disk where the storage is located Ethiris has used for video.

Disk Free Space is a read-only value that presents the free disk space of the disk where the storage is located. At the time of writing, there is no information about how much of the used space is used by Ethiris. If everything is OK, this value has a *white* background. If something is not quite OK, the background will be *red*. And if free space is less than *10 GiB,* the background is red.

Volume type is a read-only value that presents the type of disk that the storage is located on. Possible types are *Fixed media* and *Network drive*. These two types are OK to use. There are a few other types that are not recommended to use. These are *Removable media*, *CD/DVD/BD drive*, *RAM disk,* or *Unknown*. In all these four cases, the *Free disk space* column will be *red*. *Removable media* includes *USB* and *Flashcard* from, e.g., a camera.

2.4.30 Storage node

Under the Storages node in the treeview, there is at least one *Storage* node.

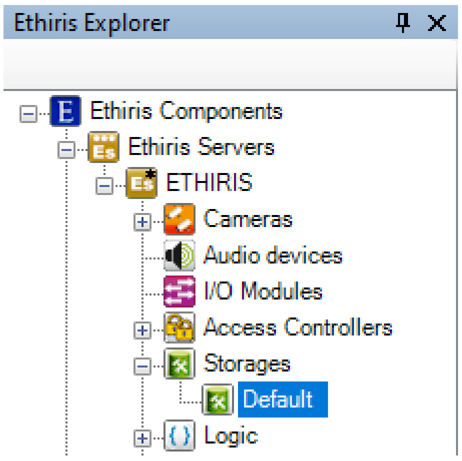


Figure 2.240 A Storage node in Ethiris Explorer treeview.

Storage popup menu

Right-clicking this node brings up a context menu.

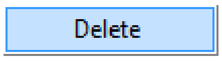
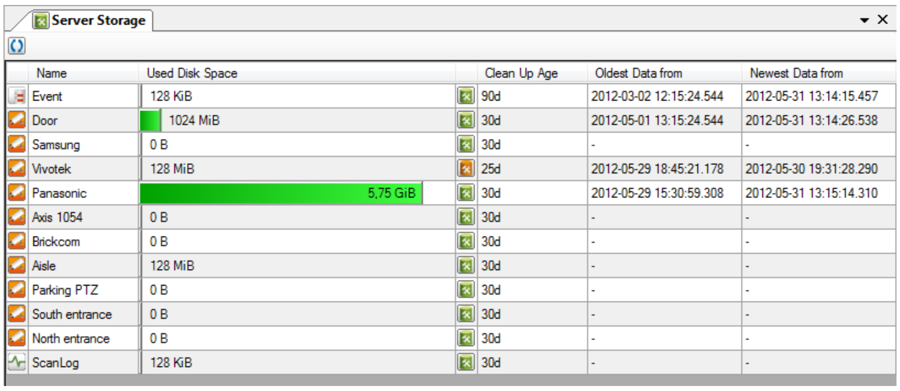


Figure 2.241 The popup menu for a Storage node.

Delete Removes the storage from the server configuration. It is immediately removed from both the treeview and the storage list in the Storages panel. Note that you cannot delete the *Default* or the *Event* storage.

Storage panel

Double-clicking a *Storage* node in the treeview opens a panel that presents an overview of what has been stored on the storage device as video data for cameras, possible event logs, and possible loggers for historical data logging.



Name	Used Disk Space	Clean Up Age	Oldest Data from	Newest Data from
Event	128 KiB	90d	2012-03-02 12:15:24.544	2012-05-31 13:14:15.457
Door	1024 MiB	30d	2012-05-01 13:15:24.544	2012-05-31 13:14:26.538
Samsung	0 B	30d	-	-
Vivotek	128 MiB	25d	2012-05-29 18:45:21.178	2012-05-30 19:31:28.290
Panasonic	5,75 GiB	30d	2012-05-29 15:30:59.308	2012-05-31 13:15:14.310
Axis 1054	0 B	30d	-	-
Brickcom	0 B	30d	-	-
Aisle	128 MiB	30d	-	-
Parking PTZ	0 B	30d	-	-
South entrance	0 B	30d	-	-
North entrance	0 B	30d	-	-
ScanLog	128 KiB	30d	-	-

Figure 2.242 The panel Storage.

This panel consists of a list of all cameras and logs that are using this storage device.

At the top of the panel, there is a toolbar.

Storage panel toolbar

Figure 2.243 The toolbar in the panel Storage.

Refresh storage information

Use this button to refresh the information about the storage device. The information may change while the panel is open, then you can click this button to retrieve current information.

Storage panel storage list

The storage list consists of several columns. All the information in the list is read-only.

Name is used for presenting the name of the camera or the log.

Used Disk Space presents the amount of data that has been stored for the camera/log

<icon> indicates if the clean up settings use the global setting or if you have made a specific setting for the camera.

Global setting for clean up

A green icon indicates that the camera uses the global setting for clean up of video.

Specific clean up setting

An orange icon indicates that the camera has specific settings for clean up of video.

Clean Up Age presents how old data is allowed before it is automatically deleted. Note that due to insufficient disk space, data may be deleted earlier.

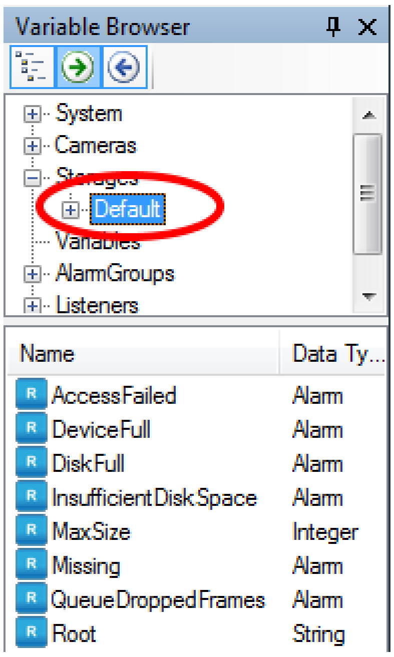
Oldest data from presents the timestamp for the oldest data for the camera/log.

Newest data from presents the timestamp for the newest data for the camera/log.

Storage variables

When defining a *Storage*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.244* for an example where a *Storage* is selected, and the variables belonging to the storage are presented in the lower pane (ringed in).



Name	Data Ty...
AccessFailed	Alarm
DeviceFull	Alarm
DiskFull	Alarm
InsufficientDiskSpace	Alarm
MaxSize	Integer
Missing	Alarm
QueueDroppedFrames	Alarm
Root	String

Figure 2.244 Variables for a Storage.

We will discuss the script and variables later on in this manual.

The most interesting variables for a Storage are the following:

AccessFailed is a read-only variable, which is *true* if Ethiris Server cannot access the storage even if the storage exists.

DeviceFull is a read-only variable, which is *true* if the device is full. Each storage device has a Max storage size for recorded video. If this size is not sufficient and all available storage space has been filled, this alarm will be activated. It implies that either too much video is recorded or that the time for automatic clean up of the old video is too long.

DiskFull is a read-only variable, which is *true* if the whole disk is full. This is more serious and happens when there is less than 1 GiB of storage space remaining on the actual disk that the storage device uses.

Missing is a read-only variable, that is *true* if a storage device for some reason suddenly does not exist. It may be due to a disk that has been removed, e.g., a USB-device (god forbid).

QueueDroppedFrames is a read-only variable, which is *true* if Ethiris Server stores frames in such a pace that the disk cannot write data on disk fast enough. Ethiris buffers frames in memory before they are written to disk. Depending on the amount of RAM available in the system, the queue of frames about to be written to disk may become too long. Instead of exhausting the memory and potentially crash the whole system, frames are dropped.

2.4.31 Logic node

The *Logic* node is just a collection node for *Variables*, *Alarms*, *Events,* and *Script* in the current project. There is neither a popup menu nor a panel associated with this node. If you double click this node, the panel *Script* will open.

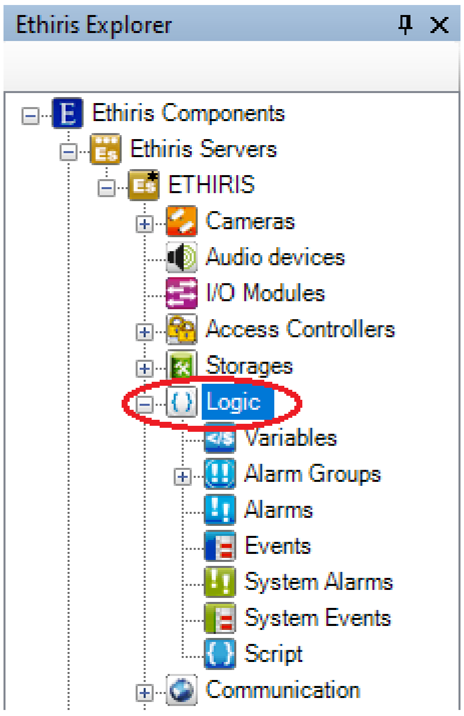


Figure 2.245 The Logic node in Ethiris Explorer treeview.

2.4.32 Variables node

Under the Logic node in the treeview, there is a *Variables* node. This is a container node for all internal variables added to the server.

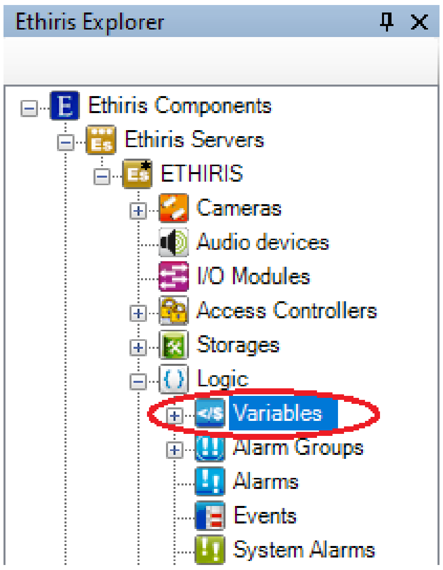


Figure 2.246 The Variables node in Ethiris Explorer treeview.

Variables are one of the cornerstones of the program. The functions in Ethiris come to life when they are linked to suitable variables and used in expressions and scripts. The variables can take their values from, for example, a motion detection state or an external sensor. They can also be used as output signals, for example, to start recording or sound a siren.

As there are various types of values, it is also necessary to have different types of variables. These different types of variables are called different data types. The different variables are created based on these different data types.

It is important to know the difference between the terms data type and variable. The data type determines the type of information the variable can store. We say that a variable is an instance of a specific type. There may be many variables of the same type. Only the individual variables contain data such as values and variable names.

The following four data types can be used in Ethiris:

Boolean consists of one bit and can thus have the value true or false. A motion detection variable which shows whether motion has been detected or not is Boolean. The variable's value is false while no motion is detected and becomes true as soon as motion is detected.

Integer is used for integers and consists of 32 bits (4 bytes) ≈ 4.2 billion different values. These can also be negative values, in such case -2.1 billion to 2.1 billion. The data type Integer is used, among other things, to store the number of events that have occurred for a camera.

Double is used for decimal numbers. The name comes from the size, 64 bits (8 bytes), which is double the size of Integer. The value interval is ”floating”, as it depends on the number of decimals. However, the precision of a double is always 15 value figures. A Double can thus be used for decimal numbers to define values more precisely. A camera's variable with the current frame rate is a Double.

String is used for texts. The size depends entirely on the length of the text string and can, in principle, be as large as you want. A camera’s variable for reading out its IP address is a String.

Each variable has a specific size, and this space is reserved in the memory and assigned an address. So that you do not need to keep track of the address of each variable, they are given names.

Variables in Ethiris are divided into different categories and are described in further detail in the following sections. In the *Variables* node, we define *internal variables*. These variables are not connected to a specific object, like a camera or a schedule. Internal variables are global variables that can be used in a script, Ethiris Clients, and even in OPC communication.

Variables popup menu

Right-clicking this node brings up a context menu.

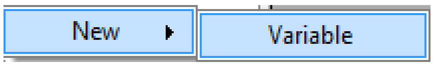


Figure 2.247 The popup menu for the Variables node.

New->Variable adds a new internal variable to the server configuration. It is immediately visible in the treeview as a new variable node. Should you have opened the *Variables* panel, the new variable would be added there too.

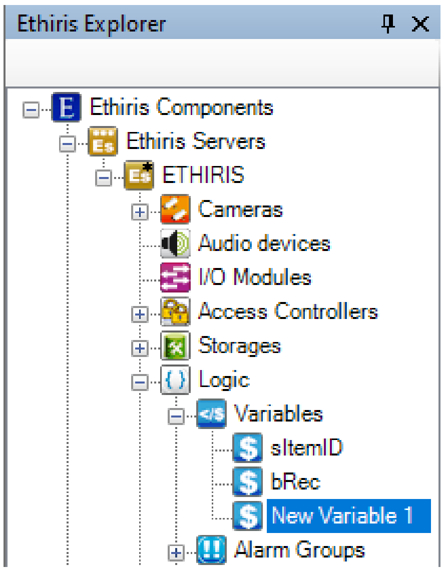
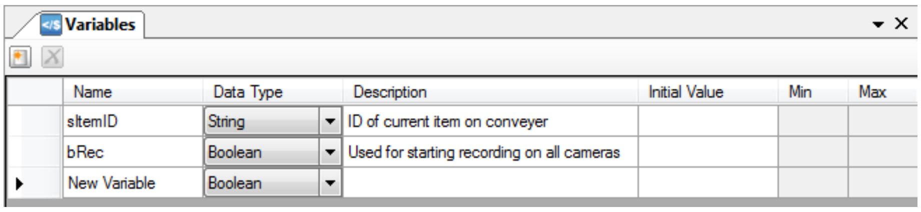


Figure 2.248 New variable added.

Variables panel

Double-clicking the *Variables* node in the treeview opens the corresponding panel.



	Name	Data Type	Description	Initial Value	Min	Max
	sItemID	String	ID of current item on conveyer			
	bRec	Boolean	Used for starting recording on all cameras			
▶	New Variable	Boolean				

Figure 2.249 The Variables panel.

This panel consists of a list of all internal variables currently part of the server configuration.

At the top of the panel, there is a toolbar.

Variables panel toolbar

Figure 2.250 The toolbar in the Variables panel.

Add variable.	Use this button to create a new variable. This is the same as *New->Variable* in the popup menu described above. A new internal variable is immediately added to the server configuration.
Delete variable	Use this button to delete the selected variable(s) from the configuration. You can select more than one variable by using the *Ctrl*-button and/or the *Shift*-button.

Variables panel variable list

The variable list consists of several columns.

Name is the desired name of the variable. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the variable in the list indicating the problem. A variable’s name can be of any length, but it must start with a letter or an underscore (_). The other characters in the variable name can be virtually any, apart from those used by the system (such as &&, ||, +, and -).

It is a good idea to give variables names that indicate their functions. For example, if you want to store the status of a function button, *buttonStatus* may be a good name.

Data type is the desired data type of the variable. The available data types are *Boolean*, *Integer*, *Double,* and *String,* as described at the beginning of this section.

Description is an optional description of the purpose of the variable. It’s a good practice to enter descriptions of your variables. After a while, the purpose may not be so obvious as it was when you first created the variable.

Initial value sets the initial value of the variable. If no initial value is specified the variable will be initialized as follows: Boolean = *false*, Integer = *0*, *Double =* 0.0 *and String = ””.*

Min sets the minimum value for the variable.

Max sets the maximum value for the variable.

2.4.33 Variable node

Under the Variables node in the treeview, there may be some *Variable* nodes.

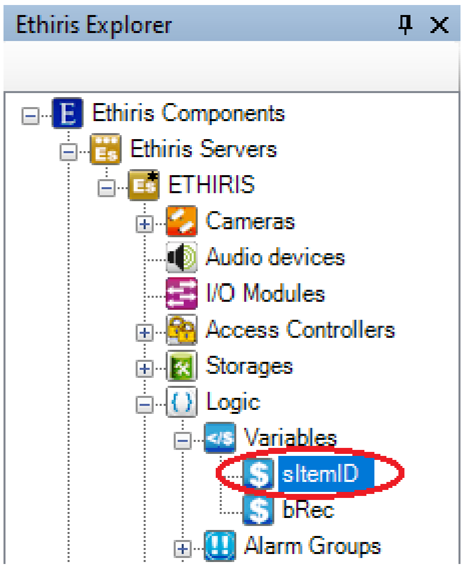


Figure 2.251 A Variable node in Ethiris Explorer treeview.

Variable popup menu

Right-clicking this node brings up a context menu.

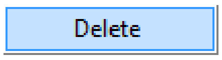


Figure 2.252 The popup menu for a Variable node.

Delete Removes the variable from the server configuration. It is immediately removed from both the treeview and the variable list in the Variables panel.

Variable panel

Double-clicking a *Variable* node in the treeview opens a panel that is the same as the *Variables* panel.

2.4.34 Alarm Groups node

Under the Logic node in the treeview, there is also an *Alarm Groups* node. This is a container node for all alarm groups that are defined in the server.

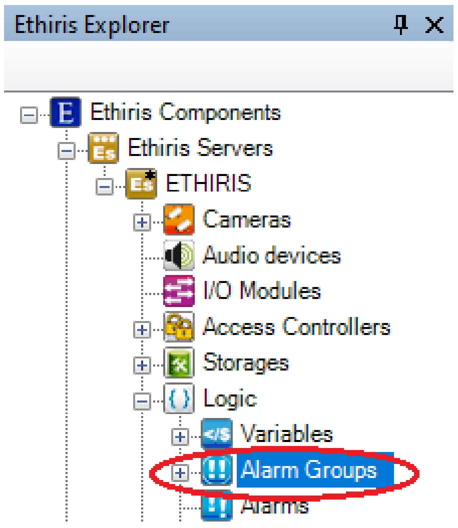


Figure 2.253 The Alarm Groups node in Ethiris Explorer treeview.

Alarms in Ethiris are used to make sure that an operator notices serious problems in the system.

Alarm groups are used for grouping alarms, hence the name. Every alarm belongs to an alarm group. There is a default system alarm group, *System Failure,* where all automatically created system alarms belong.

You can create your own user-defined alarms. These alarms can be put in the *System Failure* group, or you can add a new alarm group for your own alarms.

Each alarm group has a *SumAlarm* signal in the Ethiris data store, which is true if any alarm in the group is active.

Alarm Groups popup menu

Right-clicking this node brings up a context menu.

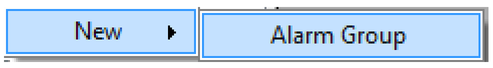


Figure 2.254 The popup menu for the Alarm Groups node.

New->Alarm Group adds a new alarm group to the server configuration. It is immediately visible in the treeview as a new alarm group node. Should you have opened the *Alarm Groups* panel, the new group would be added there too.

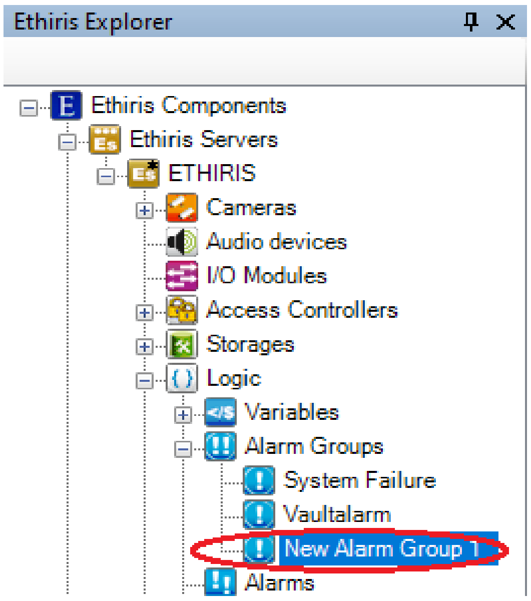
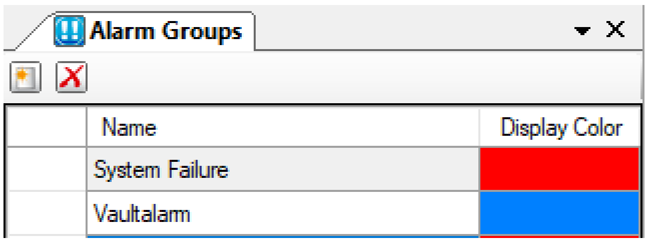


Figure 2.255 New alarm group added.

Alarm Groups panel

Double-clicking the *Alarm Groups* node in the treeview opens the corresponding panel.



Name	Display Color
System Failure	🟥
Vaultalarm	🟦

Figure 2.256 The Alarm groups panel.

This panel consists of a list of all alarm groups currently part of the server configuration.

At the top of the panel, there is a toolbar.

Alarm groups panel toolbar

Figure 2.257 The toolbar in the Alarm groups panel.

New alarm group	Use this button to create a new alarm group. This is the same as *New->Alarm Group* in the popup menu described above. A new alarm group is immediately added to the server configuration.
Delete alarm group	Use this button to delete the selected alarm group(s) from the configuration. You can select more than one group by using the *Ctrl*-button and/or the *Shift*-button. The first group, *System Failure*, is the system's default alarm group and cannot be deleted.

Alarm Groups panel alarm group list

The alarm group list consists of several columns.

Name is the desired name of the alarm group. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the variable in the list indicating the problem. The name cannot be changed for the *System Failure* group.

Display Color is the background color alarms belonging to this group will have in the Alarms list in Ethiris Client. Click in the field to change the current color. The display color cannot be changed for the *System Failure* group.

2.4.35 Alarm group node

Under the Alarm Groups node in the treeview, there may be some *Alarm group* nodes.

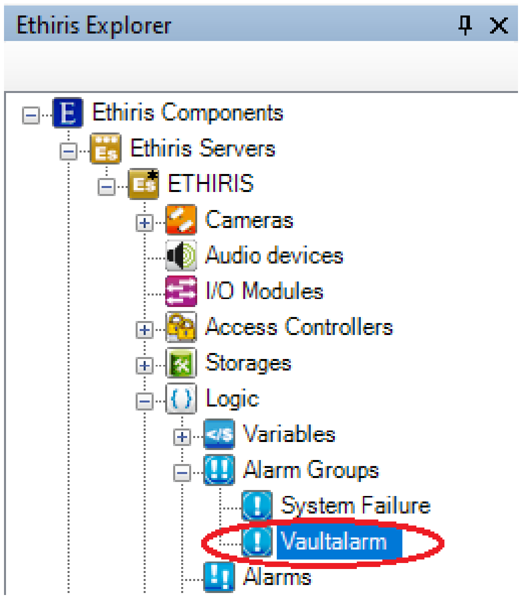


Figure 2.258 An Alarm group node in Ethiris Explorer treeview.

Alarm group popup menu

Right-clicking this node brings up a context menu.

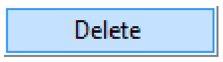


Figure 2.259 The popup menu for an Alarm group node.

Delete Removes the alarm group from the server configuration. It is immediately removed from both the treeview and the alarm group list in the Alarm groups panel.

Alarm group panel

Double-clicking an *Alarm group* node in the treeview opens a panel that is the same as *Alarm Groups* panel.

Alarm group variables

When defining an *Alarm Group*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.260* for an example where an *Alarm group* is selected, and the variables belonging to the alarm group are presented in the lower pane (ringed in).

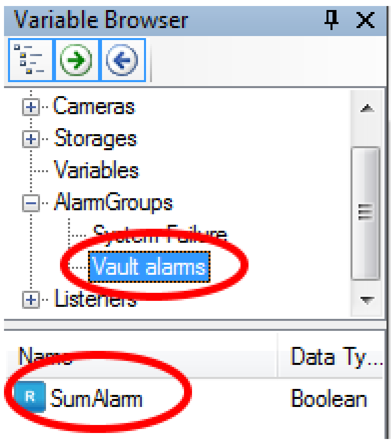


Figure 2.260 Variables for an Alarm group.

SumAlarm is a read-only variable, which is *true* if any alarm belonging to the group is active.

2.4.36 Alarms node

Under the Logic node in the treeview, there is also an *Alarms* node. This is a container node for all custom alarms that are defined in the server.

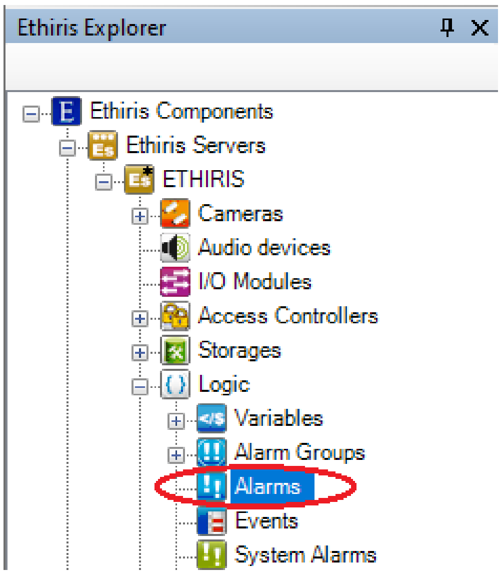


Figure 2.261 The Alarms node in Ethiris Explorer treeview.

Alarms in Ethiris are used to make sure that an operator notices serious problems in the system. There are quite a few *System Alarms* that are automatically created together with objects such as a *camera* or a *storage*.

In this context, *Alarms*, we are dealing with custom alarms, that you as a user can create yourself.

Alarms popup menu

Right-clicking this node brings up a context menu.

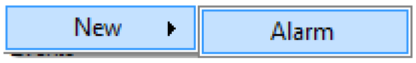


Figure 2.262 The popup menu for the Alarms node.

New->Alarm adds a new alarm to the server configuration. It is immediately visible in the treeview as a new alarm node. Should you have opened the *Alarms* panel, the new alarm would be added there too.

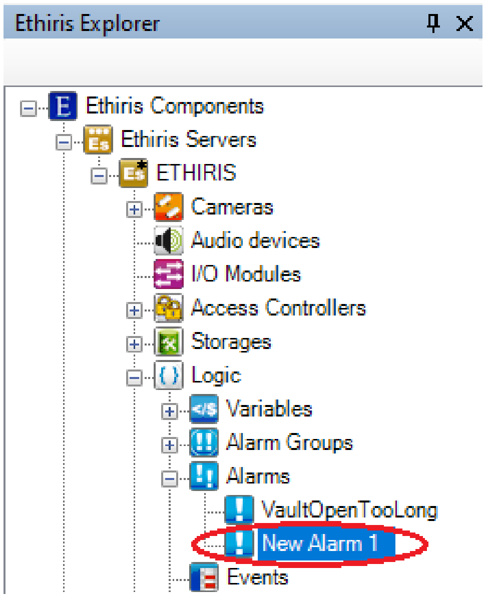
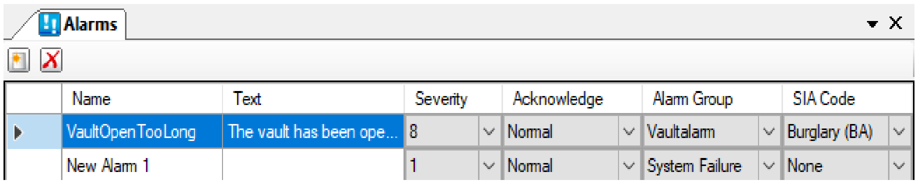


Figure 2.263 New alarm added.

Alarms panel

Double-clicking the *Alarms* node in the treeview opens the corresponding panel.



	Name	Text	Severity	Acknowledge	Alarm Group	SIA Code
▶	VaultOpenTooLong	The vault has been ope...	8	Normal	Vaultalarm	Burglary (BA)
	New Alarm 1		1	Normal	System Failure	None

Figure 2.264 The Alarms panel.

This panel consists of a list of all custom alarms currently part of the server configuration.

At the top of the panel, there is a toolbar.

Alarms panel toolbar

Figure 2.265 The toolbar in the Alarms panel.

New alarm	Use this button to create a new custom alarm. This is the same as *New->Alarm* in the popup menu described above. A new alarm is immediately added to the server configuration.
Delete alarm	Use this button to delete the selected alarm (s) from the configuration. You can select more than one alarm by using the *Ctrl*-button and/or the *Shift*-button.

Alarms panel alarm list

The alarm list consists of several columns.

Name is the desired name of the alarm. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the alarm in the list indicating the problem. The name is how the alarm is identified in various contexts like script and OPC.

Text is the text that appears in the *Alarm list* in Ethiris client. The same text is displayed in the *Event list*, where all status changes for alarms are logged and displayed.

Severity is a value between 1 – 10, where 10 is considered to be most severe, and 1 is least severe. In the Alarm list in Ethiris Client, you can sort by different columns. In that way, you can sort alarms in severity order.

Acknowledge determines how the alarm shall be acknowledged. There are three options;

Normal, which is a default, means that the alarm can be acknowledged both when the alarm still is active, and when it has become inactive, i.e., the alarm condition is no longer true.

Strict means that you cannot acknowledge the alarm as long as it is still active. The alarm condition has to be false before acknowledge is permitted.

Automatic means that the alarm is automatically acknowledged when the alarm becomes inactive. The operator will not have to acknowledge the alarm manually.

Alarm group determines which alarm group the alarm belongs to. In the Alarm list in Ethiris Client, the alarm group is displayed in the group’s display color to make it easier for the operator to distinguish different alarms from each other in the list.

SIA Code This column will only be visible if alarm central functions are enabled in the Ethiris Server panel. Select the SIA code this alarm should report to the alarm central when activated.

2.4.37 Alarm node

Under the Alarms node in the treeview, there may be some *Alarm* nodes.

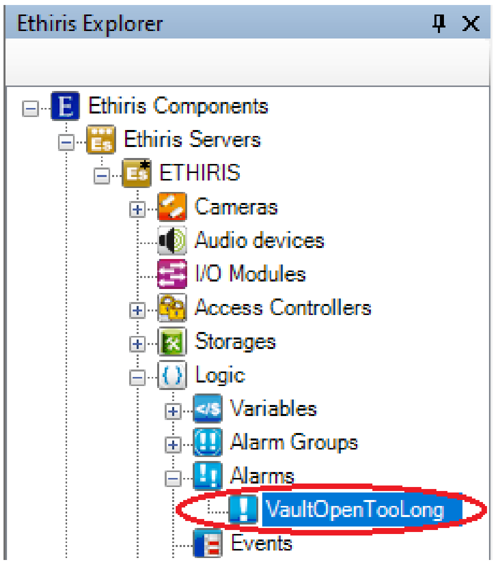


Figure 2.266 An Alarm node in Ethiris Explorer treeview.

Alarm popup menu

Right-clicking this node brings up a context menu.

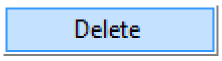
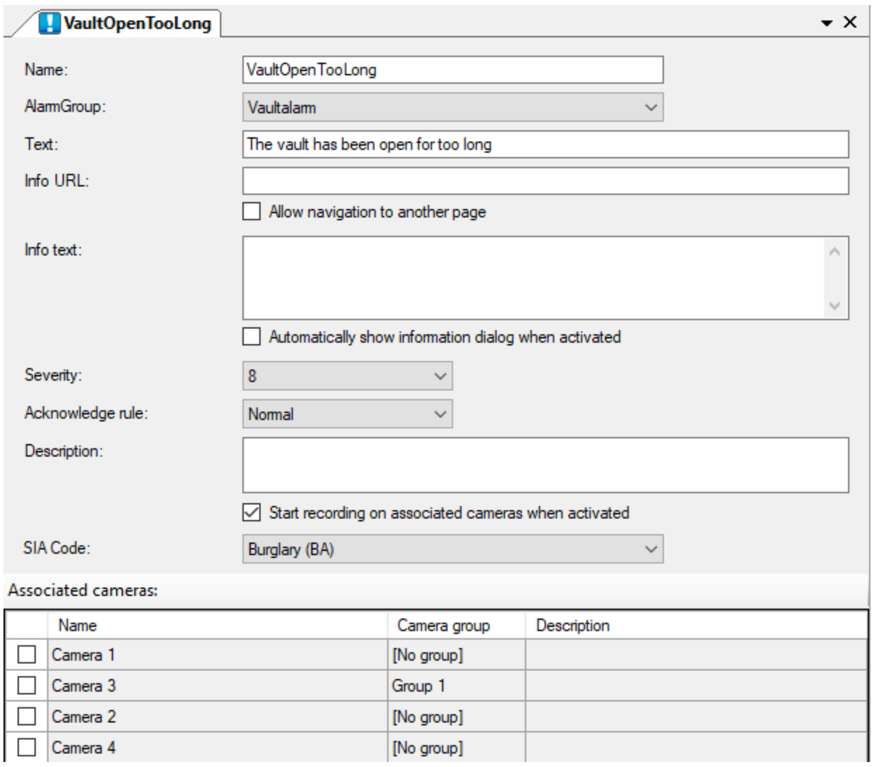


Figure 2.267 The popup menu for an Alarm node.

Delete Removes the alarm from the server configuration. It is immediately removed from both the treeview and the alarm list in the Alarms panel.

Alarm panel

Double-clicking an *Alarm* node in the treeview opens the corresponding panel.


VaultOpenTooLong

Name:	VaultOpenTooLong
AlarmGroup:	Vaultalarm
Text:	The vault has been open for too long
Info URL:	
	☐ Allow navigation to another page
Info text:	
	☐ Automatically show information dialog when activated
Severity:	8
Acknowledge rule:	Normal
Description:	
	☑ Start recording on associated cameras when activated
SIA Code:	Burglary (BA)

Associated cameras:

	Name	Camera group	Description
☐	Camera 1	[No group]	
☐	Camera 3	Group 1	
☐	Camera 2	[No group]	
☐	Camera 4	[No group]	

Figure 2.268 The Alarm panel.

In this panel, you can enter specific settings for the alarm. Most of the fields in this panel are the same as those that are present in the list of alarms in the *Alarms* panel.

Name, *Alarm group*, *Text*, *Severity,* and *Acknowledge rule* is the same as in the *Alarms list* above.

Info URL defines the address to a web-page that can be shown from Ethiris Client. This page can contain information/instructions to the operator regarding handling this alarm. It can be combined with the **Info text** below.

Allow navigation to another page when this box is checked, the operator can navigate using links on the web-page. If not checked, links will not work.

Info text defines information/instructions for the operator regarding this alarm. It can be combined with **Info URL** above.

Automatically show information dialog when activated means that the window with information for the operator is automatically opened when the alarm is activated. In other cases, the operator needs to double-click the information icon in the alarm list in Ethiris Client.

Severity is a value between 1 – 10, where 10 is considered to be most severe, and 1 is least severe. In the Alarm list in Ethiris Client, you can sort by different columns. In that way, you can sort alarms in severity order.

Acknowledge rule determines how the alarm shall be acknowledged. There are three options;

Normal, which is a default, means that the alarm can be acknowledged both when the alarm still is active, and when it has become inactive, i.e., the alarm condition is no longer true.

Strict means that you cannot acknowledge the alarm as long as it is still active. The alarm condition has to be false before acknowledge is permitted.

Automatic means that the alarm is automatically acknowledged when the alarm becomes inactive. The operator will not have to acknowledge the alarm manually.

Description is just a description of the alarm that is just displayed here and in the *Variable Browser* tool window. Still, it is a good idea for future reference to describe the purpose of your custom alarms.

Start recording on associated cameras when activated Checking this box means that associated cameras automatically will record an event as long as the alarm is active. This is the default setting for newly created alarms. In this mode, the regular recording events will not be generated for the associated cameras.

SIA Code This setting is only visible if alarm central functions are activated. Select the SIA code this alarm will report to the alarm central when activated. Currently, Ethiris supports three different types of alarm codes: *Burglary (BA), Fire (FA)* och *Communication error (YC).* The selected alarm code will be transmitted when the alarm is activated. When the alarm is deactivated, a restore code will automatically be sent to the alarm central (different codes depending on selected SIA code).

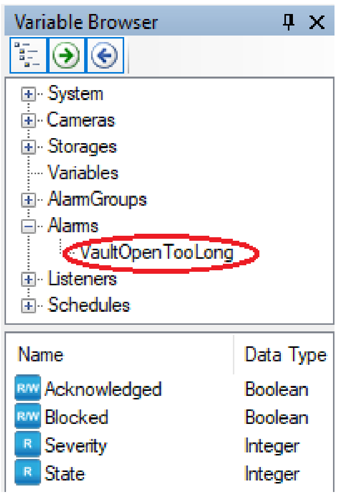
Associated cameras are used for associating desired cameras with the alarm. Check desired cameras in the list to associate them with the alarm. In the *Event list* in Ethiris Client, each status change for an alarm is logged and displayed. If you double-click on such an event, the *Player* is automatically selected with the associated cameras loaded into the player, and the time ruler is initiated with the time of the event. This is a very powerful feature that increases the efficiency of the operator.

Select only cameras belonging to the same camera group This text is shown if any camera group has a different setting for *account identification* and/or *operator integration* than the alarm central has in the tab *Alarm* central in the Ethiris Server panel. If you select cameras from different camera groups anyway, this text will be red to indicate an error.

Alarm variables

When defining an *Alarm*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.269* for an example where an *Alarm* is selected, and the variables belonging to the alarm group are presented in the lower pane (ringed in).



Name	Data Type
Acknowledged	Boolean
Blocked	Boolean
Severity	Integer
State	Integer

Figure 2.269 Variables for an Alarm.

First of all, you can use the alarm variable itself as a *Boolean* value that is *true* when the alarm is active and *false* when the alarm is not active.

Acknowledged is a read/write variable, that is *true* if the alarm is acknowledged. Since the variable is writeable, you can use it to acknowledge the alarm in a script or in some other way by setting the variable to *true*.

Blocked is a read/write variable, that is *true* if the alarm is blocked. Since the variable is writeable, you can use it to block the alarm in a script or in some other way by setting the variable to *true*.

Severity is a read-only variable that is an *integer*. The following values are possible:

0 – Disabled, the alarm is disabled.

1-10 – Severity, The level of the alarm.

State is a read-only variable that is an *integer*. The following values are possible:

0 – Inactive, the alarm is inactive and acknowledged.

1 – Acked, the alarm is active and acknowledged.

2 – Unacked, the alarm is inactive, but not yet acknowledged.

3 – Active, the alarm is active and not yet acknowledged.

4 – Blocked, the alarm is blocked.

2.4.38 Events node

Under the Logic node in the treeview, there is also an *Events* node. This is a container node for all custom events that are defined in the server.

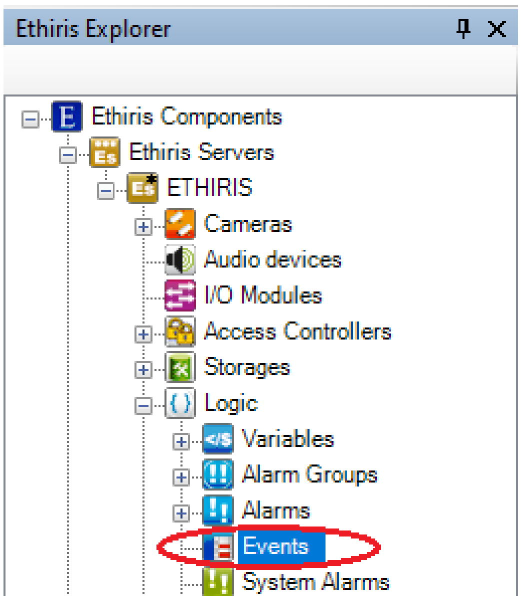


Figure 2.270 The Events node in Ethiris Explorer treeview.

Events in Ethiris are used for logging various events in the system. You can then look back and find out what has happened in the system. Events are not as serious as alarms. Events cannot be acknowledged. The purpose is merely to log important events.

In this context, *Events*, we are dealing with custom events that you, as a user, can create yourself.

Events popup menu

Right-clicking this node brings up a context menu.

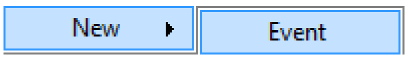


Figure 2.271 The popup menu for the Events node.

New->Event adds a new event to the server configuration. It is immediately visible in the treeview as a new event node. Should you have opened the *Events* panel, the new event would be added there too.

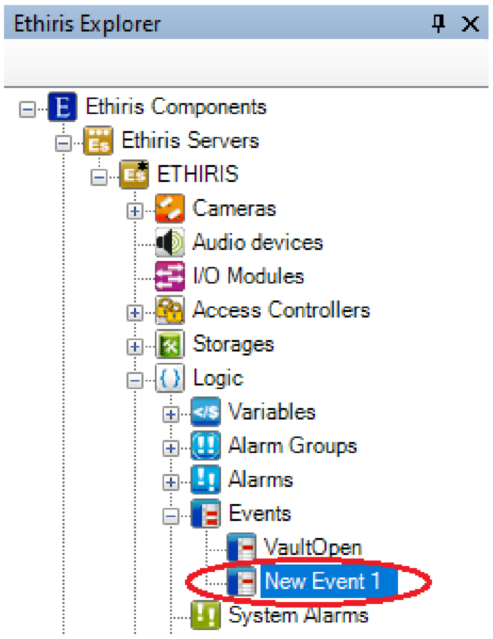
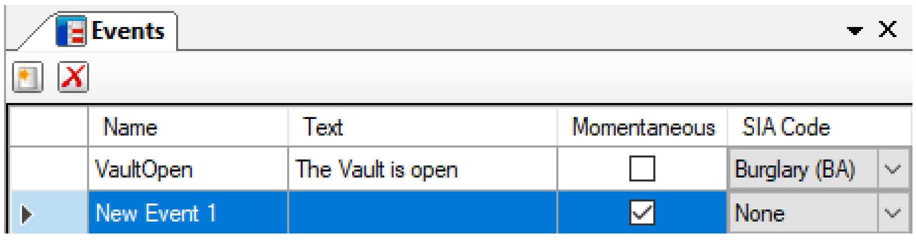


Figure 2.272 New event added.

Events panel

Double-clicking the *Events* node in the treeview opens the corresponding panel.



	Name	Text	Momentaneous	SIA Code
	VaultOpen	The Vault is open	☐	Burglary (BA)
▶	New Event 1		☑	None

Figure 2.273 The Events panel.

This panel consists of a list of all custom events currently part of the server configuration.

At the top of the panel, there is a toolbar.

Events panel toolbar

Figure 2.274 The toolbar in the Events panel.

New event Use this button to create a new custom event. This is the same as *New->Event* in the popup menu described above. A new event is immediately added to the server configuration.

Delete event Use this button to delete the selected event(s) from the configuration. You can select more than one event by using the *Ctrl*-button and/or the *Shift*-button.

Events panel event list

The event list consists of several columns.

Name is the desired name of the event. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the event in the list indicating the problem. The name is how the event is identified in various contexts like script and OPC.

Text is the text that appears in the *Event list* in Ethiris Client.

Momentaneous determines the nature of the event.

When *Momentaneous* is checked, the event is supposed to be irreversible, e.g., *the door has been open*. This kind of event cannot be undone. In the *Event list* in Ethiris Client, such an event has no *Status*. As a default, momentaneous is checked for new events.

When *Momentaneous* is unchecked, the event is supposed to have two states; *active* and *inactive*. An example is *The door is open*. In this case, the event can be ”undone”, i.e., the door can be closed again. This kind of event will be logged both when it becomes active and when it becomes inactive.

SIA Code The column will only be visible if alarm central functions are enabled in the Ethiris Server panel. Select the SIA code this event will report to the alarm central when activated.

If the event is *Momentaneous*, the selected SIA code will be reported to the alarm central when the event is activated. Nothing will be reported when the event is inactivated.

If the event is *Momentaneous,* the selected SIA code is reported to the alarm central when the event is activated. When the event is inactivated, the corresponding restore code will be reported to the alarm central. For *BA*, *BH* will be reported as a restore code. For *FA*, *FH* is the restore code, and finally, for *YC*, *YK* is the restore code that will be reported.

2.4.39 Event node

Under the Events node in the treeview, there may be some *Event* nodes.

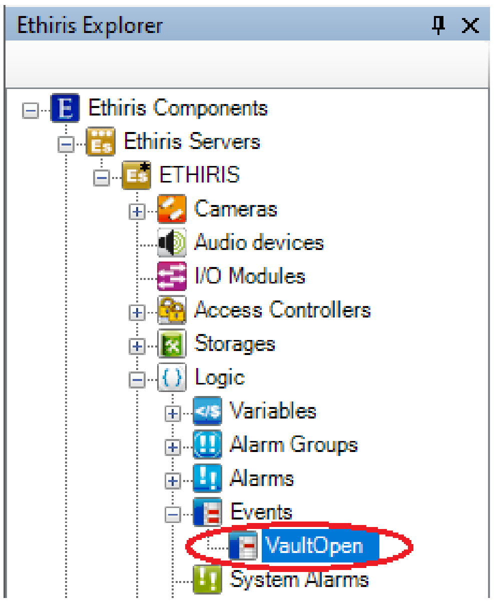


Figure 2.275 An Event node in Ethiris Explorer treeview.

Event popup menu

Right-clicking this node brings up a context menu.

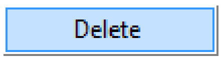
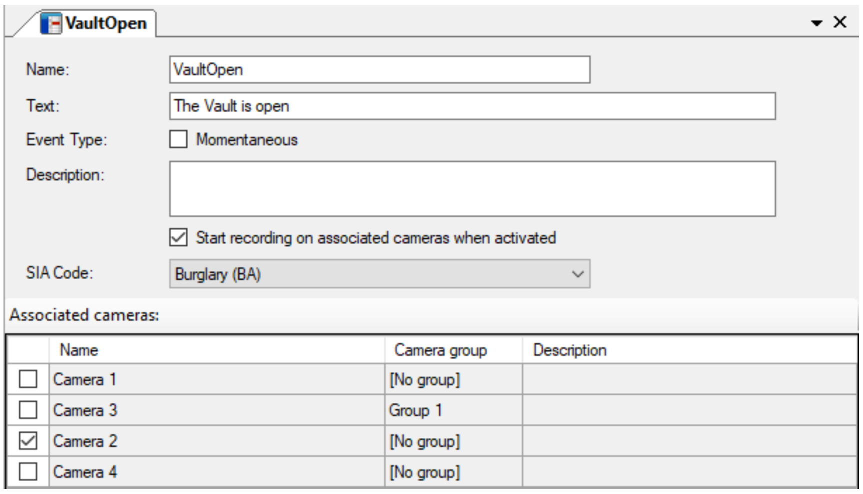


Figure 2.276 The popup menu for an Event node.

Delete Removes the event from the server configuration. It is immediately removed from both the treeview and the event list in the Events panel.

Event panel

Double-clicking an *Event* node in the treeview opens the corresponding panel.



VaultOpen

Name: VaultOpen
Text: The Vault is open
Event Type: ☐ Momentaneous
Description:
☑ Start recording on associated cameras when activated
SIA Code: Burglary (BA)

Associated cameras:

	Name	Camera group	Description
☐	Camera 1	[No group]	
☐	Camera 3	Group 1	
☑	Camera 2	[No group]	
☐	Camera 4	[No group]	

Figure 2.277 The Event panel.

In this panel, you can enter specific settings for the event. Most of the fields in this panel are the same as those that are present in the list of events in the *Events* panel.

Name, *Text,* and *Event type* are the same as in the Events list above.

Description is just a description of the event that is just displayed here and in the *Variable Browser* tool window. Still, it is a good idea for future reference to describe the purpose of your custom events.

Start recording on associated cameras when activated Checking this box means that associated cameras automatically will record an event as long as the event is active. This is the default setting for newly created events. In this mode, the regular recording events will not be generated for the associated cameras.

SIA Code This setting is only visible if alarm central functions are activated. Select the SIA code this event will report to the alarm central when activated. Currently, Ethiris supports three different types of alarm codes: *Burglary (BA), Fire (FA)* och *Communication error (YC).* The selected alarm code will be transmitted when the event is activated. If the event is not *Momentaneous* and the event is deactivated, a restore code will automatically be sent to the alarm central (different codes depending on selected SIA code).

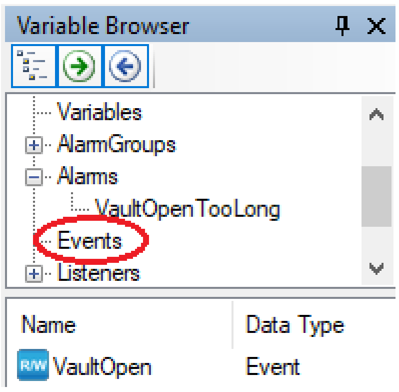
Select only cameras belonging to the same camera group This text is shown if any camera group has a different setting for *account identification* and/or *operator integration* than the alarm central has in the tab *Alarm* central in the Ethiris Server panel. If you select cameras from different camera groups anyway, this text will be red to indicate an error.

Associated cameras are used for associating desired cameras with the event. Check desired cameras in the list to associate them with the event. In the *Event list* in Ethiris Client, the associated cameras are listed in the *Objects* column. If you double-click on an event with associated cameras, the *Player* is automatically selected with the associated cameras loaded into the player, and the time ruler is initiated with the time of the event. This is a very powerful feature that increases the efficiency of the operator.

Event variables

When defining an *Event*, a *variable* is automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.278* for an example where *Events* are selected, and all the event variables are presented in the lower pane (ringed in).



Name	Data Type
VaultOpen	Event

Figure 2.278 Event variables.

You can use the event variable itself as a *Boolean* value that is *true* when the event is active and *false* when the event is not active.

2.4.40 System Alarms node

Under the Logic node in the treeview, there is also a *System Alarms* node. This is a container node for all system alarms that are automatically created in the server.

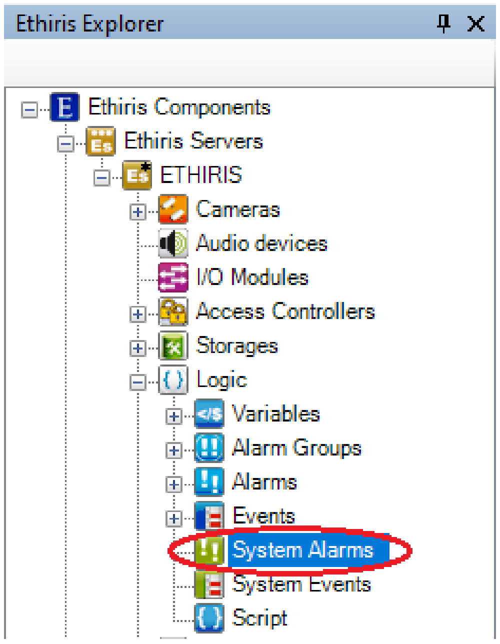


Figure 2.279 The System Alarms node in Ethiris Explorer treeview.

Alarms in Ethiris are used to make sure that an operator notices serious problems in the system.

There are quite a few system alarms that are automatically created in Ethiris.

Some of them are always created, and some will be created along with the creation of other objects like the Camera communication error alarm is created for each camera that is added to the configuration.

All the system alarms belong to the System Failure alarm group, which also is created automatically.

The following 6 alarms are automatically created for each Storage device in the system. There is always a storage device named Default, but you can also add additional storage devices if desired.

Storage Device Access Failed is activated when the storage device exists, but Ethiris Server, for some reason, gets an access failure error trying to access the device.

Storage Device Full. Each storage device has a Max storage size for recorded video. If this size is not sufficient and all available storage space has been filled, this alarm will be activated. It implies that either too much video is recorded or that the time for automatic clean up of the old video is too long.

Storage Device Disk Full is more serious. This happens when there is less than 1 GiB of storage space remaining on the actual disk that the storage device uses.

Storage Device Queue Frames Dropped is activated when Ethiris Server stores frames in such a pace that the disk cannot write data on disk fast enough. Ethiris buffers frames in memory before they are written to disk. Depending on the amount of RAM available in the system, the queue of frames about to be written to disk may become too long. Instead of exhausting the memory and potentially crash the whole system, frames are dropped.

Storage Device Missing is activated if a storage device for some reason suddenly does not exist. It may be due to a disk that has been removed, e.g., a USB-device (god forbid).

Storage Device Insufficient Disk Space is activated if there is not sufficient space on the disk where the storage device is defined. In other words, *Used Disk Space* + *Disk Free Space* < *Max storage size* in the *Server Storages* panel.

For each camera in the Ethiris Server configuration the following two alarms will be created:

Camera Recording Error is activated if you have enabled *Recording supervision* for the camera, and there has been no recording within the specified time.

Camera Communication Error is activated if Ethiris Server has no contact with the camera. It happens if the network cable is unplugged or if the camera is simply powered off.

For each audio device in the Ethiris Server configuration the following alarm will be created:

Audio device communication error is activated if Ethiris can not get in contact with the audio device. This can happen if the network cable is disconnected or the audio device is switched off.

For each I/O module in the Ethiris Server configuration the following alarm will be created:

I/O module communication error is activated if Ethiris can not get in contact with the I/O module. This can happen if the network cable is disconnected or the I/O module is switched off.

For each Access Controller in the Ethiris Server configuration the following alarms will be created:

Access Controller communication error is activated if Ethiris can not communicate with the Access Controller. This can happen if the network cable is disconnected or the I/O module is switched off.

Casing open is activated if the Access Controller casing is opened.

Configuration is incorrect in Access Controller is activated if Ethiris discovers that the configuration in the Access Controller has been changed without refreshing it in Ethiris Admin. The Access Controller will not work correctly in Ethiris. The problem can easily be corrected by connecting to the Ethiris Server with Ethiris Admin and refresh the Access Controller in the configuration. Don’t forget to save the configuration.

The following 9 alarms are always created in an Ethiris system:

Script Runtime Error is activated whenever an error occurs in Ethiris Script. Ethiris script is interpreted in runtime, and errors can occur due to syntax errors as well as for logical errors that depend on various circumstances. It can happen for many different reasons.

System Internal Error should normally not happen. If it does, more information about the causes can be found in the Ethiris log files.

System Memory Exhausted is activated when the system is about to run out of available RAM. One reason can be that too many cameras have a too long pre-alarm buffer on event recording.

System CPU Exhausted is activated when the CPU runs on 95% or more for more than two minutes and is deactivated when returning below 90% for more than 24 seconds.

System GPU Exhausted is activated when the GPU runs on 98% or more for more than two minutes and is deactivated when returning below, for more than 24 seconds.

System Restarted means that Ethiris Server has been automatically restarted due to a problem. Information about the cause can be found in the Ethiris Server log files.

Ethiris client communication error. This can happen when there is a problem in the communication between an Ethiris Client and the Ethiris Server.

Configuration needs to be upgraded. This probably rarely occurs. If it occurs, it means that the Ethiris Server configuration file (Conf.esc) is of an older version than the Ethiris Server. Normally the configuration is automatically upgraded both when you upgrade via a new installation and by restoring a backup. But somebody might have restored an older configuration file manually and then restarted Ethiris Server. When Ethiris Server is started, it tries to read the current configuration and if it may discover that the configuration is too old. In this situation, the alarm is activated and appears in all connected clients.

The problem is easily fixed by connecting to the Ethiris Server via an Ethiris Admin (with the same version as the Ethiris Server). Ethiris Admin will automatically upgrade the configuration which the user then can save.

NTP failure, see log file for details. This can happen when there is a problem with NTP.

The following two alarms are created for each OPC Server that is connected to the Ethiris Server:

Driver Communication Error is activated whenever there is a problem in the communication between Ethiris Server and the OPC Server.

Driver Configuration Error is activated if any problem in the OPC configuration is detected. For example, if an OPC group contains no items.

The following two alarms are created for each Listener that is created:

Listener Communication Error is activated whenever there is a problem in the communication between Ethiris Server and the external equipment that the listener is connected to.

Listener Configuration Error is activated if any problem in the listener configuration is detected. For example, if the answer from the external equipment is unexpected.

The following three alarms are created if a UPS is configured:

UPS Communication Error is activated whenever there is a problem in the communication between Ethiris Server and the UPS.

UPS Configuration Error is activated if any problem in the UPS configuration is detected.

UPS running on battery is activated when the UPS loses power and begins to run on batteries.

The following alarms are created if an Alarm central is configured:

Alarm central Communication Error is activated whenever there is a problem in the communication between Ethiris Server and the Alarm central.

Alarm central Secondary Communication Error is activated whenever there is a problem in the communication between Ethiris Server and the secondary Alarm central if configured.

The following alarms are created if an SIA Server is configured:

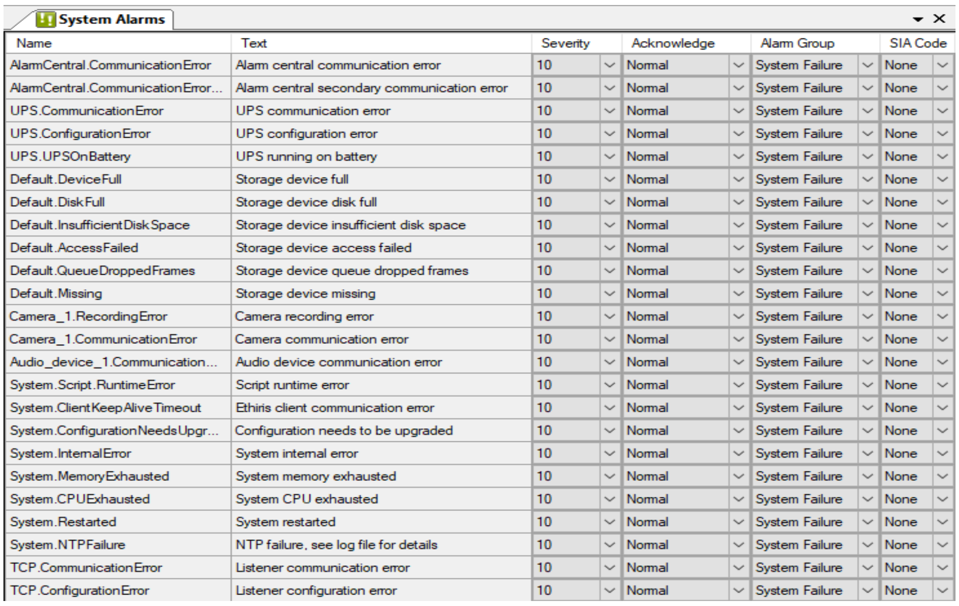
Alarm central Communication Error is activated whenever there is a problem in communication with the SIA server.

The following alarm is created for each *Remote Client* that is created in the list of remote clients:

Remote Client Communication Error is activated if there is a communication problem between Ethiris Server and the client. For each client defined in the list, you can set a *Connection supervision* time. By default, this supervision is *Off*, in which case, there will be no alarm. If a time is specified, the alarm will be activated after Ethiris has discovered the connection problem plus the time specified. If someone closes the client, Ethiris will discover that immediately, but if someone, e.g., pulls out the network cable, it can take up to 45 seconds before discovery.

System Alarms panel

Double-clicking the *System Alarms* node in the treeview opens the corresponding panel.



System Alarms

Name	Text	Severity	Acknowledge	Alarm Group	SIA Code
AlarmCentral.CommunicationError	Alarm central communication error	10	Normal	System Failure	None
AlarmCentral.CommunicationError...	Alarm central secondary communication error	10	Normal	System Failure	None
UPS.CommunicationError	UPS communication error	10	Normal	System Failure	None
UPS.ConfigurationError	UPS configuration error	10	Normal	System Failure	None
UPS.UPSOnBattery	UPS running on battery	10	Normal	System Failure	None
Default.DeviceFull	Storage device full	10	Normal	System Failure	None
Default.DiskFull	Storage device disk full	10	Normal	System Failure	None
Default.InsufficientDiskSpace	Storage device insufficient disk space	10	Normal	System Failure	None
Default.AccessFailed	Storage device access failed	10	Normal	System Failure	None
Default.QueueDroppedFrames	Storage device queue dropped frames	10	Normal	System Failure	None
Default.Missing	Storage device missing	10	Normal	System Failure	None
Camera_1.RecordingError	Camera recording error	10	Normal	System Failure	None
Camera_1.CommunicationError	Camera communication error	10	Normal	System Failure	None
Audio_device_1.Communication...	Audio device communication error	10	Normal	System Failure	None
System.Script.RuntimeError	Script runtime error	10	Normal	System Failure	None
System.ClientKeepAliveTimeout	Ethiris client communication error	10	Normal	System Failure	None
System.ConfigurationNeedsUpgr...	Configuration needs to be upgraded	10	Normal	System Failure	None
System.InternalError	System internal error	10	Normal	System Failure	None
System.MemoryExhausted	System memory exhausted	10	Normal	System Failure	None
System.CPUExhausted	System CPU exhausted	10	Normal	System Failure	None
System.Restarted	System restarted	10	Normal	System Failure	None
System.NTPFailure	NTP failure, see log file for details	10	Normal	System Failure	None
TCP.CommunicationError	Listener communication error	10	Normal	System Failure	None
TCP.ConfigurationError	Listener configuration error	10	Normal	System Failure	None

Figure 2.280 The System Alarms panel.

This panel consists of a list of all system alarms currently part of the server configuration. The content of the list depends on which other objects are defined in the system. Some system alarms are always there, and some are created together with other objects, as described above.

System Alarms panel alarm list

The alarm list consists of several columns. All but *Severity* is read-only for system alarms.

Name is the name of the alarm. The name is how the alarm is identified in various contexts like script and OPC.

Text is the text that appears in the *Alarm list* in Ethiris client. The same text is displayed in the *Event list*, where all status changes for alarms are logged and displayed.

Severity is a value between 1 – 10, where 10 is considered to be most severe, and 1 is least severe. In the Alarm list in Ethiris Client, you can sort by different columns. That way, you can sort alarms in severity order.

If you select *Severity = Disabled*, it means that the alarm is disabled. In this case, it will not be visible at all in Ethiris Client.

Acknowledge determines how the alarm shall be acknowledged. There are three options;

Normal, which is a default, means that the alarm can be acknowledged both when the alarm still is active, and when it has become inactive, i.e., the alarm condition is no longer true.

Strict means that you cannot acknowledge the alarm as long as it is still active. The alarm condition has to be false before acknowledge is permitted.

Automatic means that the alarm is automatically acknowledged when the alarm becomes inactive. The operator will not have to acknowledge the alarm manually.

Note that it is not possible to change *Acknowledge* for a system alarm.

Alarm group determines which alarm group the alarm belongs to. In the Alarm list in Ethiris Client, the alarm group is displayed in the group's display color to make it easier for the operator to distinguish different alarms from each other in the list. A system alarm always belongs to the *System Failure* alarm group.

SIA Code This column will only be visible if alarm central functions are enabled in the Ethiris Server panel. Select the SIA code this alarm should report to the alarm central when activated.

2.4.41 System Events node

Under the Logic node in the treeview, there is also a *System Events* node. This is a container node for all system events that are automatically created in the server.

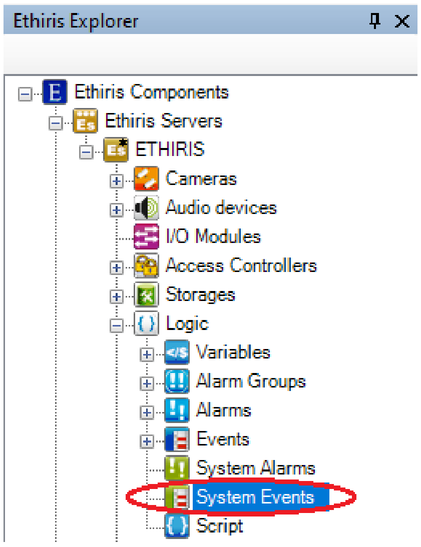


Figure 2.281 The System Events node in Ethiris Explorer treeview.

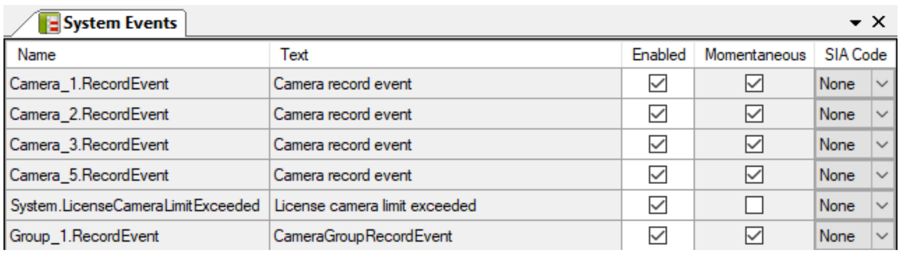
Events are simply events in the system which are not serious enough to be considered Alarms. They are an excellent tool to provide information about what has happened in the system.

Some system events are automatically created in Ethiris. The most common is the *Camera Record Event*. Each time an event-recording starts on a camera, it is logged and displayed in the Events panel in Ethiris Client.

Another system event is the *License Camera Limit Exceeded* event. This should rarely happen, but if it does, it indicates that the maximum number of cameras allowed by the current license has been exceeded. The only way this can happen is when you have a Premium license level, and the number of Active cameras exceeds the license limit. The Premium level accepts that more than the allowed number of cameras are entered into the Server configuration, but only the allowed number of cameras must be active at the same time.

System Events panel

Double-clicking the *System Events* node in the treeview opens the corresponding panel.



Name	Text	Enabled	Momentaneous	SIA Code
Camera_1.RecordEvent	Camera record event	☑	☑	None
Camera_2.RecordEvent	Camera record event	☑	☑	None
Camera_3.RecordEvent	Camera record event	☑	☑	None
Camera_5.RecordEvent	Camera record event	☑	☑	None
System.LicenseCameraLimitExceeded	License camera limit exceeded	☑	☐	None
Group_1.RecordEvent	CameraGroupRecordEvent	☑	☑	None

Figure 2.282 The System Events panel.

This panel consists of a list of all system events currently part of the server configuration. The content of the list depends on which cameras are defined in the system.

System Events panel event list

The events list consists of several columns. All, except for *Enabled* and *SIA Code*, are read-only for system events.

Name is the name of the event. The name is how the event is identified in various contexts like script and OPC.

Text is the text that appears in the *Event list* in Ethiris Client.

Enabled is ticked as default. If you don't want this event to be logged in the event list, you can remove the tick and make the event disabled.

Momentaneous determines the nature of the event.

When *momentaneous* is checked the event is supposed to be irreversible. This kind of event cannot be undone. In the *Event list* in Ethiris Client, such an event has no *Status*. The *RecordEvent* event for a camera is momentaneous since it cannot be undone.

When *momentaneous* is unchecked, the event is supposed to have two states; *active* and *inactive*. This kind of event will be logged both when it becomes active and when it becomes inactive. The system event *License Camera Limit Exceeded* is not momentaneous, since it can be both active and inactive.

Note that it is not possible to change *Momentaneous* for a system event.

SIA Code The column will only be visible if alarm central functions are enabled in the Ethiris Server panel. Select the SIA code this event will report to the alarm central when activated.

If the event is *Momentaneous*, the selected SIA code will be reported to the alarm central when the event is activated. Nothing will be reported when the event is inactivated.

If the event is *Momentaneous,* the selected SIA code is reported to the alarm central when the event is activated. When the event is inactivated, the corresponding restore code will be reported to the alarm central. For *BA*, *BH* will be reported as restore code, for *FA*, *FH* is the restore code that will be reported.

2.4.42 Script node

Under the Logic node in the treeview, there is a *Script* node. This node contains all script code for the Ethiris Server. Ethiris script is extremely flexible and can be used to create custom functionality in the system.

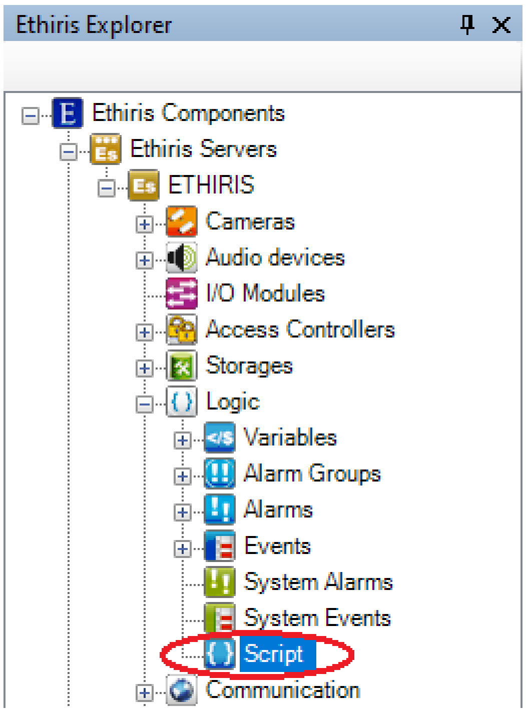
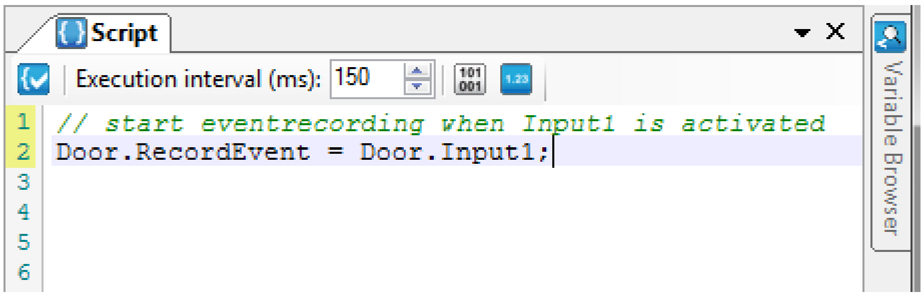


Figure 2.283 The Script node in Ethiris Explorer treeview.

Script panel

Double-clicking the *Script* node in the treeview opens the corresponding panel.



```
// start eventrecording when Input1 is activated
Door.RecordEvent = Door.Input1;
```

Figure 2.284 The Script panel.

This panel consists of two parts; the *Script editor* to the left and the *Variable Browser* tool window docked to the right.

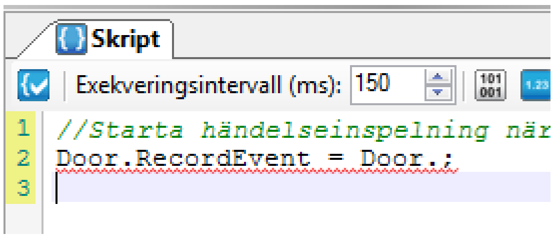
The script editor is where you enter the script.

Scripts in Ethiris are written in a programming language that is a subset of the language ECMAScript. ECMAScript is the common standard on which JavaScript and JScript are based. These languages are used on the World Wide Web to make websites more flexible, and they are well-known to many people.

This section assumes basic knowledge of ECMAScript. Anyone who has used similar languages such as Java, C, or C++, will feel at home and get going and write advanced scripts fast.

You can use a JavaScript or JScript manual to learn the principles of programming with ECMAScript. As we mentioned above, these languages are based on the same standard.

At regular intervals during script editing, an automatic syntax control is performed. Any discovered syntax error is highlighted in the script text with a curly red underline.

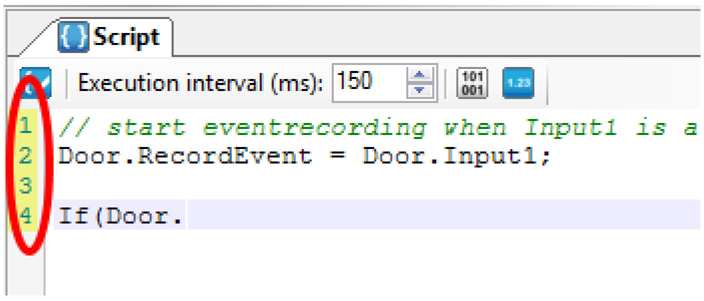


```
1 //Starta händelseinspelning när
2 Door.RecordEvent = Door.;
3 
```

Figure 2.285 Script with a syntax error

There are row numbers to the left in the script editor. This makes it a snap to locate the offending statement from the row number that is logged in the Ethiris Server log file if a script execution error occurs.

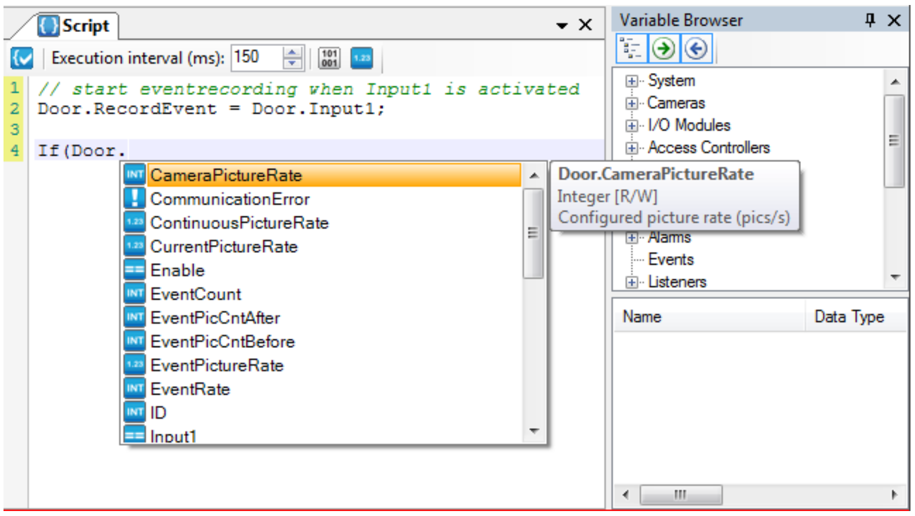
The script editor will also highlight the text that has been changed since the editor was opened; this is done with a yellow overstrike of the row numbers in the left column of the changed rows.



```
1 // start eventrecording when Input1 is a
2 Door.RecordEvent = Door.Input1;
3 
4 If(Door.
```

Figure 2.286 Row numbers in the script editor

The script editor also implements *code completion*, which means that as you start typing the name of an object, a list is shown that contains the name of all objects that matches what you have written so far. In the same manner, a list of all matching properties is shown after you have written a period (.) after an object name.



```
1 // start eventrecording when Input1 is activated
2 Door.RecordEvent = Door.Input1;
3 
4 If(Door.
```

CameraPictureRate
CommunicationError
ContinuousPictureRate
CurrentPictureRate
Enable
EventCount
EventPicCntAfter
EventPicCntBefore
EventPictureRate
EventRate
ID
Input1

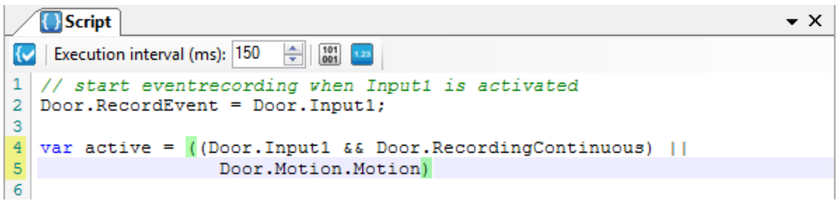
Door.CameraPictureRate
Integer [R/W]
Configured picture rate (pics/s)

Figure 2.287 Code completion for a camera

As you can see, you will also get a description of the selected property.

The list of property names can be opened manually by putting the cursor on the object name and pressing <CTRL> + space.

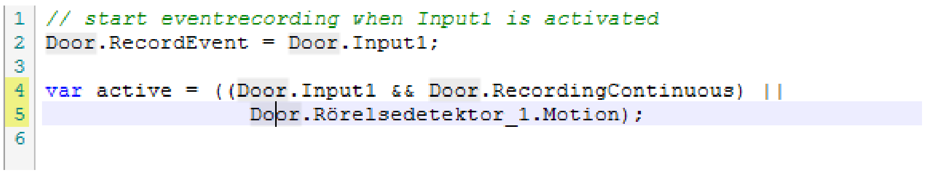
When writing code, the editor helps in a few ways. When writing parentheses, the editor will show which parentheses belong together with a green overstrike. The same will happen if you just put the cursor on the parenthesis in the script text.



```
// start eventrecording when Input1 is activated
Door.RecordEvent = Door.Input1;

var active = ((Door.Input1 && Door.RecordingContinuous) ||
                    Door.Motion.Motion)
```

Figure 2.288 Parentheses belonging together gets highlighted

When you put the cursor on a word in the editor, all occurrences of that word will be highlighted in grey.

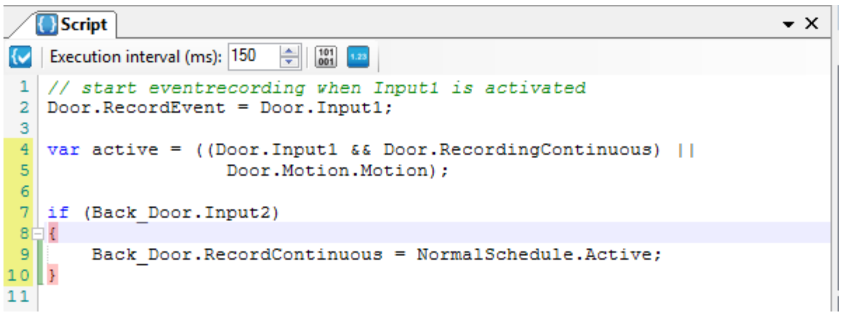


```
// start eventrecording when Input1 is activated
Door.RecordEvent = Door.Input1;

var active = ((Door.Input1 && Door.RecordingContinuous) ||
                    Door.Rörelsedetektor_1.Motion);
```

Figure 2.289 All occurrences of a word gets highlighted automatically

If you write a left brace and press *Enter,* the following text will be automatically indented to make it easier to read the code.

A green mark in the left column shows which rows belong together. You can also hide the part of the code inside braces by clicking the small minus sign by the top brace. Matching braces are marked with red overstrike.



```
// start eventrecording when Input1 is activated
Door.RecordEvent = Door.Input1;

var active = ((Door.Input1 && Door.RecordingContinuous) ||
                    Door.Motion.Motion);

if (Back_Door.Input2)
{
    Back_Door.RecordContinuous = NormalSchedule.Active;
}
```

Figure 2.290 Automatic indent of code

You can change the indent of a part of the code by selecting several text rows and then press <TAB> to increase the indent, or <Shift> + <TAB> to decrease it.

Script panel toolbar

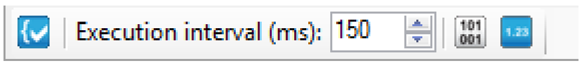


Figure 2.291 The toolbar in the Script editor panel.

Validate script syntax	Use this button to validate the syntax of the script in the script editor. If the syntax is OK, the whole background of the script editor is colored in green. If a syntax error is found, the row where the error is found is marked with a red underline. Note that this is not a complete validation of the script. The script syntax is validated. At this point, it is not possible to validate object names. Ethiris script is interpreted when it runs in Ethiris Server. Validation is done when the code is executed. Any runtime script error is indicated in all connected Ethiris Clients as an alarm.
Execution interval	The execution interval determines how often the script will execute. The value is in milliseconds. The script code is executed from beginning to end over and over again. As a default, this is done every 150 ms. If there is a lot of script code, the execution may take longer than the specified execution interval. Then the execution starts from the beginning as soon as it has executed the whole script. This means that the script always is completely executed.
Show Debug window	Use this button to open the *Debug* window. In the debug window, you can see information printed from the script using ”Debug.Print” or ”alert” as well as potential runtime errors in the script.
Show Watch panel	Use this button to open the *Watch* panel. The watch panel is very useful when trying to understand what happens in the script. Here you can watch current values for desired variables.

In case of the current configuration being converted from an earlier version of Ethiris and if there were a script then, a checkbox, *Compatibility mode*, will be shown to the right in the toolbar. The checkbox will be checked after the conversion.

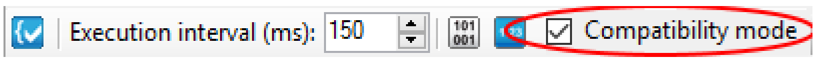


Figure 2.292 Toolbar with compatibility mode

Compatibility mode means that scripts written in previous versions of Ethiris will continue to work as before. Note that the new function described in *3.5.1 Events in DataStore variables* won’t work. Read more in release notes for Ethiris 8, in section *Changed behavior*.

Script editor popup menu

Right-click in the script editor opens a popup menu.

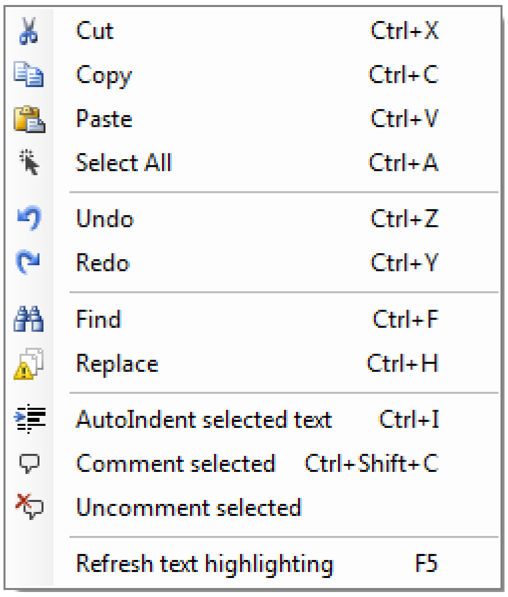


Figure 2.293 Popup menu in the script editor

Cut copies all selected text for later paste and removes the selection.

Copy copies all selected text for later pasting.

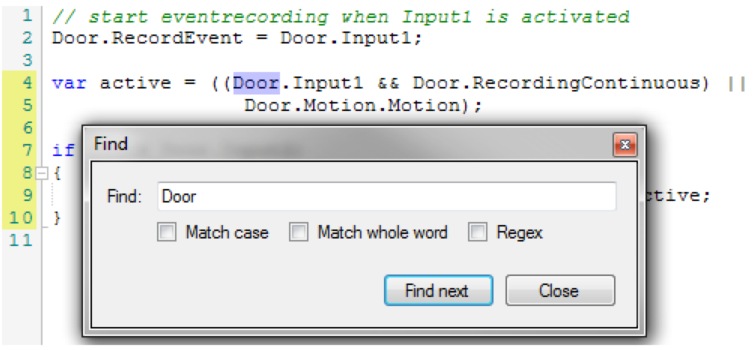
Paste inserts the text that was previously copied at the cursor.

Select all selects all text in the editor.

Undo revokes the latest change. It can be done in several steps.

Redo performs changes again that was previously undone.

Find shows a dialog where you can search for text in the script. Texts that are found will be highlighted in purple. Click on *Find next* to locate the next match.



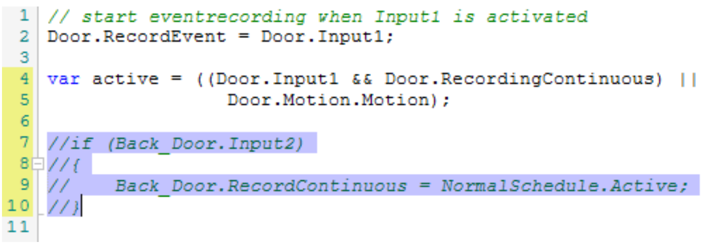
```
// start eventrecording when Input1 is activated
Door.RecordEvent = Door.Input1;

var active = ((Door.Input1 && Door.RecordingContinuous) ||
                Door.Motion.Motion);


if
{
                                                    tive;
}

```

Figure 2.294 Find dialog in the script editor

Replace displays a dialog where you can specify a text to search for and another text that will replace the search text. Note that the replacement will take place either in the current selection (purple highlighted text) or in the complete script text if no text is selected.

Autoindent selected text recalculates the indent for the selected text.

Comment selected comment out the selected rows. Useful if you would like to comment out many rows.



```
// start eventrecording when Input1 is activated
Door.RecordEvent = Door.Input1;

var active = ((Door.Input1 && Door.RecordingContinuous) ||
                Door.Motion.Motion);

//if (Back_Door.Input2)
//{
//      Back_Door.RecordContinuous = NormalSchedule.Active;
//}
```

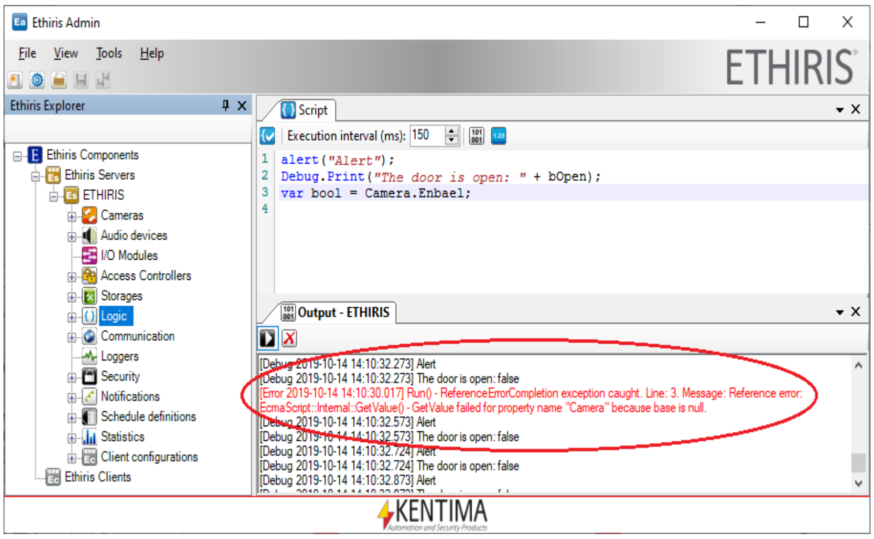
Figure 2.295 Selected rows commented out

Uncomment selected removes the comment (//) on the selected rows. Useful if you would like to uncomment many rows. F.i. to reenable an earlier commented script.

Refresh text highlighting forces an update of the highlighting of text in the script. This is normally done automatically.

Debug Window

In the Debug window, you can see debug information printed from the script. This can make it easier to debug the script and see what happens during runtime. It is also possible to print the value of variables in the debug window, for example, the value of bOpen: `Debug.Print("The door is open: " + bOpen)`. Note the misspelling of the Camera's property "Enable" below, which leads to a runtime error.



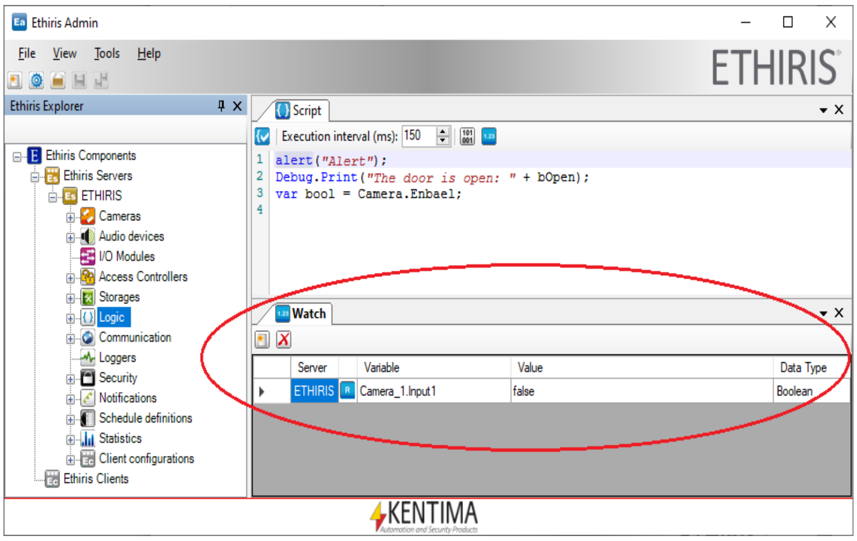
```
alert("Alert");
Debug.Print("The door is open: " + bOpen);
var bool = Camera.Enbael;
```

```
[Debug 2019-10-14 14:10:32.273] Alert
[Debug 2019-10-14 14:10:32.273] The door is open: false
[Error 2019-10-14 14:10:30.017] Run() - ReferenceErrorCompletion exception caught. Line: 3. Message: Reference error:
EcmaScript::Internal::GetValue() - GetValue failed for property name "Camera" because base is null.
[Debug 2019-10-14 14:10:32.573] Alert
[Debug 2019-10-14 14:10:32.573] The door is open: false
[Debug 2019-10-14 14:10:32.724] Alert
[Debug 2019-10-14 14:10:32.724] The door is open: false
[Debug 2019-10-14 14:10:32.873] Alert
```

Figure 2.296 The Debug window is showing a few messages

Watch panel

When you have decided to show the Watch panel, it is located at the bottom under the script editor.

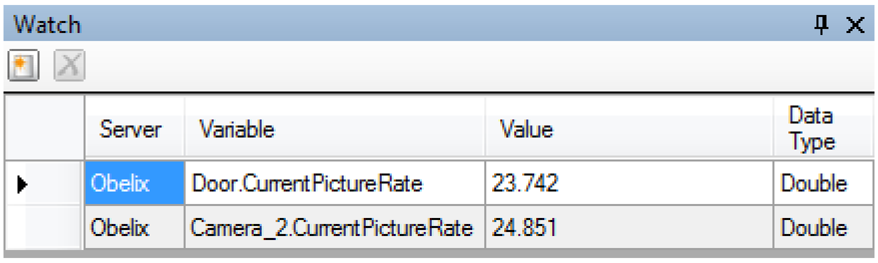


```
alert("Alert");
Debug.Print("The door is open: " + bOpen);
var bool = Camera.Enbael;
```

Server	Variable	Value	Data Type
ETHIRIS	Camera_1.Input1	false	Boolean

Figure 2.297 The Watch panel is docked at the bottom by default.

You can add variables to the list in the watch panel in two different ways; by clicking the *Add variable* button or by dragging variables from the Variable browser.



Server	Variable	Value	Data Type
Obelix	Door.CurrentPictureRate	23.742	Double
Obelix	Camera_2.CurrentPictureRate	24.851	Double

Figure 2.298 Two variables are added to the Watch panel.

The values in the list are updated once every second. By double-clicking on a variable's value cell in the watch panel, its value can be changed directly. The list can be sorted on *Variable* and *Value* in ascending and descending order. Just click the desired column header to change the sorting direction.

Watch panel toolbar

Figure 2.299 Toolbar in the Watch panel.

Add variable to Watch panel

Use this button to add a variable to the list. A dialog containing all available variables is displayed in which you can select a variable.

Delete selected Variable(s)

Use this button to delete selected variables from the list. You can select several variables by dragging the mouse or using the *Ctrl*-key and/or the *Shift*-key.

Watch panel variable list

The variable list is comprised of several columns, of which all are read-only.

Server is the name of the Ethiris Server the variable belongs to.

<Icon> show if the variable is readable, writable, or both.

Variable is the name of the variable.

Value is the current value of the variable.

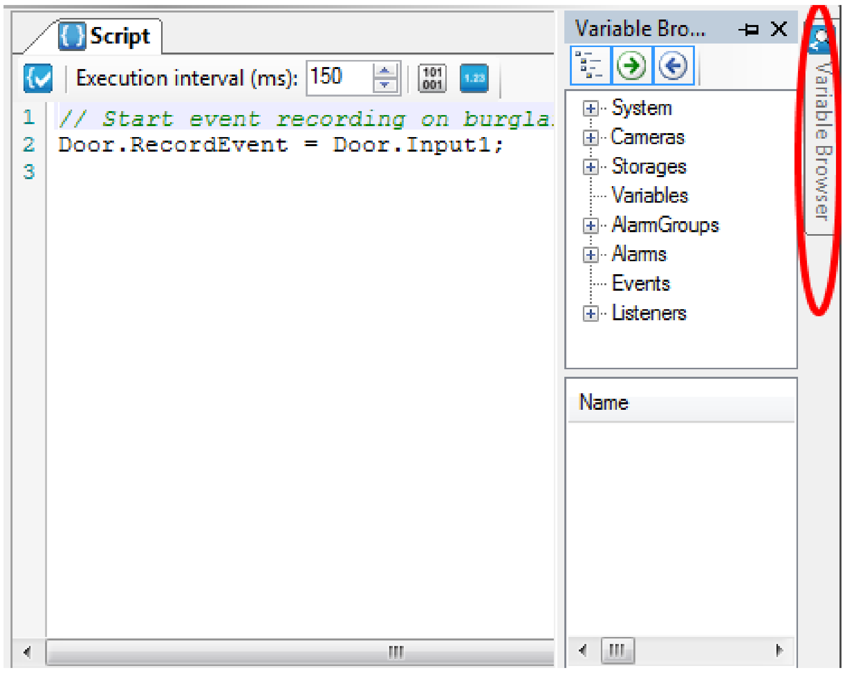
Data Type is the data type of the variable.

No big surprise there.

Variable Browser panel

To the right in the Script panel, there is a *Variable Browser* tool window that is docked. If you move the mouse pointer over the Variable Browse tab, it slides out.

The variable browser contains all available signals in the Ethiris Server data store. The signals are categorized into different types of objects in Ethiris.



```
// Start event recording on burgla
Door.RecordEvent = Door.Input1;
```

Figure 2.300 The Variable Browser when out.

Click in the window to make it stay out alternatively *pin* it by clicking the pin icon.

There are two panes in this window; the upper pane contains all available objects and the lower pane contains the corresponding signals to the currently selected object in the upper pane.

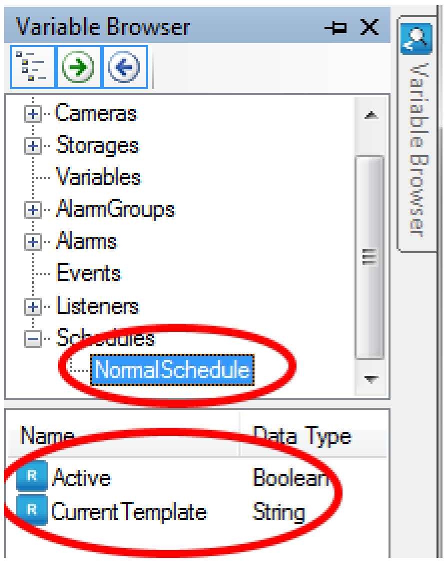


Figure 2.301 A schedule is selected in the Variable browser.

In the example above, a schedule is selected in the upper pane, and the corresponding variables are displayed in the lower pane.

There are two alternatives for selecting variables in the variable browser; *double-click* to copy the signal to the current insertion point (where the cursor is) in the script editor or *drag-and-drop* the variable into the desired location in the script editor.

When a variable is selected into the script, the variable gets fully qualified, i.e., the complete name of the variable is entered into the script. In this case, the *Active* variable of the schedule is called *NormalSchedule.Active*.

Variable browser toolbar

Figure 2.302 The toolbar in the Variable browser panel.

Show variable tree — This is a toggle button. When active, there is a thin blue frame around the button. If this button is not active, all variables are displayed in a long list in alphabetical order. The default is active, and the variables are categorized under the various objects in Ethiris Server’s data store.

Show readable variables — This is a toggle button. When active, there is a thin blue frame around the button. If this button is not active, no readable variables are displayed.

Show writeable variables — This is a toggle button. When active, there is a thin blue frame around the button. If this button is not active, no writeable variables are displayed.

Read more about the script in chapter *Script* on *page 3:1*.

2.4.43 Communication node

The node *Communication* is just a collection node for *Listeners*, *OPC-Servers,* and *Remote Clients* in the current configuration. There is neither a popup menu nor a panel connected to this node.

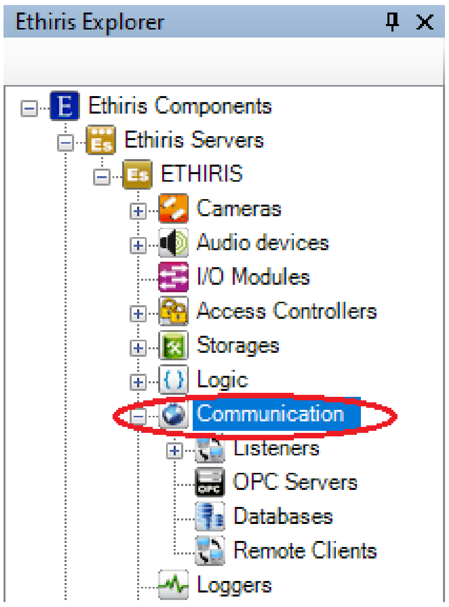


Figure 2.303 The node Communication in Ethiris Explorer treeview.

2.4.44 Listeners node

Under the node Logic in the treeview, there is a *Listeners* node. This is a collection node for all the so-called listeners that are added to the server configuration.

Listeners are first of all used for listening to messages from external equipment such as, e.g. cameras and video encoders. But, somewhat contradictable, listeners can also be used for sending information to external equipment, and in that way, you can achieve two-way communication. The main purpose, though, is to receive information, hence the name *Listeners*.

There are four main types of listeners:

- TCP inbound
- TCP outbound
- HTTP inbound
- HTTP outbound

Which type to use depends on in which way the external equipment can communicate.

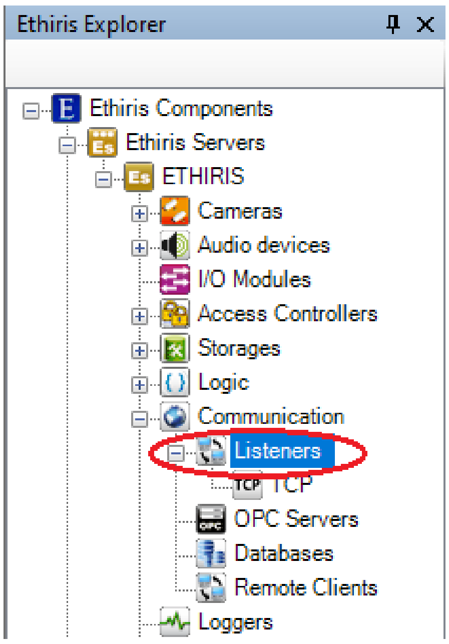


Figure 2.304 The node Listeners in Ethiris Explorer treeview.

Listeners general

Listeners represent a relatively complex part of Ethiris. The upside is that they can be utilized in many situations for communicating with external equipment.

Which protocol to use is determined by *Type*, where you can choose between *TCP* and *HTTP*.

Via *Type,* you also select which one of Ethiris Server or the external equipment is responsible for starting the communication.

Inbound designates the external equipment that is supposed to take the initiative of the communication. In this case, Ethiris Server listens to the specified port and waits for the external equipment to start sending messages. Note that the specified port has to be open in a possible firewall.

Outbound designates that it is Ethiris Server that is supposed to take the initiative of communication. Then Ethiris Server will actively try to connect to the external equipment using the specified IP address and port.

When a connection is established, and the communication commences Ethiris Server receives data from the external equipment (regardless of whether the type is inbound or outbound). Now the whole idea is to activate *triggers* by *matching* corresponding texts in the data received by Ethiris Server.

A *trigger* consists of three parts; *Name*, *Function,* and *Match (Text to match)*.

Each trigger is represented in the Ethiris Server data store as a variable of type *Boolean*. A Boolean variable can hold two different values; *true* or *false*. In this context, this means that the trigger is either triggered or not triggered.

Each listener can have a list of several triggers.

You could say that the whole process consists of two main parts. The first part is about the listeners receiving data, and the second part is about the script engine activating the triggers.

The reception of data looks slightly different depending on whether you have specified a *SOT (Start Of Text)* and/or *EOT(End Of Text)* or not.

Let’s start by assuming that neither *SOT* nor *EOT* has been specified. As soon as the listener has received data it will check its triggers’ *Match* string for matching strings in the received data. If no match is found, the listener keeps receiving data that is kept in the listener’s internal buffer.

Each time new data is received by the listener, it checks its triggers in the order the triggers are defined in the list. When a match is found, all data received so far is copied to a variable named *TriggerString*. The variable is automatically created for each listener and is available via script. The trigger will be put in a queue that the script engine will process. Several triggers can, in this way, match at the same time for a listener. In that case, they will be put in the queue in the order they are defined in the list. The script engine will process them one at a time. If any of the triggers match, the internal buffer of data will be cleared, and the whole process will start all over again.

When there are triggers in the queue, the script engine will process them one at a time. The trigger first in line will be active during one round of script execution. That means that the corresponding variable is *true* during one execution interval. After that, the variable is set to *false* again, and the trigger is removed from the queue. The corresponding *TriggerString* will also be cleared. Then the script engine takes the next trigger in line from the queue, sets its value to *true* during one round of execution, and so on.

Now, let’s assume we have entered a text for *SOT*. Now the listener will discard all data received until the text specified for *SOT* appears in the received data. Then the listener starts to collect received data in its internal buffer and check its triggers in the same way as described above. When we find a match the corresponding trigger is put in the queue for the script engine, the *TriggerString* variable will be updated with the data that has been received so far starting with the *SOT,* and after that, the internal buffer will be cleared, and the listener will start looking for a new *SOT* in the incoming data.

If we specify an *EOT,* the check of triggers will not be performed until the text specified as *EOT* appears in the received data. If there is no match, all data, including the *EOT,* received so far will be discarded, and the process starts all over again. If there is a match, the matching trigger will be put in the queue, the *TriggerString* variable will be updated, and the internal buffer up to and including *EOT* will be cleared. Note that there may be several *EOT* in the received data. In that case, they will be processed one at a time. First, the part of the data up to and including the firs *EOT* will be processed. Triggers will be checked and possibly put in the queue. Then that part of the data will be discarded, and the next part of the data up to the second *EOT* will be processed, and so on.

Finally, if both *SOT* and *EOT* are specified, the listener will start buffering data after the *SOT* is received, and then the listener collects data until an *EOT* appears. Not before that, the list of triggers will be checked. In all other respects, it works in the same way as described above.

Listeners popup menu

Right-click on this node will open a menu.

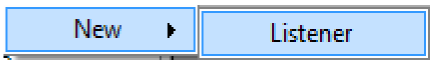


Figure 2.305 Popup menu for the node Listeners.

New->Listener adds a new listener to the server configuration. It will immediately be visible in the treeview as a new listener node. Should the panel *Listeners* be open, it will be visible there as well.

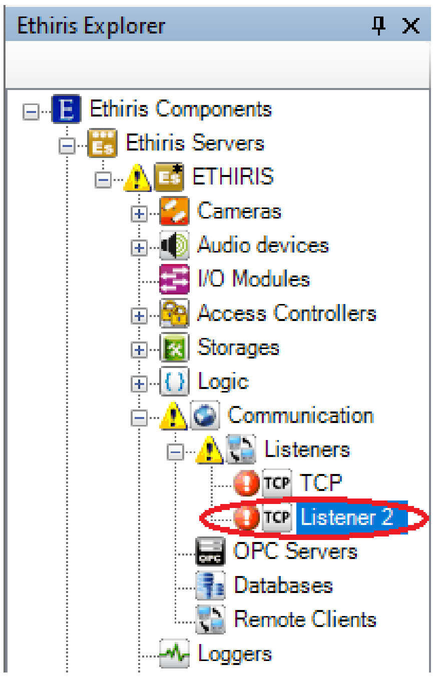
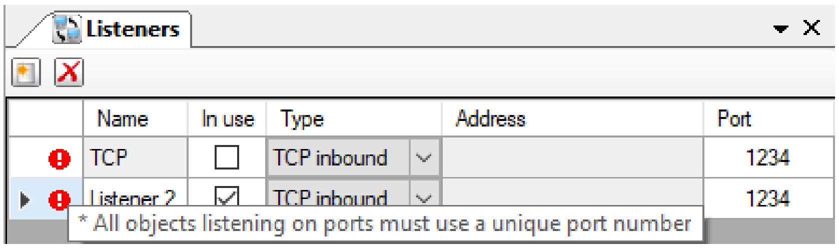


Figure 2.306 New listener added.

When you have just added a new listener, an error is indicated in the treeview. The reason for this is that by default, *Type TCP inbound* and *Port 1234* are used. These are the same as for the automatically created TCP listener. Hence an error is indicated. This problem will go away when we are finished configuring the new listener.

Listeners panel

Double-click on the node *Listeners* in the treeview will open the corresponding panel.



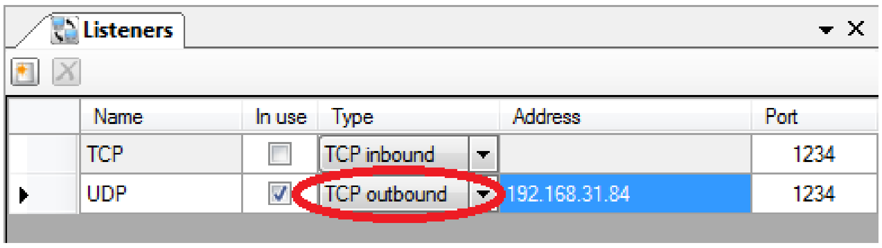
Listeners

	Name	In use	Type	Address	Port
●	TCP	☐	TCP inbound		1234
●	Listener 2	☑	TCP inbound		1234

* All objects listening on ports must use a unique port number

Figure 2.307 The Listeners panel.

If you hover the mouse pointer over the red error indicator, information about what’s wrong is displayed. In the example above, we can see that the problem is that two different listeners are listening to the same port. If we change the type to *TCP outbound* for the new listener, the problem will go away. The purpose of the new listener is to communicate with a UDP camera, which has support for TCP communication. We will come back to the UDP camera later on in the manual.



	Name	In use	Type	Address	Port
	TCP	☐	TCP inbound		1234
▶	UDP	☑	TCP outbound	192.168.31.84	1234

Figure 2.308 The Listeners panel with type changed for the new listener.

This panel consists of a list of all listeners currently defined in the server configuration.

At the top of the panel, there is a toolbar.

Listeners panel toolbar

Figure 2.309 The toolbar in the Listeners panel.

New Listener	Use this button to create a new listener. Clicking this button is the same as selecting the menu item *New->Listener* in the popup menu described above. A new listener will immediately be added to the configuration.
Delete selected Listener(s)	Use this button to delete selected listeners from the configuration. You can select more than one listener by using the *Ctrl*-key and/or the *Shift*-key.

Listeners panel listener list

The listener list consists of several columns.

Name is the desired name of the listener. This name has to be unique in the configuration. If you enter an illegal name there will be an icon to the left of the listener in the list that indicates the error.

In use determines if the listener is used or not. This property is especially useful if you don't want to use the automatically created TCP listener that can't be deleted. If the listener is not used, no connection to the external equipment will be set up.

Type determines what type of listener this will be. You can choose between *TCP* and *HTTP* and also between *inbound* and *outbound*. TCP/HTTP determines what protocol to use while inbound/outbound determines who takes the initiative for communication. When the type is inbound, it's the external equipment that is supposed to take the initiative, and when the type is outbound, it's Ethiris Server's responsibility to initiate the communication.

Address is only used when the type is *outbound*. Then it determines the address of the external equipment.

Port determines the TCP port to use when communication is initiated. When the type is inbound, this is the port Ethiris Server will listen to for incoming messages. When the type is *outbound,* this is the port Ethiris Server will use when it establishes contact with the external equipment.

2.4.45 TCP node

Under the node Listeners in the treeview, there is a *TCP* node. This is a listener that is automatically created by the system, and it cannot be deleted.

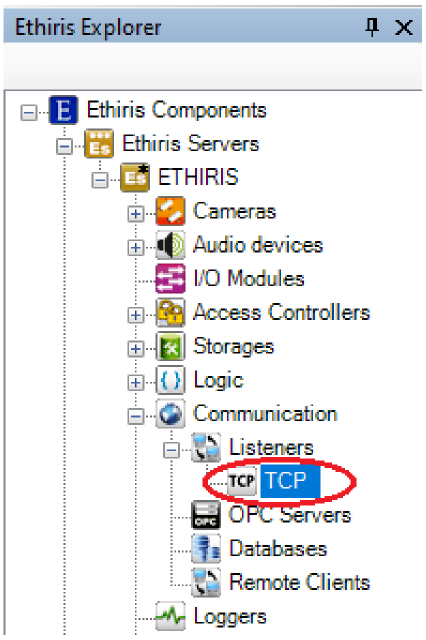
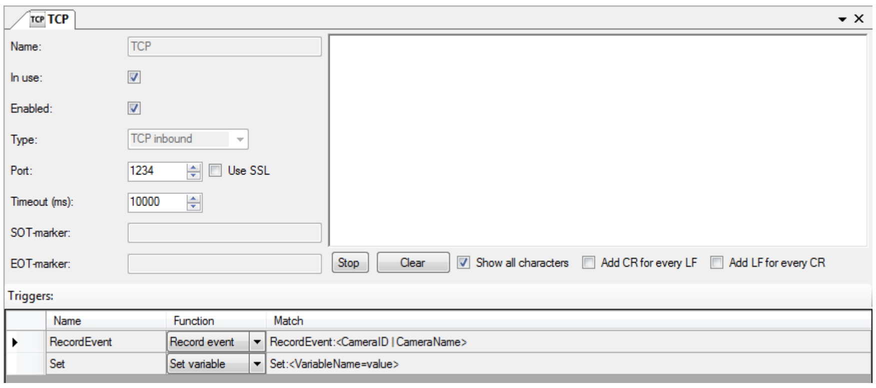


Figure 2.310 The node TCP in Ethiris Explorer treeview.

This listener is pre-configured in a way that corresponds to the way TCP communication was done in previous versions of Ethiris. When enabled, Ethiris Server listens to incoming TCP requests. The purpose is to receive notifications from cameras or video encoders via TCP when an alarm condition is detected by the camera/video encoder. In this way, it is not necessary to send video from the camera to Ethiris Server unless an alarm occurs. The load of the network is thus decreased. You can also utilize the functions in the camera/video encoder, such as motion detection, and take the load off the server by letting the camera do the work.

TCP panel

Double-click on the TCP node in the treeview opens the corresponding panel.


TCP

Name:	TCP
In use:	✓
Enabled:	✓
Type:	TCP inbound
Port:	1234 ☐ Use SSL
Timeout (ms):	10000
SOT-marker:	
EOT-marker:	

Stop Clear ✓ Show all characters ☐ Add CR for every LF ☐ Add LF for every CR

Triggers:

Name	Function	Match
RecordEvent	Record event	RecordEvent:<CameraID | CameraName>
Set	Set variable	Set:<VariableName=value>

Figure 2.311 The TCP panel.

The following fields exist in the panel:

Name is, in this case, not possible to change but is always *TCP*. The name is used to identify the listener in the script.

In use determines if the listener will be used. If not Ethiris Server will not set up the listener and respond to incoming messages.

Enabled determines if the TCP listener will be enabled from the beginning when Ethiris Server starts. This property is available as a writeable variable in the script named *Enable*. This variable can be used in a script to control when the listener is enabled.

Type determines the type of listener. In this case, the type cannot be changed but is always *TCP inbound*. That means the protocol is *TCP,* and the external equipment is supposed to initiate the communication with Ethiris Server, hence *inbound*.

Port determines the TCP-port that Ethiris Server listens to for inbound TCP messages. By default, this is *1234,* but it can be changed if necessary. Remember that the port number has to be unique such as only one unit at the time for the same IP address can listen to the same port. E.g., you cannot have two different listeners in the same Ethiris Server listening at the same port.

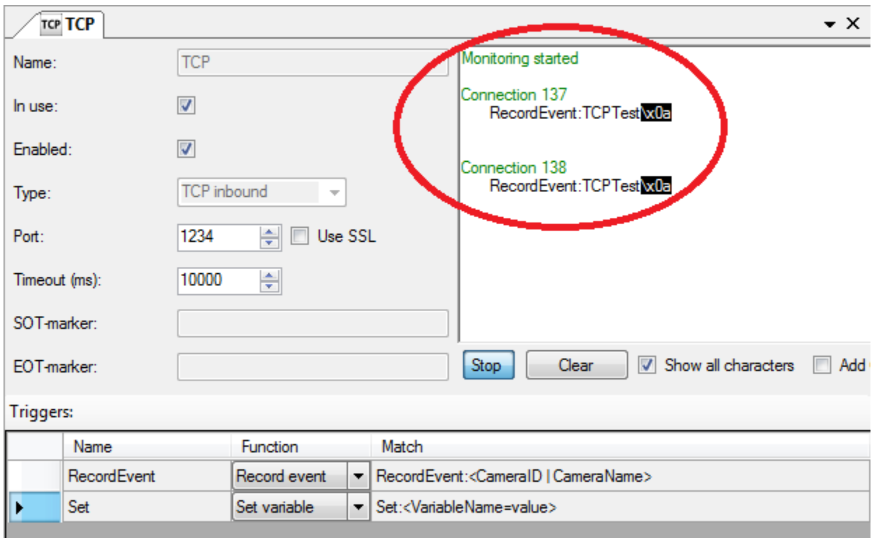
Use SSL specifies if the connection should use encryption. To use this, the other side needs to support SSL as well.

Timeout (ms) determines the maximum time in milliseconds within which the external equipment has to receive the message in case of Ethiris Server sends a response.

SOT-marker stands for *Start Of Text* and can be used to set a limit to which messages that the listener will trig on. No trigging will occur until the text defined in *SOT* has been received by the listener. For this automatically created TCP listener, no SOT can be specified.

EOT-marker stands for *End Of Text* and can be used to set a limit to which messages that the listener will trig on. No trigging will occur after the text defined in *EOT* has been received by the listener. For this automatically created TCP listener, no EOT can be specified.

Start is used to start displaying the text that the listener receives. The text is displayed in the monitor window to the right. In the example below, an Axis camera has been configured to send TCP messages on motion, containing the text *RecordEvent:TCPTest*, to the Ethiris Server. See the configuration example below where an Axis camera is configured for using the TCP listener.


TCP

Name:	TCP
In use:	✓
Enabled:	✓
Type:	TCP inbound
Port:	1234 ☐ Use SSL
Timeout (ms):	10000
SOT-marker:	
EOT-marker:	

```
Monitoring started

Connection 137
    RecordEvent:TCPTest\x0a

Connection 138
    RecordEvent:TCPTest\x0a
```

Triggers:

Name	Function	Match
RecordEvent	Record event	RecordEvent:<CameraID | CameraName>
Set	Set variable	Set:<VariableName=value>

Figure 2.312 The TCP panel with the monitoring started.

Stop is used to stop displaying the received text.

Clear is used for clearing the monitor window of all text.

Show all characters should be checked if you want to display hidden characters such as CR (Carriage Return) and LF (Line Feed) as plain text. In the example above, LF, which has ASCII code 10 (*0a* in hexadecimal), is shown in plain text *\x0a* with a black background color.

Add CR for each LF is used for simply adding a *Carriage Return* after a *Line Feed*.

Add LF for each CR is simply used for adding a *Line Feed* after a *Carriage Return*.

Triggers

The purpose of triggers is to specify texts that the listener will catch in the flow of data received by the listener. When the listener discovers the text specified under the column *Match,* the function selected under the column *Function* is activated.

Normally you can add and remove *triggers* in this list. But, in this case, with the automatically created TCP listener, there are two pre-defined triggers, which cannot be changed.

RecordEvent is used for starting an event recording for a camera. The external equipment the listener listens to, most often a camera, is supposed to send the text *RecordEvent:* directly followed by either the camera ID or the camera name. E.g., the text *RecordEvent:Door* should be sent for starting event recording on the camera *Door*.

Set is used for setting the value of any writeable variable in the Ethiris Server data store. This is a very powerful feature, which potentially can activate any function in Ethiris by using a suitable script. E.g., the text *Set:sendMail=true* could be used via script to send one or several email messages when something is triggered in the camera.

This function requires some configuration in the camera/video encoder to work, so let’s take a look at an example of how to do that...

Camera/video encoder configuration example

In this example, we will use an Axis 216FD camera, where we will let the camera detect motion and then send a TCP notification to Ethiris Server. The message sent from the camera will write a value to a variable in Ethiris Server's data store. This is a very powerful feature that can potentially be used to fire up any function in Ethiris. Examples are starting recording, activating PTZ preset positions, popup live views in any Ethiris Client, etc.

The first task is to create a motion detection definition in the camera's configuration. Enter *Setup* in the camera, select *Event Config->Motion Detection,* and then click the *New* button to the right. Enter an appropriate name of the detection window and position and resize the window as desired.

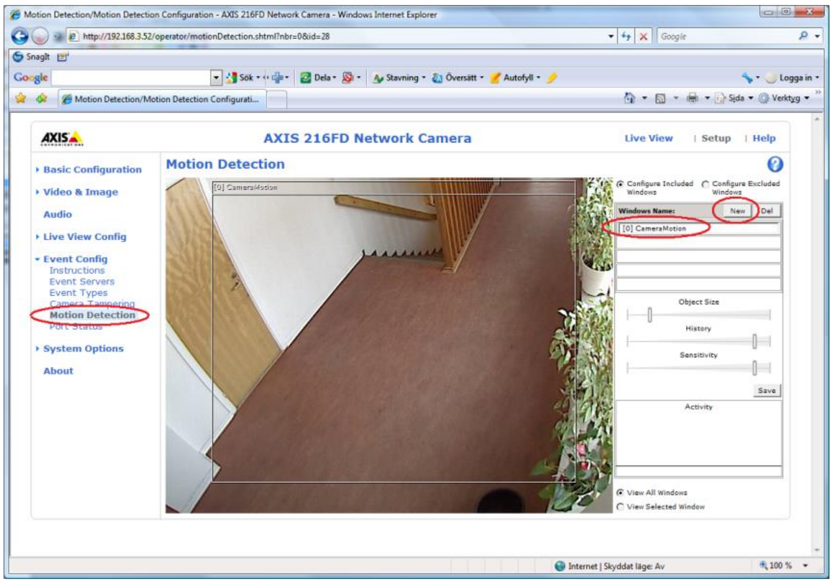


Figure 2.313 Add Motion detection for a camera.

The next task is to create an *Event Server* in the camera configuration. Select *Event Config->Event Servers* and then click the *Add TCP…* button.

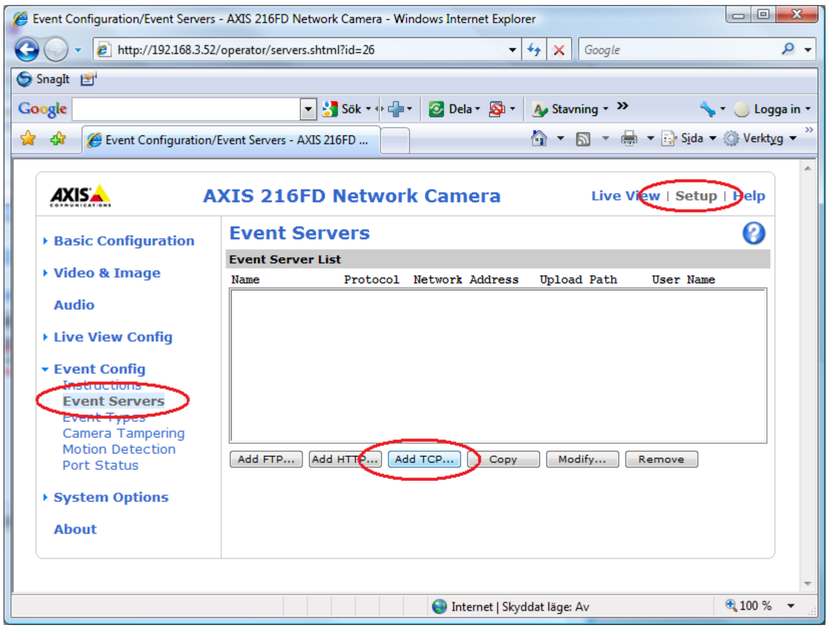
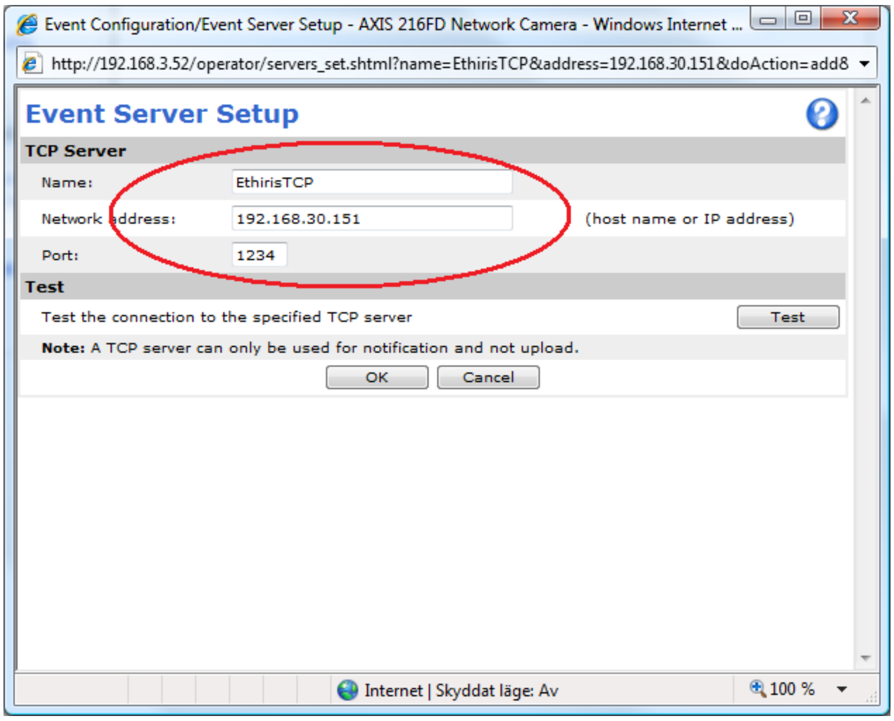


Figure 2.314 Add TCP Event Server for a camera.

In the *Event Server Setup* dialog, enter an appropriate name, the IP address of the computer where Ethiris Server runs, and the port number which Ethiris Server listens to for TCP notifications (1234 as default). Click the *Test* button to test the connection.



Event Configuration/Event Server Setup - AXIS 216FD Network Camera - Windows Internet ...

http://192.168.3.52/operator/servers_set.shtml?name=EthirisTCP&address=192.168.30.151&doAction=add8

Event Server Setup

TCP Server

Name: EthirisTCP

Network address: 192.168.30.151 (host name or IP address)

Port: 1234

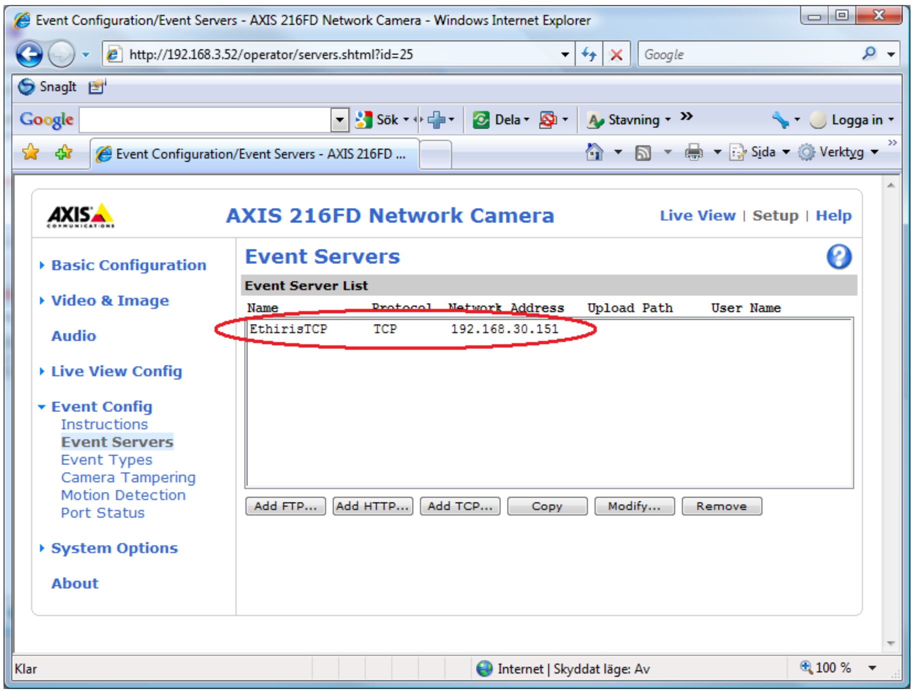
Test

Test the connection to the specified TCP server

Note: A TCP server can only be used for notification and not upload.

Figure 2.315 TCP Event Server dialog.

When you are done, click the *OK* button. There should be a new item in the list of Event Servers.



Event Configuration/Event Servers - AXIS 216FD Network Camera - Windows Internet Explorer

http://192.168.3.52/operator/servers.shtml?id=25

AXIS 216FD Network Camera

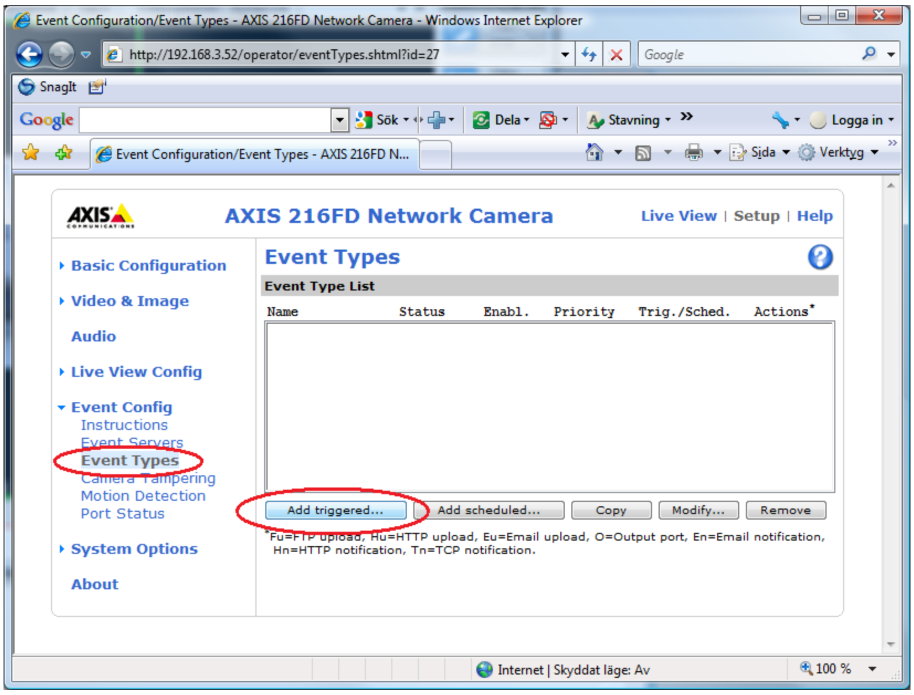
Event Servers

Event Server List

```
Name          Protocol  Network Address  Upload Path    User Name
EthirisTCP    TCP       192.168.30.151
```

Figure 2.316 A new TCP Event Server is created.

The next task is to add two new *Event Types*, one for the start of motion and one for the stop of motion. Select *Event Config->Event Types* and then click the *Add triggered...* button.



*Fu=FTP upload, Hu=HTTP upload, Eu=Email upload, O=Output port, En=Email notification, Hn=HTTP notification, Tn=TCP notification.

Figure 2.317 Add TCP Event Type for a camera.

In the *Event Type Setup* dialog, enter an appropriate name, select the motion detection in the *Triggered by…* section, check the *Send TCP notification to* checkbox and select the appropriate *Event Server* in the *When Triggered…* section.

In the *Message* field, enter the text *Set:* followed by the name of the variable, an equal sign (”=”), and the desired value of the variable. In our example, we want to set the *RecordEvent* variable to *true* for the camera called *Aisle*. The whole message will then be:

Set:Aisle.RecordEvent=true

You can set any variable to (almost) any value you like.

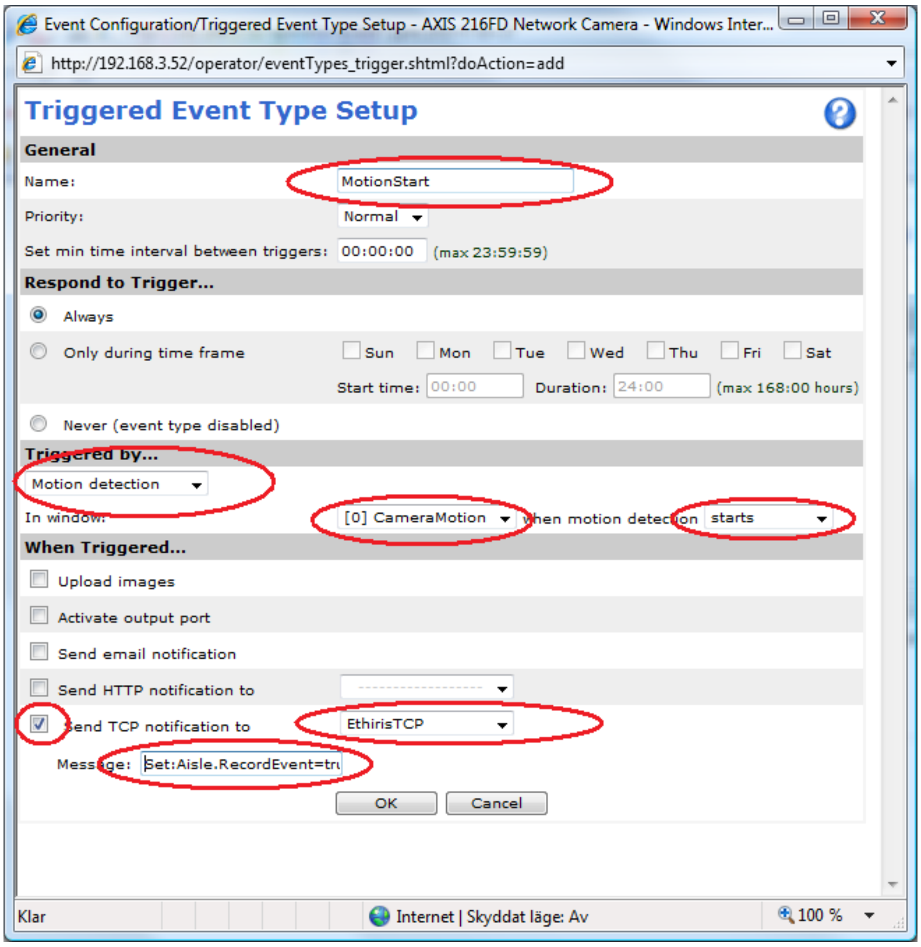
Ethiris Server listens for two different commands:

Set:<Variable name>=<Value>

RecordEvent:<Camera name | Camera ID>

The command *Set* is used for writing a value to a variable in Ethiris Server’s data store.

The command *RecordEvent* is used for starting event recording on a camera using either the camera’s name or ID. This has the same effect as clicking the *Manual Recording button* in Ethiris Client for a certain camera.



Event Configuration/Triggered Event Type Setup - AXIS 216FD Network Camera - Windows Inter...

http://192.168.3.52/operator/eventTypes_trigger.shtml?doAction=add

Triggered Event Type Setup

General

Name:	MotionStart
Priority:	Normal
Set min time interval between triggers:	00:00:00 (max 23:59:59)

Respond to Trigger...

- [✓ Always]
- Only during time frame: Sun, Mon, Tue, Wed, Thu, Fri, Sat; Start time: 00:00; Duration: 24:00 (max 168:00 hours)
- Never (event type disabled)

Triggered by...

Motion detection

In window: [0] CameraMotion when motion detection starts

When Triggered...

- Upload images
- Activate output port
- Send email notification
- Send HTTP notification to
- [✓ Send TCP notification to] EthirisTCP

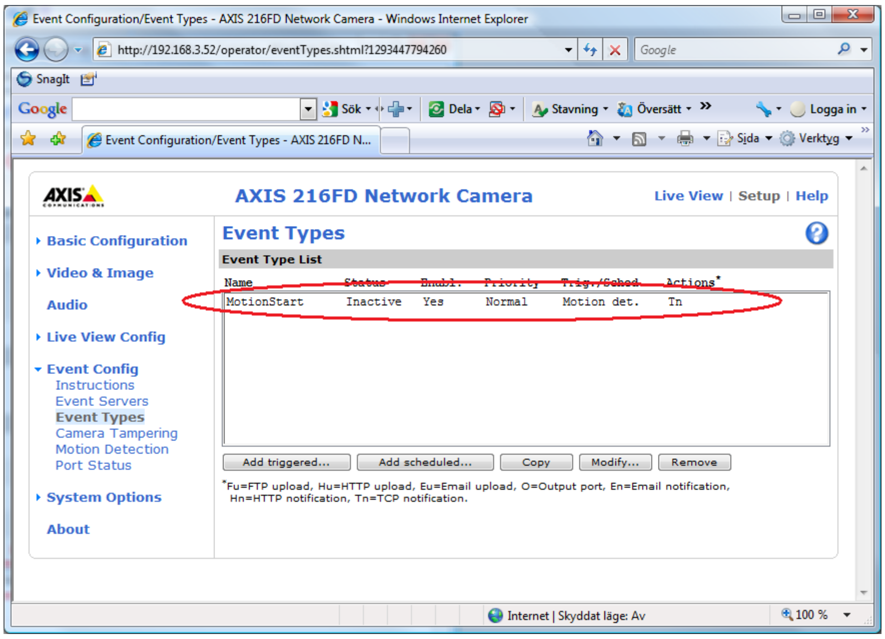
Message: Set:Aisle.RecordEvent=tru

OK Cancel

Klar | Internet | Skyddat läge: Av | 100 %

Figure 2.318 Event Type dialog.

When you are done, click the *OK* button. There should be a new item in the list of Event Types.



Event Configuration/Event Types - AXIS 216FD Network Camera - Windows Internet Explorer

http://192.168.3.52/operator/eventTypes.shtml?1293447794260

AXIS 216FD Network Camera Live View | Setup | Help

- Basic Configuration
- Video & Image
- Audio
- Live View Config
- Event Config
  - Instructions
  - Event Servers
  - Event Types
  - Camera Tampering
  - Motion Detection
  - Port Status
- System Options
- About

Event Types

Event Type List

Name	Status	Enabl.	Priority	Trig./Sched.	Actions*
MotionStart	Inactive	Yes	Normal	Motion det.	Tn

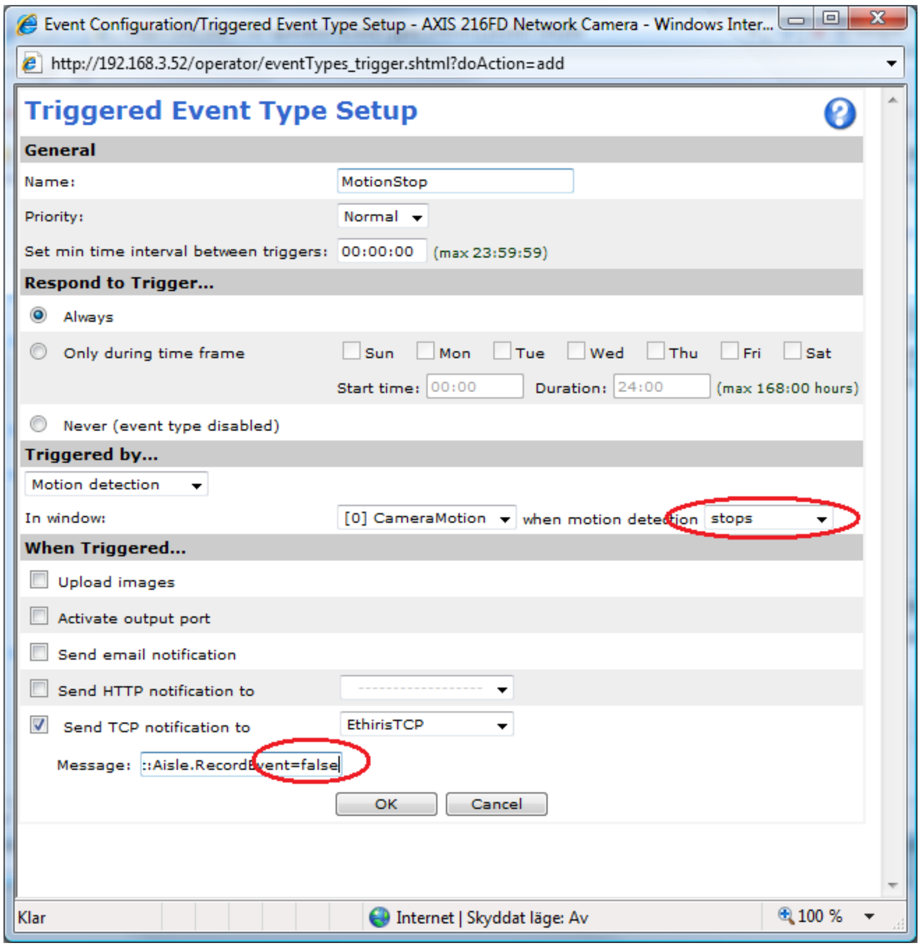
Add triggered... Add scheduled... Copy Modify... Remove

*Fu=FTP upload, Hu=HTTP upload, Eu=Email upload, O=Output port, En=Email notification, Hn=HTTP notification, Tn=TCP notification.

Internet | Skyddat läge: Av | 100 %

Figure 2.319 A new Event Type is created.

Now, add another *Event Type* called *Motion Stop,* where you use *Triggered by…* when motion detection *stops* instead of starts. You also set the value of Aisle.RecordEvent to *false*.


Event Configuration/Triggered Event Type Setup - AXIS 216FD Network Camera - Windows Inter...
http://192.168.3.52/operator/eventTypes_trigger.shtml?doAction=add

Triggered Event Type Setup

General

Name:	MotionStop
Priority:	Normal
Set min time interval between triggers:	00:00:00 (max 23:59:59)

Respond to Trigger...

- [● Always]
- Only during time frame: Sun, Mon, Tue, Wed, Thu, Fri, Sat. Start time: 00:00 Duration: 24:00 (max 168:00 hours)
- Never (event type disabled)

Triggered by...

Motion detection
In window: [0] CameraMotion when motion detection stops

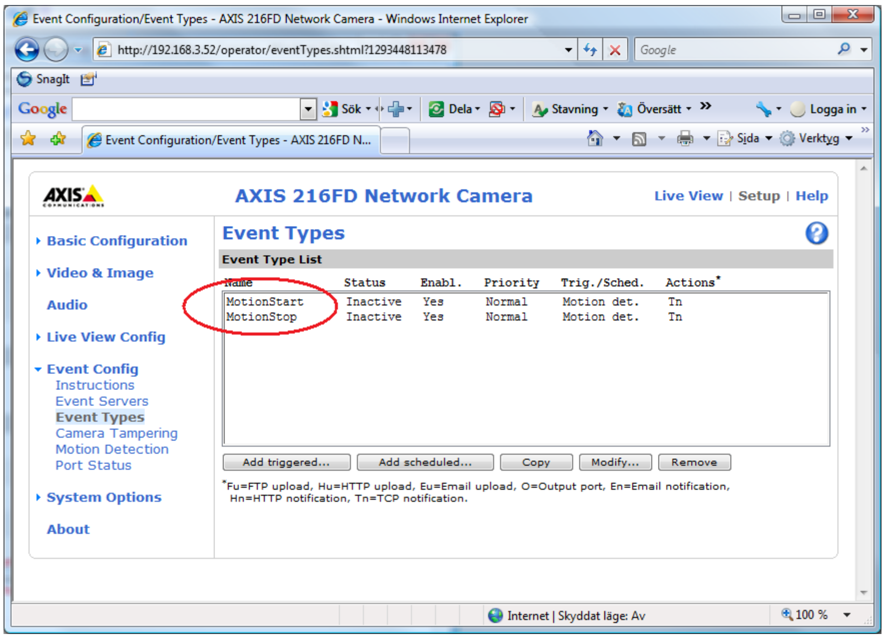
When Triggered...

- Upload images
- Activate output port
- Send email notification
- Send HTTP notification to
- [✓ Send TCP notification to] EthirisTCP

Message: ::Aisle.RecordEvent=false

Figure 2.320 Event Type dialog for motion stop.

Now there are two Event Types added, and we are done. Try it out by making sure there are some motion in front of the camera and verify that Ethiris Server starts event recording accordingly.



```
Name          Status    Enabl.  Priority  Trig./Sched.  Actions*
MotionStart   Inactive  Yes     Normal    Motion det.   Tn
MotionStop    Inactive  Yes     Normal    Motion det.   Tn
```

*Fu=FTP upload, Hu=HTTP upload, Eu=Email upload, O=Output port, En=Email notification, Hn=HTTP notification, Tn=TCP notification.

Figure 2.321 Two Event Types added.

2.4.46 Listener node

Under the node Listeners in the treeview, you can add new *Listeners*. Previously in the manual, we added a listener called *UDP*, which is supposed to communicate with a UDP camera.

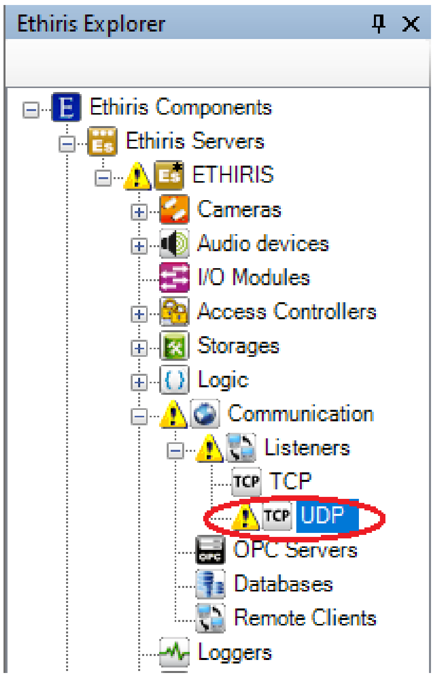


Figure 2.322 The Listener node in the Ethiris Explorer treeview.

The observant observes a warning icon to the left of the new listener in the treeview. It warns about the listener configuration not being complete. We will take care of that shortly.

Listener popup menu

Right-clicking this node opens a menu.

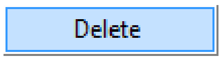


Figure 2.323 Popup menu for the Listener node.

Delete deletes the listener from the server configuration. If you have opened the *Listeners* panel, the listener will be deleted there as well.

Listener panel

Double-clicking the *Listener* node in the treeview opens the corresponding panel.

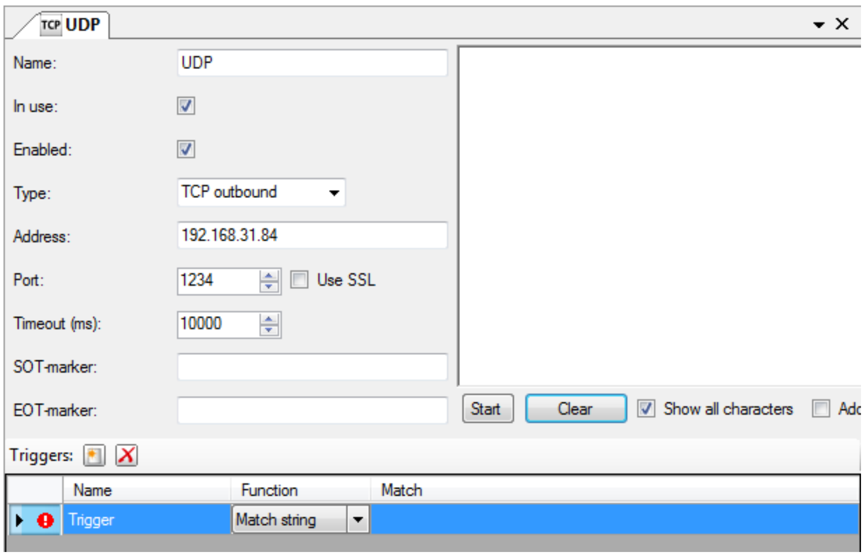


Figure 2.324 Listener panel.

The following fields exist in the panel (different fields are shown depending on the type of listener. However, all fields are described here):

Name The name is used to identify the listener in the script.

In use specifies if the listener is in operation. If not, Ethiris Server will not set up the listener or listen for incoming messages.

Active means if the listener should be active when Ethiris Server starts. This property is available as a writable variable in the script named *Enable*. Via this variable, you can control if the listener should be active or not from the script.

Type is the type of listener. There are four types of listeners: *TCP inbound, TCP outbound, HTTP inbound,* and *HTTP outbound.* Depending on the type of listener, certain fields in this panel will not be shown.

Port specifies the TCP-port that Ethiris Server will listen on for incoming TCP-messages. The default setting is *1234,* but this can be changed. Note that the port number must be unique on the server computer as only one device can listen to each port. So you can not have two listeners listening on the same port.

Use SSL specifies if the connection should use encryption. To use this, the other part needs to support SSL as well.

Timeout (ms) is the maximum time Ethiris Server will allow before the external device receives a message in case Ethiris is sending information back.

Auto reconnect specifies that Ethiris should automatically reconnect if the connection should be dropped. Only available for *HTTP outbound.*

Request is the string Ethiris will send to the external device after a connection has been established. Only available for *TCP outbound* or *HTTP outbound*.

Send auto response instructs Ethiris to automatically handle login if the remote device requires this. Only available for *HTTP inbound* or *HTTP outbound.*

Authentication specifies if Ethiris should require that the external equipment authenticates with the selected method. The alternatives are *None, Basic,* or *Digest.* Only available for *HTTP inbound.*

User name is the user name Ethiris will authenticate with if *Type* is *HTTP outbound. I*t is also the name Ethiris will require the external equipment to authenticate with if *Authentication* is *Basic* or *Digest* and *Type* is *HTTP inbound.*

Password is the password Ethiris will authenticate with if *Type* is *HTTP outbound,* it is also the password Ethiris will require the external equipment to authenticate with if *Authentication* is *Basic* or *Digest* and *Type* is *HTTP inbound.*

SOT-marker stands for *Start Of Text* and can be used to limit the message part where the listener should trigger since the triggering will take place on the part of the message preceding the *SOT* marker. For this automatically created TCP-listener, you can not specify any SOT.

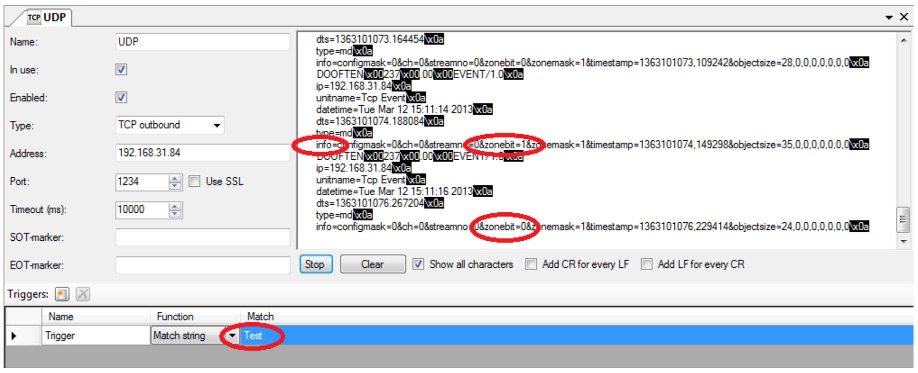
EOT-markering stands for *End Of Text* and can be used to limit the message part where the listener should trigger since no triggering will take place on the part of the message succeeding the *EOT*. For this automatically created TCP-listener, you can not specify any EOT.

Here we can see the reason for the warning in the treeview. At the bottom of the panel, there is a list of *Triggers*. A trigger with the function *Match string* has to have a text defined for the column *Match* to be complete.

Now, it might not be completely clear what string to look for. Then it's a good idea to start the monitoring. Unfortunately, you can't do that until you have saved the configuration, and since it contains errors, it's unwilling to be saved.

The solution is to enter a temporary text, any at all, for *Match*, e.g., *Test*.

Let's do that, save, and then click the *Start* button to display the text the camera sends.

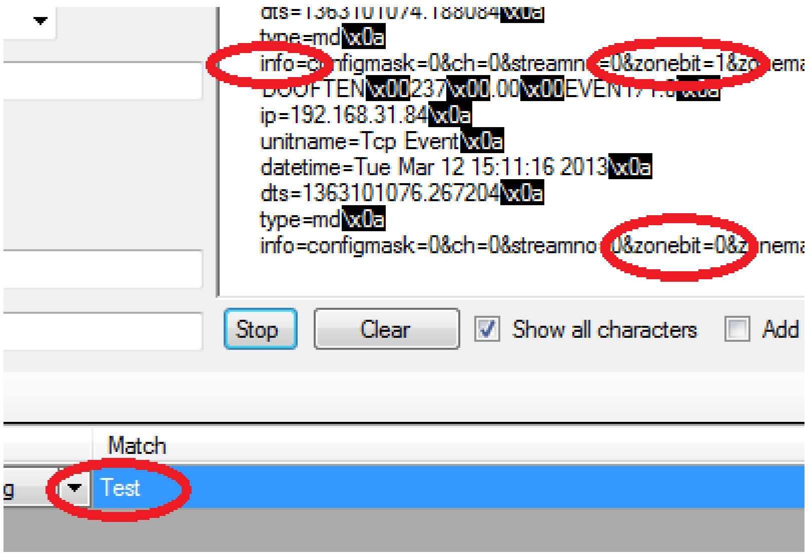


Name:	UDP
In use:	✓
Enabled:	✓
Type:	TCP outbound
Address:	192.168.31.84
Port:	1234 ☐ Use SSL
Timeout (ms):	10000
SOT-marker:	
EOT-marker:	

```
dts=1363101073.164454\x0a
type=md\x0a
info=configmask=0&ch=0&streamno=0&zonebit=0&zonemask=1&timestamp=1363101073,109242&objectsize=28,0,0,0,0,0,0,0\x0a
DOOFTEN\x00237\x00.00\x00EVENT/1.0\x0a
ip=192.168.31.84\x0a
unitname=Tcp Event\x0a
datetime=Tue Mar 12 15:11:14 2013\x0a
dts=1363101074.188084\x0a
type=md\x0a
info=configmask=0&ch=0&streamno=0&zonebit=1&zonemask=1&timestamp=1363101074,149298&objectsize=35,0,0,0,0,0,0,0\x0a
DOOFTEN\x00237\x00.00\x00EVENT/1.0\x0a
ip=192.168.31.84\x0a
unitname=Tcp Event\x0a
datetime=Tue Mar 12 15:11:16 2013\x0a
dts=1363101076.267204\x0a
type=md\x0a
info=configmask=0&ch=0&streamno=0&zonebit=0&zonemask=1&timestamp=1363101076,229414&objectsize=24,0,0,0,0,0,0,0\x0a
```

	Name	Function	Match
▶	Trigger	Match string	Test

Figure 2.325 Listener panel with some received data.



```
dts=1363101074.188084\x0a
type=md\x0a
info=configmask=0&ch=0&streamno=0&zonebit=1&zonema
DOOFTEN\x00237\x00.00\x00EVENT/1.0\x0a
ip=192.168.31.84\x0a
unitname=Tcp Event\x0a
datetime=Tue Mar 12 15:11:16 2013\x0a
dts=1363101076.267204\x0a
type=md\x0a
info=configmask=0&ch=0&streamno=0&zonebit=0&zonema
```

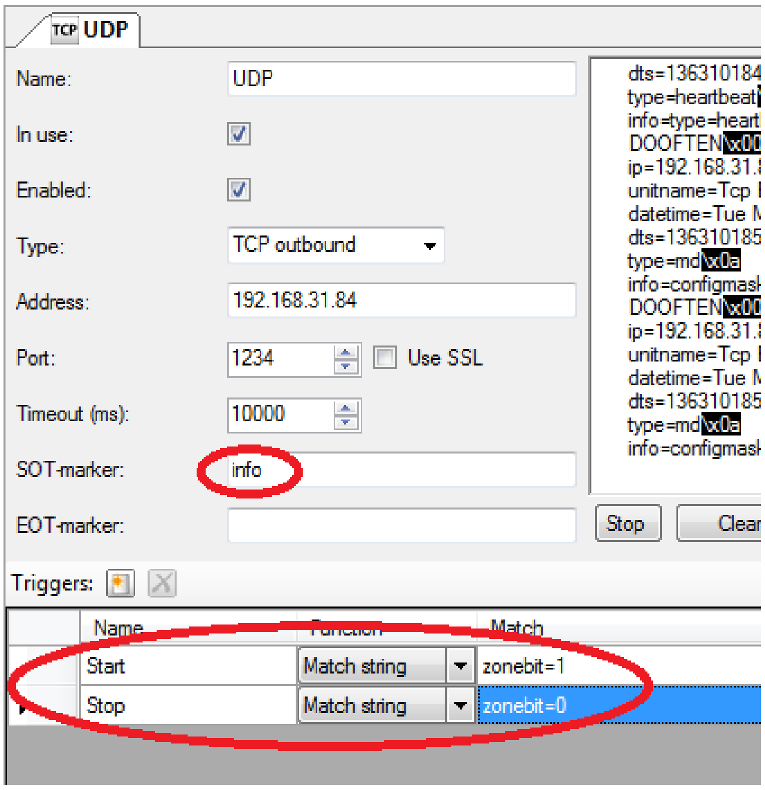
Figure 2.326 An enlarged Listener panel with some received data.

Via the monitor window, it's much easier to find out which texts are suitable for matching.

Depending on how you have configured your camera, the information can look different. In this case, we have set up a motion detection in the camera. When there is enough movement in front of the camera, it sends, among other data, *zonebit=1,* and when it stops moving, it sends *zonebit=0*. Furthermore, it sends the text *info* as the start of each message block.

With this new insight, we can finish the configuration of our listener.

We will create two triggers; one will be called *Start,* and another will be called *Stop*. We will also enter *info* as *SOT* to make the listener slightly more efficient.



TCP UDP

Name: UDP
In use: ✓
Enabled: ✓
Type: TCP outbound
Address: 192.168.31.84
Port: 1234 Use SSL
Timeout (ms): 10000
SOT-marker: info
EOT-marker:

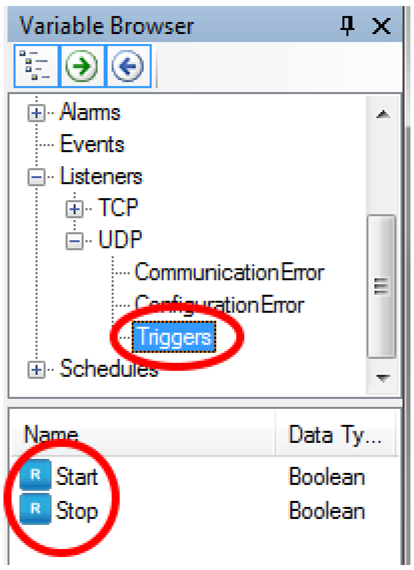
```
dts=136310184
type=heartbeat
info=type=heart
DOOFTEN\x00
ip=192.168.31.
unitname=Tcp
datetime=Tue
dts=136310185
type=md\x0a
info=configmasl
DOOFTEN\x00
ip=192.168.31.
unitname=Tcp
datetime=Tue
dts=136310185
type=md\x0a
info=configmasl
```

Triggers:

Name	Function	Match
Start	Match string	zonebit=1
Stop	Match string	zonebit=0

Figure 2.327 SOT defined and two new triggers created.

The two triggers will appear as *Boolean* variables in the variable browser in the script panel.



Name	Data Ty...
Start	Boolean
Stop	Boolean

Figure 2.328 The new trigger variables are available in the variable browser.

With these new variables, we can write a small script for starting a recording when there is movement in front of the camera.

The following script makes this happen:

```
UDPCam.RecordEvent = (UDPCam.RecordEvent || UDP.Start)
&& !UDP.Stop;
```

Since the trigger variables are *true* only during one round of script execution, we need this construct of the script where the *RecordEvent* variable holds itself until we get a stop signal in the form of *UPD.Stop* becoming *true*.

`||` means ”or”.

`!` means ”not”.

If you found listeners interesting, you can read more about them in the *Integration with Ethiris* manual.

2.4.47 OPC Servers node

Also, under the Communication node in the treeview is the *OPC Servers* node. This is a container node for all OPC Servers that are defined in the server.

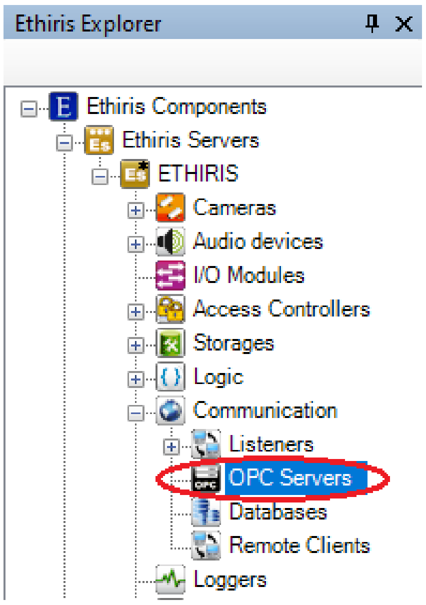


Figure 2.329 The OPC Servers node in Ethiris Explorer treeview.

In many cases, you need to communicate with systems and units outside the system, for example, to obtain information on various states such as a motion detector or the status of a switch. In addition to needing to collect information from the outside world, you often need to influence your own environment, for example, by opening a door or sounding a siren.

OPC (OLE for Process Control) is an industry-standard created to produce a common interface for communication and integration with automation equipment. OPC is based on COM technology for providing a communication link between OPC servers and OPC clients.

Ethiris Server has an integrated OPC client that implements the OPC Data Access standard. This enables integration and communication with all systems with OPC server support.

To link a variable to an object in an OPC server, a link is first created to the OPC server.

OPC Servers popup menu

Right-clicking this node brings up a context menu.

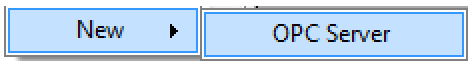


Figure 2.330 The popup menu for the OPC Servers node.

New->OPC Server adds a new OPC server to the server configuration. It is immediately visible in the treeview as a new OPC Server node. Should you have opened the *OPC Servers* panel, the new server would be added there too.

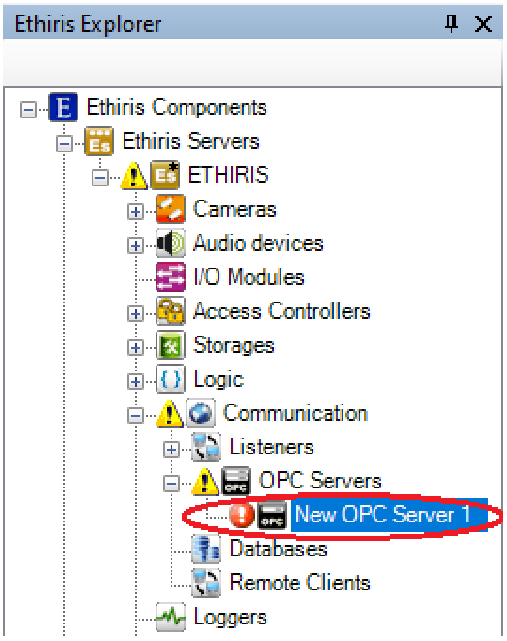
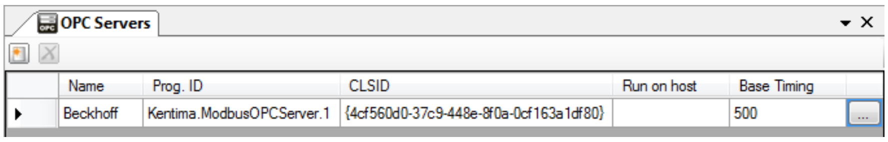


Figure 2.331 New OPC Server added.

Notice the error icon to the left of the new OPC Server node. This is due to the new OPC Server has neither a *Prod ID* nor a *CLSID* yet. There are warning icons further up in the treeview indicating an error somewhere in the configuration.

OPC Servers panel

Double-clicking the *OPC Servers* node in the treeview opens the corresponding panel.



Name	Prog. ID	CLSID	Run on host	Base Timing
Beckhoff	Kentima.ModbusOPCServer.1	{4cf560d0-37c9-448e-8f0a-0cf163a1df80}		500

Figure 2.332 The OPC Servers panel.

This panel consists of a list of all OPC Servers currently part of the server configuration.

At the top of the panel, there is a toolbar.

OPC Servers panel toolbar

Figure 2.333 The toolbar in the OPC Servers panel.

Add a new OPC Server	Use this button to create a new OPC Server. This is the same as *New->OPC Server* in the popup menu described above. A new OPC Server is immediately added to the server configuration.
Delete selected OPC Server(s)	Use this button to delete the selected OPC Server(s) from the configuration. You can select more than one server by using the *Ctrl*-button and/or the *Shift*-button.

OPC Servers panel OPC Server list

The OPC Server list consists of several columns.

Name is the desired name of the OPC Server. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the server in the list indicating the problem.

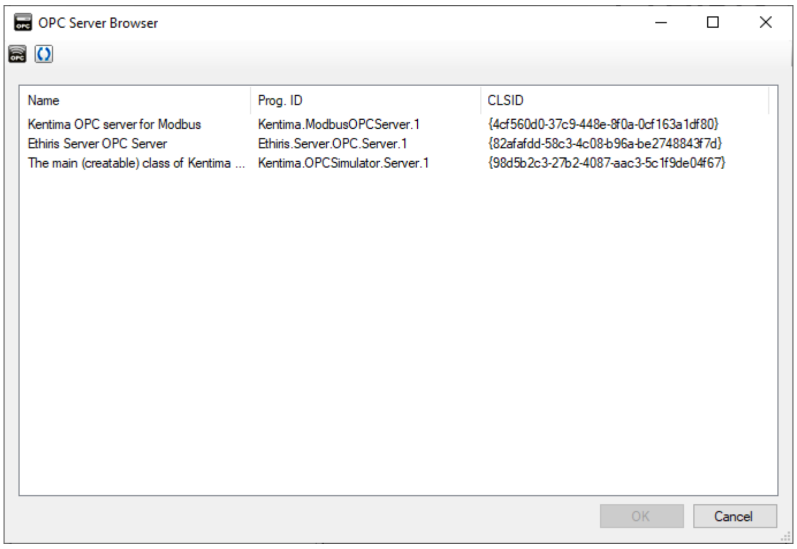
Prog. ID is the programmatic ID of the OPC Server. This is intended for humans since it is easier to understand than the *CLSID*, which is the actual ID of the OPC Server. You can enter this manually, but you can also browse for available OPC Servers by clicking the browse button to the far right in each row.

CLSID is the real, truly unique ID of an OPC Server. You probably want to browse for the OPC Server rather than entering the CLSID manually.

Run on host is used when the OPC Server software is installed on another computer than the computer where Ethiris Server runs. In most cases, the OPC Server software runs on the same computer as Ethiris Server does. This is an advantage since when both Ethiris Server and the OPC Server run on the same computer, the communication is via COM instead of DCOM. DCOM has to be used when communicating between different computers. There are quite a few security settings to handle in order to get DCOM working.

Base timing is the OPC server protocol's basic time in milliseconds, which is to be used as the basis for writing to the OPC server.

Browse button is used for browsing available OPC Servers on both the local computer and if also desired the network.



Name	Prog. ID	CLSID
Kentima OPC server for Modbus	Kentima.ModbusOPCServer.1	{4cf560d0-37c9-448e-8f0a-0cf163a1df80}
Ethiris Server OPC Server	Ethiris.Server.OPC.Server.1	{82afafdd-58c3-4c08-b96a-be2748843f7d}
The main (creatable) class of Kentima ...	Kentima.OPCSimulator.Server.1	{98d5b2c3-27b2-4087-aac3-5c1f9de04f67}

Figure 2.334 The OPC Servers browse dialog.

Display list of remote host

In the example above, the locally installed OPC Servers are listed. To list OPC Servers on other computers, click the *Display list of remote host* button in the upper left of the browse dialog.

When selecting an OPC Server on a remote host, the *Run on host* field will be filled out automatically with the remote host computer name.

Again, this is not the recommended way of OPC communication. Running the OPC Server locally on the same computer as Ethiris Server is preferable.

Refresh

Use this button to refresh the list of discovered OPC-servers.

2.4.48 OPC Server node

Under the OPC Servers node in the treeview, there may be some *OPC Server* nodes.

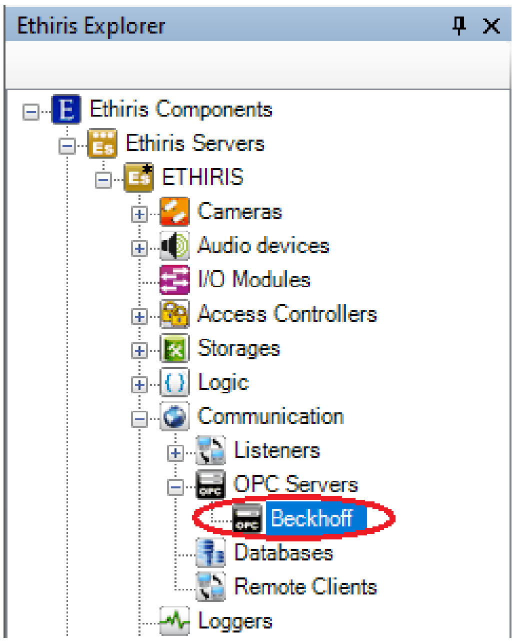


Figure 2.335 An OPC Server node in Ethiris Explorer treeview.

OPC Server popup menu

Right-clicking this node brings up a context menu.

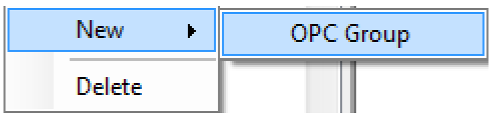


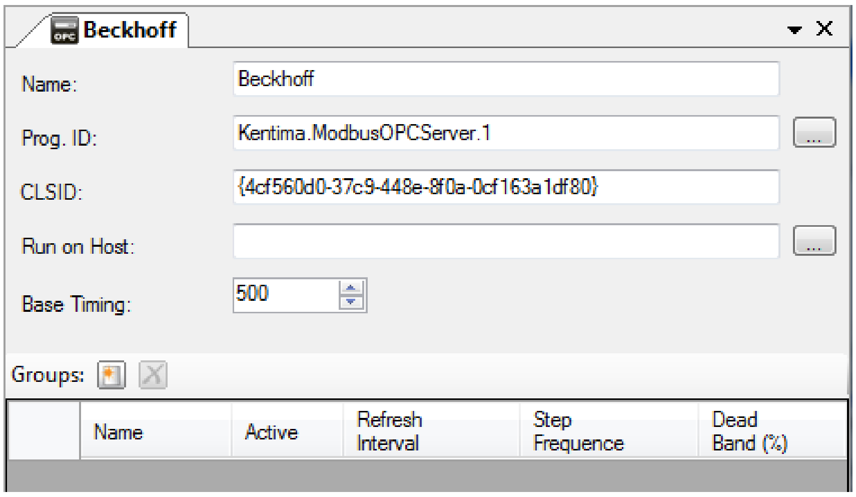
Figure 2.336 The popup menu for an OPC Server node.

New->OPC Group adds a new OPC group to the OPC Server. Groups are created for an OPC server as a container for one or more OPC variables. It is a good idea to add several groups to the same OPC Server to achieve individual settings for, among other things, the update frequency and to emulate the system's structure.

Delete Removes the OPC Server from the server configuration. It is immediately removed from both the treeview and the OPC Server list in the OPC Servers panel.

OPC Server panel

Double-clicking an *OPC Server* node in the treeview opens the corresponding panel.



Beckhoff

Name:	Beckhoff
Prog. ID:	Kentima.ModbusOPCServer.1
CLSID:	{4cf560d0-37c9-448e-8f0a-0cf163a1df80}
Run on Host:	
Base Timing:	500

Groups:

Name	Active	Refresh Interval	Step Frequence	Dead Band (%)

Figure 2.337 The OPC Server panel.

In this panel, you can enter the same information as in the OPC Servers list. The new information is about *OPC Groups*.

For each OPC Server, you can create several groups where each group can contain OPC variables.

OPC Server panel Groups toolbar

Figure 2.338 The Groups toolbar in the OPC Server panel.

New group — Use this button to create a new group for the OPC Server. This is the same as *New->OPC Group* in the popup menu described above. A new OPC group is immediately added to the server configuration.

Delete group — Use this button to delete the selected group(s) from the OPC Server. You can select more than one group by using the *Ctrl*-button and/or the *Shift*-button.

OPC Server panel Group list

The Group list consists of several columns.

Name is the desired name of the group. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the server in the list indicating the problem.

Active determines whether the OPC group is to be active. Only variables in an active group are updated with data changes from the OPC server.

Refresh interval is specified with the time interval, in milliseconds, at which the OPC server is to send data changes to Ethiris.

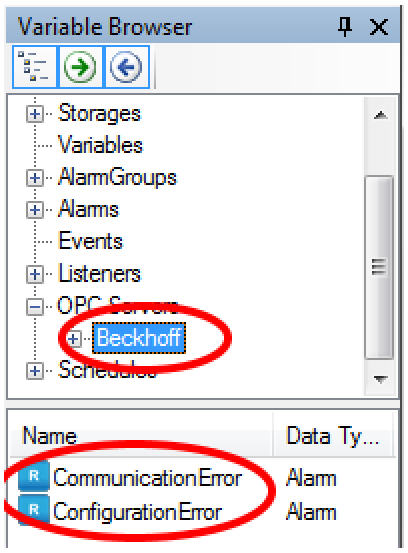
Step frequency specifies the frequency by which the group is to write to the server based on the OPC server protocol's basic time. If you entered the basic time as 50 ms and the step frequency as 4, the group writes every 200 milliseconds (4×50 = 200).

Dead band is a hysteresis that indicates the percent by which the value must be changed for the OPC server to send an update to Ethiris.

OPC Server variables

When defining an *OPC Server*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.339* for an example where an *OPC Server* is selected, and the variables belonging to the OPC Server is presented in the lower pane (ringed in).



Name	Data Ty...
CommunicationError	Alarm
ConfigurationError	Alarm

Figure 2.339 Variables for an OPC Server.

CommunicationError is a read-only alarm variable, which is *true* if there is a communication error to the OPC Server.

ConfigurationError is a read-only alarm variable, which is *true* if there is something wrong with the configuration data for the OPC Server.

2.4.49 OPC Group node

Under the OPC Server node in the treeview, there may be some *OPC Group* nodes.

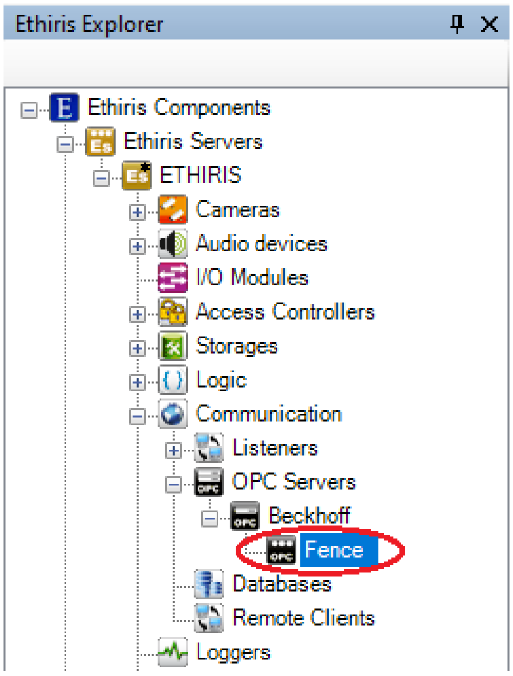


Figure 2.340 An OPC Group node in Ethiris Explorer treeview.

OPC Group popup menu

Right-clicking this node brings up a context menu.

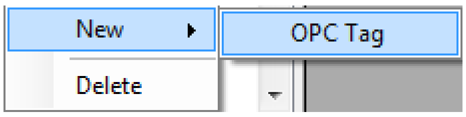


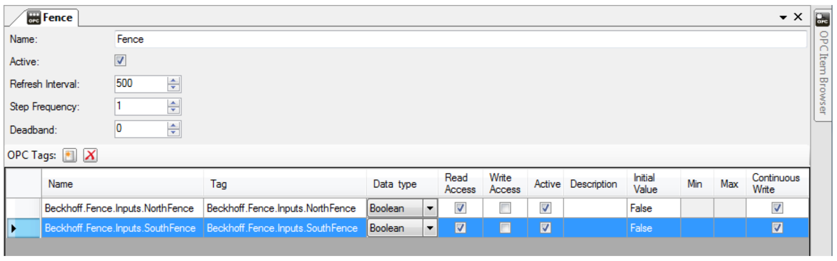
Figure 2.341 The popup menu for an OPC Server node.

New->OPC Tag adds a new OPC Tag to the OPC group. Tags are the actual variables that are communicated between Ethiris Server and the OPC Server. The normal way of adding OPC Tags is to browse for them in the OPC Server. This can be done from the *OPC Group panel* described further down.

Delete Removes the OPC Group from the OPC Server. It is immediately removed from both the treeview and the OPC Group list in the OPC Server panel.

OPC Group panel

Double-clicking an *OPC Group* node in the treeview opens the corresponding panel.



Name: Fence
Active: ✓
Refresh Interval: 500
Step Frequency: 1
Deadband: 0
OPC Tags:

	Name	Tag	Data type	Read Access	Write Access	Active	Description	Initial Value	Min	Max	Continuous Write
	Beckhoff.Fence.Inputs.NorthFence	Beckhoff.Fence.Inputs.NorthFence	Boolean	✓	☐	✓		False			✓
▶	Beckhoff.Fence.Inputs.SouthFence	Beckhoff.Fence.Inputs.SouthFence	Boolean	✓	☐	✓		False			✓

Figure 2.342 The OPC Group panel.

In this panel, you can enter the same information as in the OPC Group list. The new information is about *OPC Tags*.

For each OPC Group, you can create/add several tags.

OPC Group panel Tags toolbar

Figure 2.343 The OPC Tags toolbar in the OPC Group panel.

Add a new OPC Tag

Use this button to create a new OPC tag in the group. This is the same as *New->OPC Tag* in the popup menu described above. A new OPC tag is immediately added to the server configuration.

Delete selected OPC Tag(s)

Use this button to delete the selected tag(s) from the OPC group. You can select more than one tag by using the *Ctrl*-button and/or the *Shift*-button.

OPC Group panel Tag list

The Tag list consists of several columns.

Name is the desired name of the tag. This is the *variable* name that will be used in the script. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the tag in the list indicating the problem.

Tag is the name of the tag in the OPC Server. This is used for identifying the corresponding signal in the OPC Server. When you add tags via the *OPC Item Browser,* this name is automatically filled out.

Data type determines which values are possible for the variable. The alternatives are *Boolean*, *Integer*, *Double,* and *String*.

Read access determines if the variable is readable in, e.g., script.

Write access determines if the variable is writeable in, e.g., script.

Active determines if the variable is active. Only active OPC variables are updated with data changes from the OPC server.

Description is used to describe the purpose of the variable. This description is visible in the *Variable Browser* window.

Initial value is the initial value of the variable. This value applies before the variable is updated with real values from the OPC Server.

Min is the minimum value allowed for the variable.

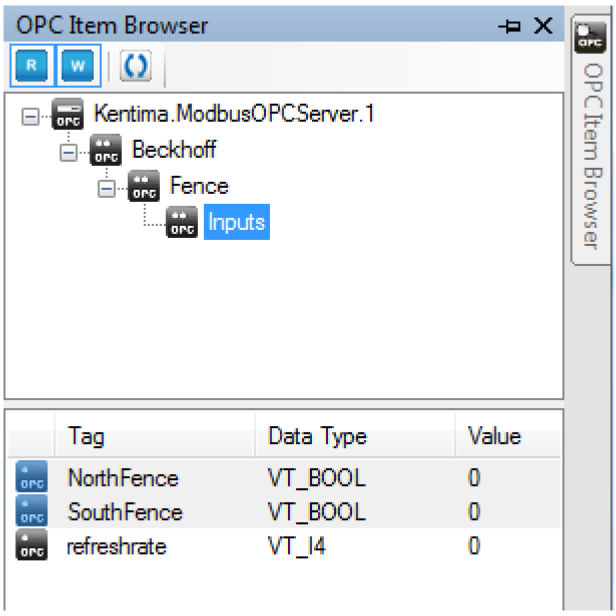
Max is the maximum value allowed for the variable.

Continuous write. When checked, the value of the variable is written to the OPC Server in each cycle. If not checked, the variable value is only written to the OPC Server when the value is changed. Some type of equipment requires a continuous write to work properly.

OPC Item Browser

When adding tags to an *OPC Group*, it is convenient to browse the available tags directly in the OPC Server. Unfortunately, the browsing functionality is not mandatory in the OPC Server specification. This means that it is not certain that you can browse the OPC Server. In that case, you have to manually add the OPC Tags to the group.

In most cases, however, you can browse the OPC Server for available tags.



Tag	Data Type	Value
NorthFence	VT_BOOL	0
SouthFence	VT_BOOL	0
refreshrate	VT_I4	0

Figure 2.344 The OPC Item Browser.

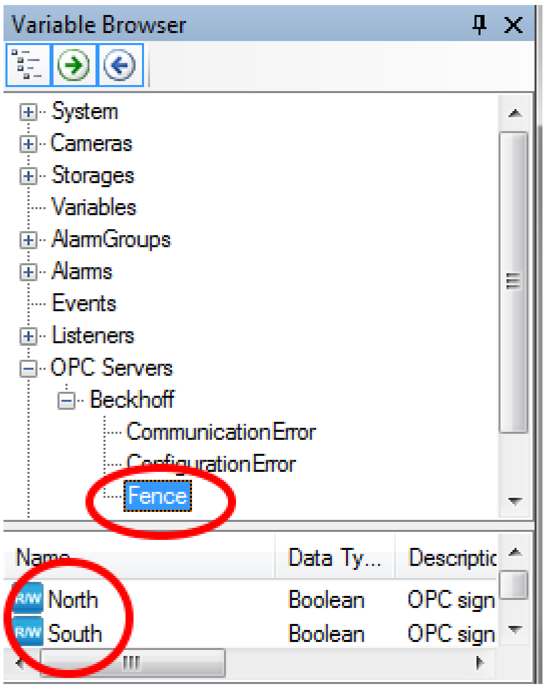
By using the *OPC Item Browser,* you can browse an OPC Server for available signals. In the lower pane are the tags/signals belonging to the currently selected group.

You can select available tags and then drag-and-drop them into the *OPC Tags* list in the OPC group panel.

OPC Group variables

When defining tags in an *OPC Group*, the tags will be visible as *variables* that you can use in a whole lot of ways, e.g. in a script or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.345* for an example where an *OPC Group* is selected, and the variables belonging to the OPC Group are presented in the lower pane (ringed in).



Name	Data Ty...	Descriptic
North	Boolean	OPC sign
South	Boolean	OPC sign

Figure 2.345 Variables for an OPC Group.

The actual variables for a group vary depending on which tags are added to the group.

2.4.50 Databases node

Under the node *Communication* in the treeview, there is a *Databases* node. This is a collection node for all databases defined in the server configuration.

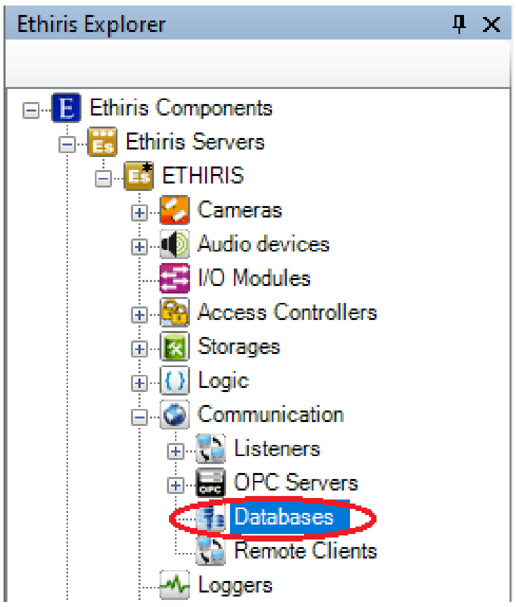


Figure 2.346 The Databases node in Ethiris Explorer treeview.

If you want to use a database to either save information from Ethiris or to read information from and then use somehow in Ethiris, it is here you define the connection to the database(s) you want to use.

Note that there are license restrictions regarding using databases in Ethiris. Read more about that in the license document to the current version of Ethiris.

Ethiris can use connections to multiple databases at the same time, either using ODBC-connections defined in Control panel - Administrative tools - Data sources (ODBC) or directly using installed ODBC drivers.

Databases popup menu

Right-clicking this node opens a menu.

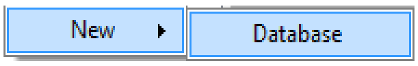


Figure 2.347 Popup menu for the node Databases.

New->Database adds a new database connection to the server configuration. It appears immediately in the treeview as a new database. Should the *Databases* panel be open, it appears there as well.

To make the configuration as simple as possible, we recommend that you search for ODBC data sources or ODBC drivers and then use the guide to configure the parameters. If you are an advanced user and used to configuring ODBC connections, you can easily configure the parameters yourself. Both ways are fine.

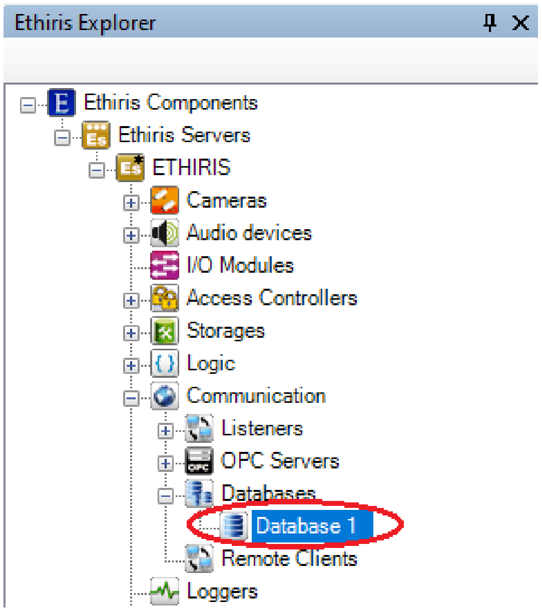
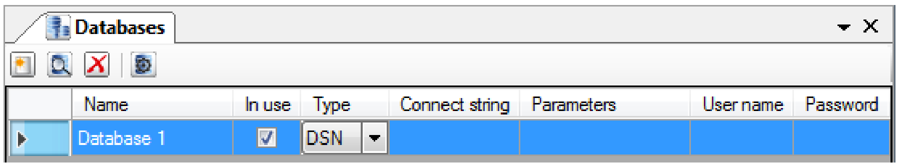


Figure 2.348 New Database added.

Databases panel

Dubble-clicking the *Databases* node opens the corresponding panel.



	Name	In use	Type	Connect string	Parameters	User name	Password
▶	Database 1	✓	DSN				

Figure 2.349 The panel Databases.

This panel consists of a list of all databases defined in the server configuration.

At the top of the panel, there is a toolbar.

Databases toolbar

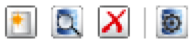


Figure 2.350 The toolbar in the Databases panel.

New Database	Use this button to add a new *Database*. This is the same as selecting *New->Database* in the popup menu described above. A new database is immediately added to the server configuration.
Browse for ODBC data sources and database drivers	Use this button to search for configured ODBC data sources and installed ODBC database drivers in the computer where Ethiris Server is running. Read more about this in section *Add Database* below.
Delete selected Database(s)	Use this button to delete selected *Databases* from the configuration. You can select more than one *Database* by using the *Ctrl* key and/or then *Shift* key.

Test database connection

Use this button to test your configured database connection. A message box will be displayed if the connection to the database succeeds.

If the connection fails, you will get a message about this and also an error message that was returned from the operating system.

If your database connection if of type *Driver,* a guide helping you to configure connection parameters to the database will be displayed*, r*ead more about this in the section *Add Database* below.

Databases panel list

The database list consists of several columns.

Name is the desired name of the database (actually the connection to the database). This name has to be unique in the configuration. If you enter an illegal name, an icon to the left of the database in the list will indicate the error.

In use is ticked as default, meaning that connection is to be made to this database, and that is it can be used from the script.

Type determines the type of connection. You can choose *DSN,* which means that a pre-defined System DSN will be used to connect to the database. In that case, all parameters must be defined outside of Ethiris, and the *connect string* will only be the name of the System DSN you want to use. If you select *Driver,* it means that Ethiris will use that ODBC driver directly and that you must define some parameters for the connection to succeed. The database connection guide will help you define these parameters. You start it by clicking the button.

In case you are using a cluster, it may be an advantage to use the *Driver* type and then configure all parameters. This means that the database connection will work from any member in the cluster (provided that the driver is installed on all members). If you choose to use a System DSN in combination with a cluster, you must make sure that all members of the cluster have identical configurations of this System DSN. It must also have an identical name for all members. When you search for System DSN in a cluster, you can only select System DSN that is available on all members. This is to ensure that the database connection will work from all members of the cluster,

Connect string is the name of the System DSN if the type is *DSN*. If type is *Driver,* enter the connection string the driver requires, usually the name of the driver. To some extent, you can get help with this via the built-in guide. Click the button to start the wizard.

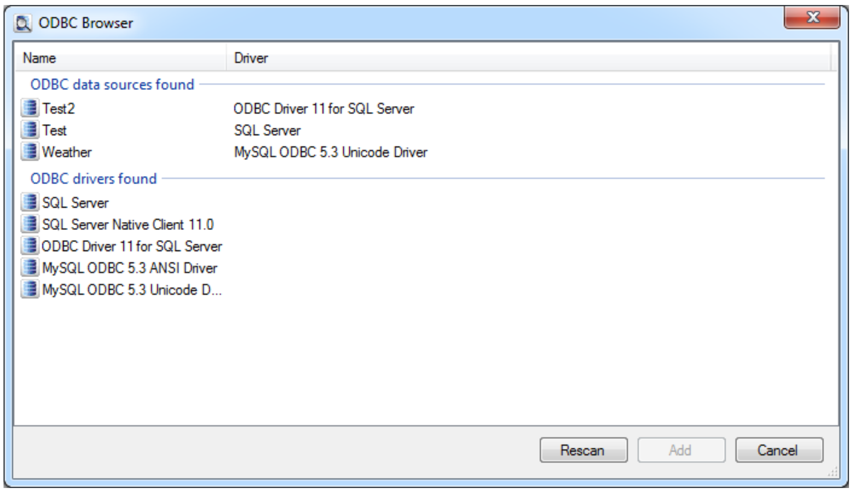
Parameters are the additional parameters that may be required for connection to the database. Examples of parameters may be SERVER, PORT, DATABASE, which simply point to the server/database you want to connect to.

User name is the user name that the database requires to allow a connection.

Password is the password that the database requires to allow a connection.

Add Database

To add a database to the configuration, we recommend that you search for available ODBC connections and ODBC drivers on the server computer. Do this by clicking the button . A dialog will then appear:



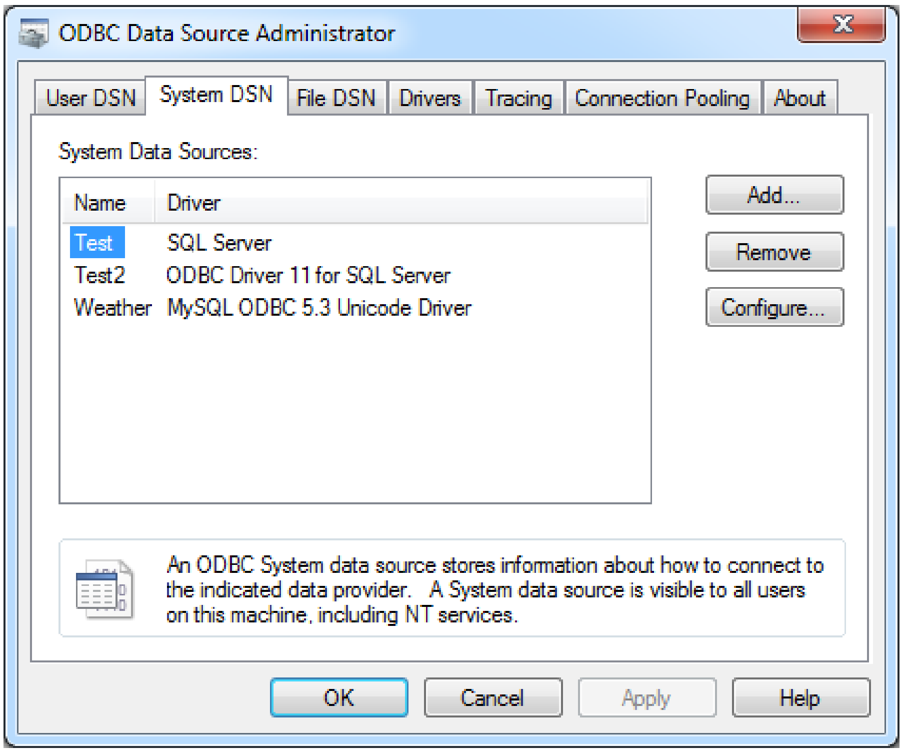
ODBC Browser

Name	Driver
ODBC data sources found	
Test2	ODBC Driver 11 for SQL Server
Test	SQL Server
Weather	MySQL ODBC 5.3 Unicode Driver
ODBC drivers found	
SQL Server	
SQL Server Native Client 11.0	
ODBC Driver 11 for SQL Server	
MySQL ODBC 5.3 ANSI Driver	
MySQL ODBC 5.3 Unicode D...	

Figure 2.351 Search result.

In the upper group, you see the defined ODBC data sources on your computer.

These are created in Control Panel - Administrative Tools - ODBC Data Sources - System DSN. Note that Ethiris Server can only use ODBC data sources installed under System DSN as only those are available to the server running as a service.



ODBC Data Source Administrator

System Data Sources:

Name	Driver
Test	SQL Server
Test2	ODBC Driver 11 for SQL Server
Weather	MySQL ODBC 5.3 Unicode Driver

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure 2.352 ODBC data sources defined on the computer.

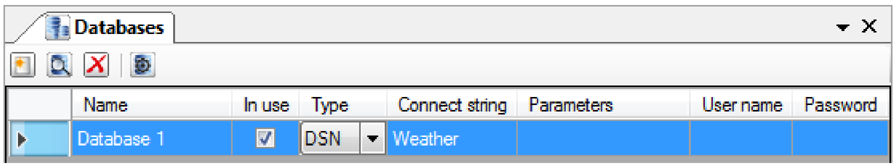
In the lower group, you see the ODBC drivers that are installed on the computer.

Double click on the ODBC data source or ODBC driver you want to use for connection to your database.

Example one

In the first example, we will use an ODBC data source as an example.

Double click on *Weather* in the search result list in *Figure 2.352 ODBC data sources defined on the computer.*

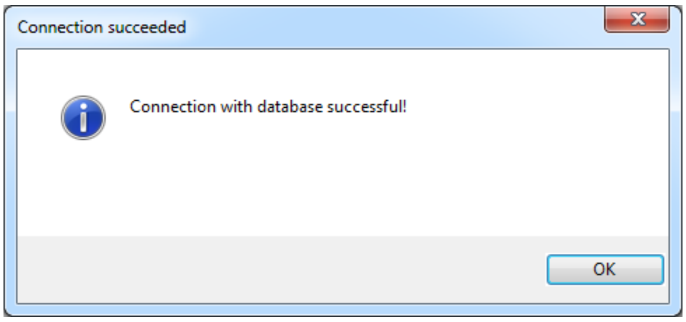


	Name	In use	Type	Connect string	Parameters	User name	Password
▶	Database 1	✓	DSN	Weather			

Figure 2.353 Database based on the ODBC data source added.

Now click on the button to test if the connection works.

If everything goes well, the following message box is displayed:


Connection succeeded

Connection with database successful!

Figure 2.354 Database connection succeeded.

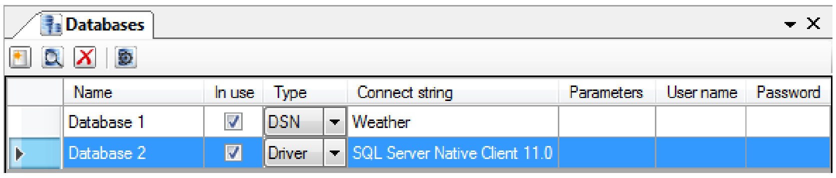
In this case, we have done the entire configuration of the database connection through the Control Panel - Administrative tools - ODBC Data sources - System DSN, and we only refer to that from Ethiris.

How to configure ODBC connections via the control panel is not within the scope of this manual.

Example two

For example two, we use an ODBC driver and take advantage of the wizard to configure the parameters.

Double click on *SQL Server Native Client 11.0 to add a database connection that uses this driver to connect to the database.*



	Name	In use	Type	Connect string	Parameters	User name	Password
	Database 1	✓	DSN	Weather			
▶	Database 2	✓	Driver	SQL Server Native Client 11.0			

Figure 2.355 Database based on ODBC driver added.

The next step is to click on the button to test the connection to the database.

The wizard appears and starts by checking for databases that can communicate with this driver.

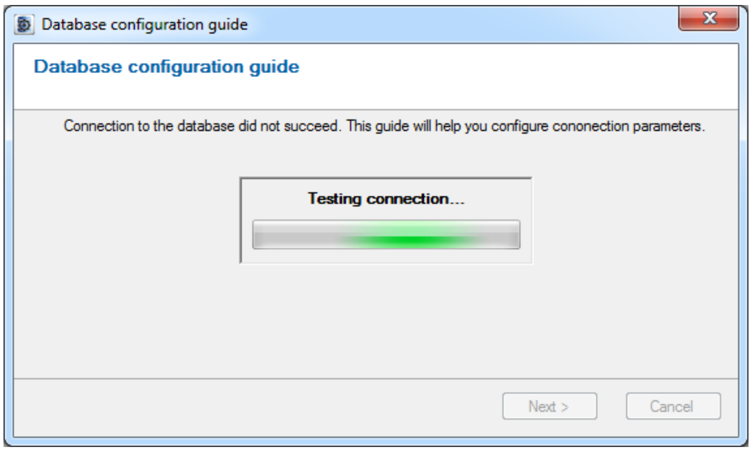

Database configuration guide
Database configuration guide
Connection to the database did not succeed. This guide will help you configure cononection parameters.
Testing connection...
Next >
Cancel

Figure 2.356 The wizard tests the connection to the database.

If the wizard finds any computers on the network that have databases that can communicate with the selected driver, the computers appear in a list.

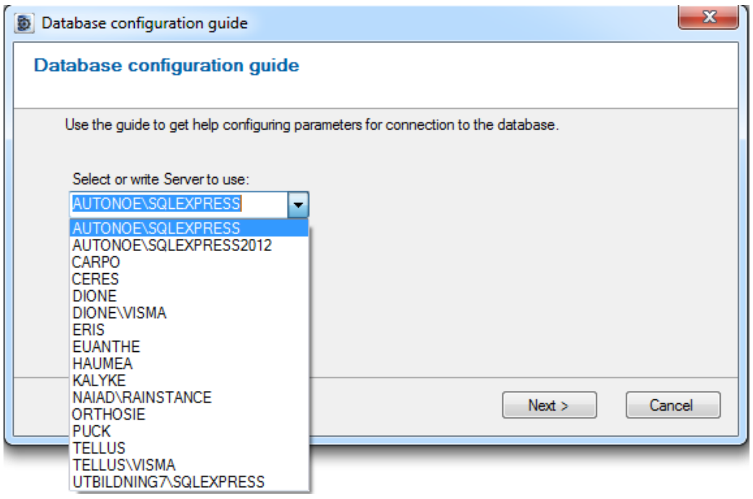


Figure 2.357 Select server to use.

In the example, we select *Eris* and then click *Next*. The wizard lists the available databases on the selected server.

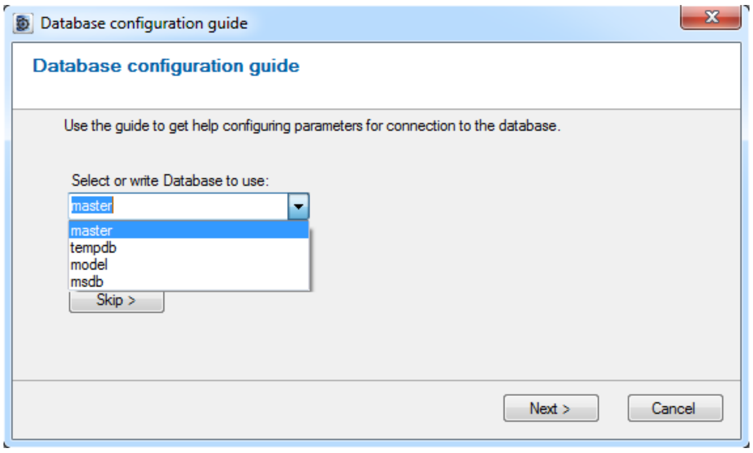
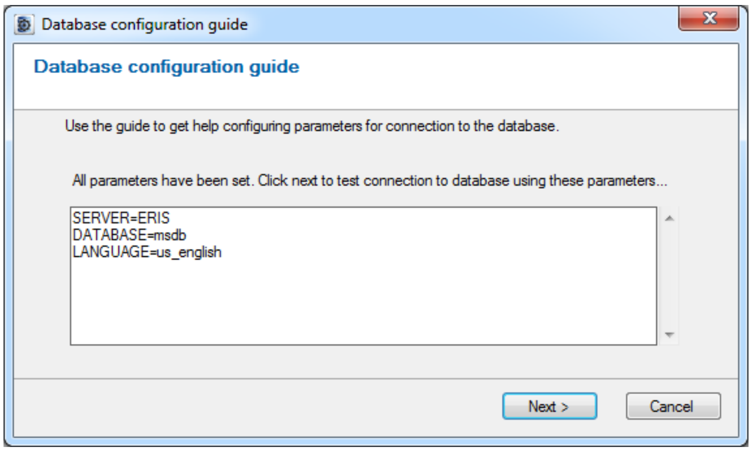

Database configuration guide
Database configuration guide
Use the guide to get help configuring parameters for connection to the database.
Select or write Database to use:
master
master
tempdb
model
msdb
Skip >
Next >
Cancel

Figure 2.358 Select database to connect to.

In the example, we choose to use the database *msdb.* Then click *Next.* If you know what you are doing and want to skip this parameter in the wizard, click *Skip >.*



Database configuration guide

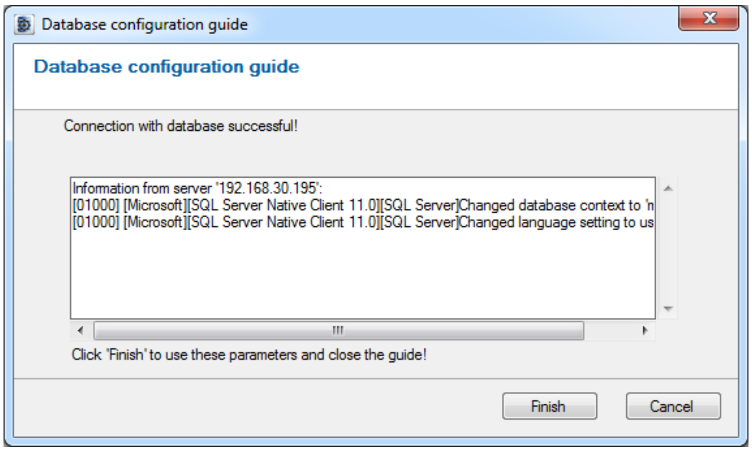
Use the guide to get help configuring parameters for connection to the database.

All parameters have been set. Click next to test connection to database using these parameters...

```
SERVER=ERIS
DATABASE=msdb
LANGUAGE=us_english
```

Figure 2.359 All parameters have been configured.

The wizard lists all parameters that will be used for this database connection. Click *Next* to test the connection to the database.



Database configuration guide

Connection with database successful!

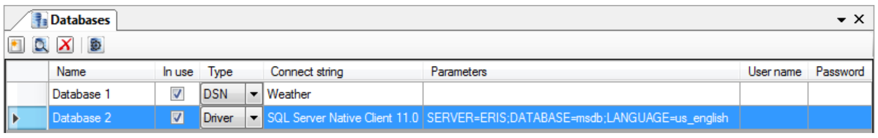
```
Information from server '192.168.30.195':
[01000] [Microsoft][SQL Server Native Client 11.0][SQL Server]Changed database context to 'n
[01000] [Microsoft][SQL Server Native Client 11.0][SQL Server]Changed language setting to us
```

Click 'Finish' to use these parameters and close the guide!

Figure 2.360 Connection to the database successful.

In this case, the connection was successful. When the connection was made, some information was returned from the driver and is displayed in the dialog.

Click on *Finish* to close the wizard and use these parameters.



Name	In use	Type	Connect string	Parameters	User name	Password
Database 1	✓	DSN	Weather			
Database 2	✓	Driver	SQL Server Native Client 11.0	SERVER=ERIS;DATABASE=msdb;LANGUAGE=us_english		

Figure 2.361 The parameters are used for the database connection.

Depending on configuration in the database, you may need to provide a user name and password for the wizard to help with the parameters. In that case, close the wizard and enter the username and password in the Databases panel. Then start the wizard by clicking the button again.

The wizard only starts if the connection to the database fails, and the wizard thinks it can help with the configuration of the parameters.

If you already know what parameters and parameter values are required for connecting to your database, you can enter them in the *Parameters* field manually.

2.4.51 Database node

Under the node *Databases* in the treeview, there might be some *Database* nodes.

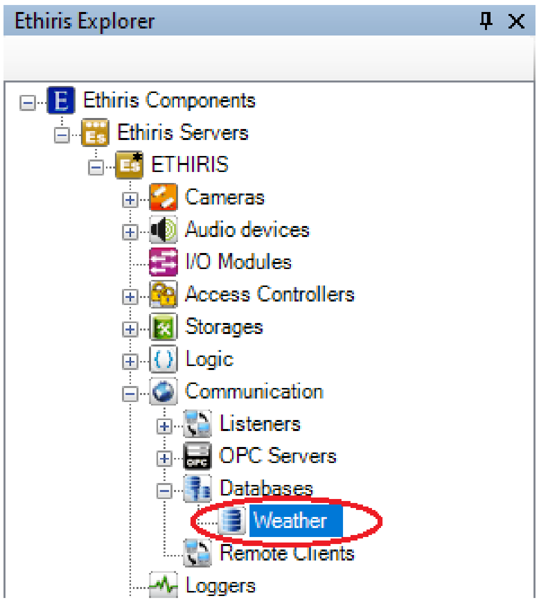


Figure 2.362 One Database node in the Ethiris Explorer treeview.

Database popup menu

Right-clicking this node opens a menu.

Figure 2.363 Popup menu for a Database node.

Delete deletes the database from the server configuration. It disappears immediately from both the treeview and the client list in the *Databases* panel.

Database variables

For each database that is added to the list, a system alarm will automatically be created in the Ethiris Server data store.

2.4.52 Remote Clients node

Last under the Communication node in the treeview, there is the *Remote Clients* node. This is a collection node of all remote clients that are defined in the server configuration.

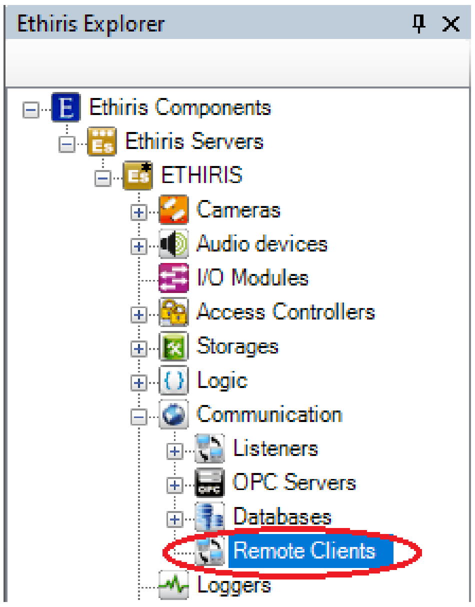


Figure 2.364 The Remote Clients node in the Ethiris Explorer treeview.

The main purpose of this section is to be able to monitor your clients from Ethiris Server. If the connection between client and server is broken, an alarm is activated, provided you have activated the connection supervision for the client.

Remote Clients popup menu

Right-clicking this node opens a menu.

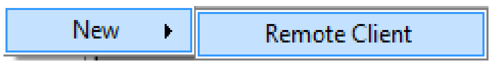


Figure 2.365 Popup menu for the node Remote Clients.

New->Remote Client adds a new remote client to the server configuration. It appears immediately in the treeview as a new remote client. Should the *Remote Clients* panel be open, it appears there as well.

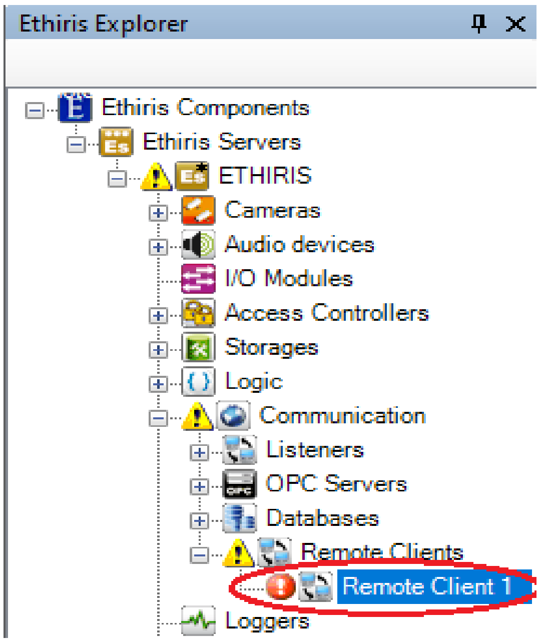
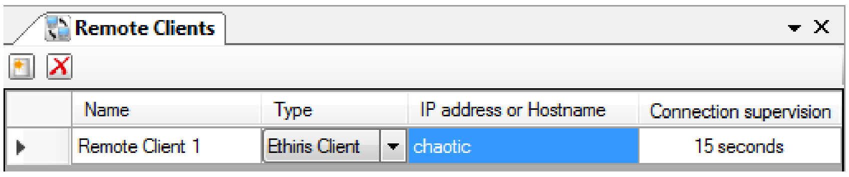


Figure 2.366 New Remote Client added.

Note the icon indicating an error to the left of the new client node. This is due to the new remote client has not yet an IP address defined. There are warning icons further up the tree to indicate that something is wrong in the configuration.

Remote Clients panel

Double-clicking the *Remote Clients* node opens the corresponding panel.



	Name	Type	IP address or Hostname	Connection supervision
▶	Remote Client 1	Ethiris Client	chaotic	15 seconds

Figure 2.367 The Remote Clients panel.

This panel consists of a list of all remote clients defined in the server configuration.

At the top of the panel, there is a toolbar.

Remote Clients toolbar

Figure 2.368 The toolbar in the Remote Clients panel.

New Remote Client	Use this button to add a new Remote Client. This is the same as selecting the menu *New->Remote Client* in the popup menu described above. A new remote client is immediately added to the server configuration.
Delete selected Remote Client(s)	Use this button to delete the selected Remote Clients from the configuration. You can select more than one Remote Client by using the *Ctrl* key and/or the *Shift* key.

Remote Clients panel client list

The client list consists of several columns.

Name is the desired name of the remote client. This name has to be unique in the configuration. If you enter an illegal name, an icon to the left of the client in the list will indicate the error.

Type determines the type of client. As of today, you can only select *Ethiris Client*.

IP address or Hostname determines the computer where the client runs.

Connection supervision is by default *Off*, which means that no monitoring of the connection occurs. If you want to monitor the client connection, you have to set a time. The time can be set between 1 second and 1 day. The time determines how long a disconnection is acceptable. In other words, the time it will take before the alarm gets activated in case of a disconnection. Note that the time before Ethiris discovers a disconnection can vary. If you deliberately close Ethiris Client, it is discovered immediately. In that case, the alarm will be activated after the time specified. However, if someone has disconnected the network cable, it can take up to 45 seconds before Ethiris discovers the disconnection. In that case, the alarm will be activated after the preset time + up to 45 seconds.

2.4.53 Remote Client node

Under the Remote Clients node, there might be some *Remote Client* nodes.

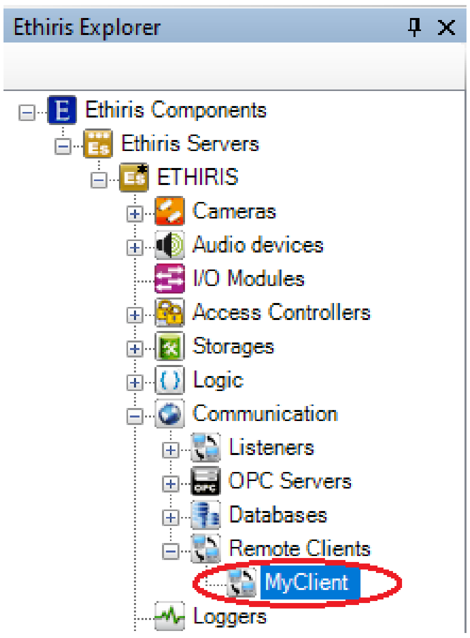


Figure 2.369 A Remote Client node in the Ethiris Explorer treeview.

Remote Client popup menu

Right-clicking this node opens a menu.

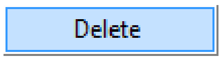


Figure 2.370 Popup menu for a Remote Client node.

Delete deletes the remote client from the server configuration. It disappears immediately from both the treeview and the client list in the *Remote Clients* panel.

Remote Client variables

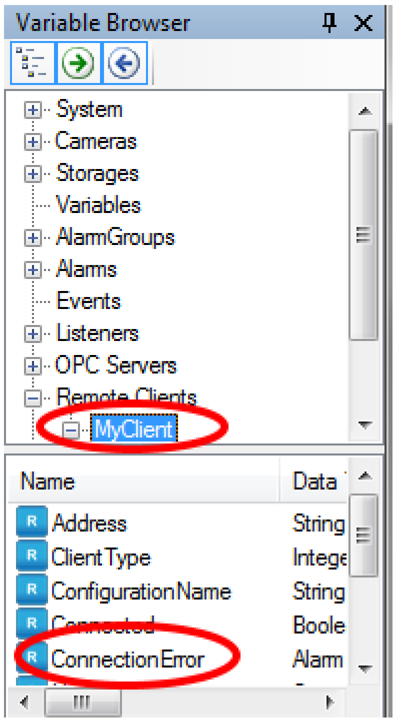
For each client, which is added to the list of remote clients, several variables will automatically be created in the Ethiris Server data store.

When you open the *Script* panel in Ethiris Admin, there is a corresponding tool window called *Variable Browser,* which is docked to the right in the main window. The *Variable Browser* tool window contains all the available variables in the Ethiris Server data store. See *Figure 2.371* for an example of when a *Remote Client* is selected, and the corresponding variables are displayed in the lower pane.

The most commonly used variables are ringed in.

Connected is a read-only variable. When its value is *true,* the client is connected.

ConnectionError is also a read-only variable that is an alarm. An alarm variable can be used both directly as a *Boolean,* and you can also use its internal variables *Acknowledged*, *Blocked,* and *State*. When this variable is *true,* it means that the alarm is active.



Name	Data
Address	String
Client Type	Intege
ConfigurationName	String
Connected	Boole
ConnectionError	Alarm

Figure 2.371 Variables for a remote client.

2.4.54 Loggers node

Under each Ethiris Server in the treeview, there is a *Loggers* node. It is a collection node for all loggers defined in the server.

The purpose of the loggers is to log variable values from Ethiris Server. The logged values can then be presented in Ethiris Client in a *Label* or *LED* depending on the data type of the value. All data types (*Boolean*, *String*, *Integer,* and *Double*) can be presented in a *label,* and the data type *Boolean* can also be presented in an *LED*.

Labels and LEDs can be used in camera views in Ethiris Client to present variable values both in *Live* and, in the case of logged variable data, in the *Player*. A logged variable that is displayed in *Live* will present the current value of the underlying variable. When the corresponding logged variable is presented in the Player, the value matching the timestamp of the recorded video will be presented as long as there is a logged value for that time.

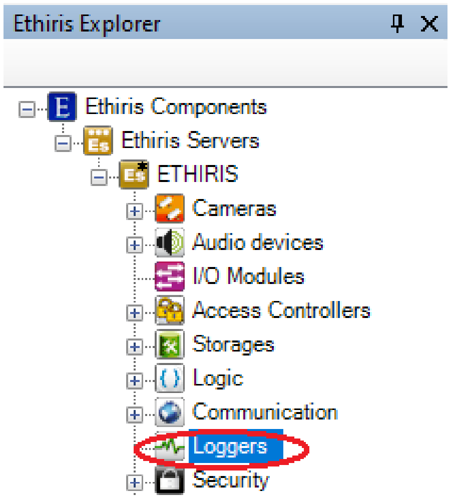


Figure 2.372 The node Loggers in the treeview of Ethiris Explorer.

Loggers popup menu

Right-click on this node will open a menu.

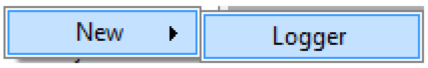


Figure 2.373 Popup menu for the node Loggers.

New->Logger adds a new logger to the server configuration. It will immediately be visible in the treeview as a new logger node. Should the panel *Loggers* be open, it will be visible there as well.

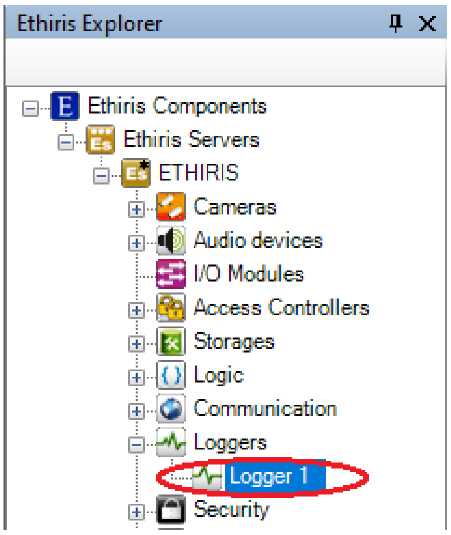
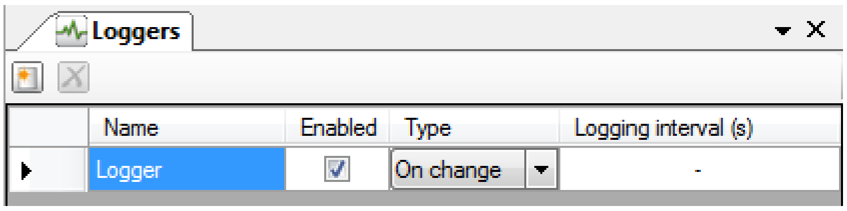


Figure 2.374 New logger added.

Loggers panel

Double-click on the node *Loggers* in the treeview will open the corresponding panel.



	Name	Enabled	Type	Logging interval (s)
▶	Logger	✓	On change	-

Figure 2.375 The panel Loggers.

This panel consists of a list of all loggers currently defined in the server configuration.

At the top of the panel, there is a toolbar.

Loggers panel toolbar

Figure 2.376 The toolbar in the panel Loggers.

New Logger	Use this button to create a new logger. Clicking this button is the same as selecting the menu item *New->Logger* in the popup menu described above. A new logger will immediately be added to the configuration.
Delete selected Logger(s)	Use this button to delete selected loggers from the configuration. You can select more than one logger by using the *Ctrl*-key and/or the *Shift*-key.

Loggers panel logger list

The logger list consists of several columns.

Name is the desired name of the logger. This name has to be unique in the configuration. If you enter an illegal name there will be an icon to the left of the logger in the list that indicates the error.

Enabled determines if the logger will be enabled or disabled when Ethiris Server starts. This state can also be controlled from a script by setting the variable *Enable* that exists for each logger. If the logger is disabled, none of the values of the variables in the logger will be logged.

Type determines what type of logging to use. For the time being, only *On change* is available. This selection means that as soon as the value in the underlying variable changes, the value is logged together with the current timestamp.

Logging interval is not used for now but is reserved for future use. The idea is to be able to select logging of values on change, but at a minimum interval as defined here.

2.4.55 Logger node

Under the node Loggers in the treeview, there might be one or several *Logger* nodes.

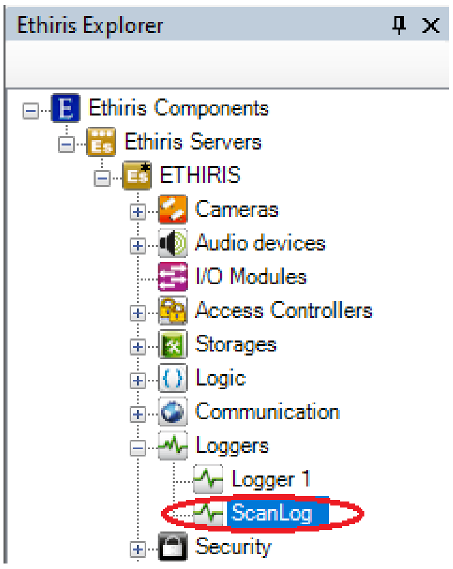


Figure 2.377 A Logger node in the Ethiris Explorer treeview.

Logger popup menu

Right-click on this node will open a menu.

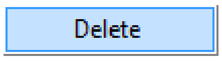
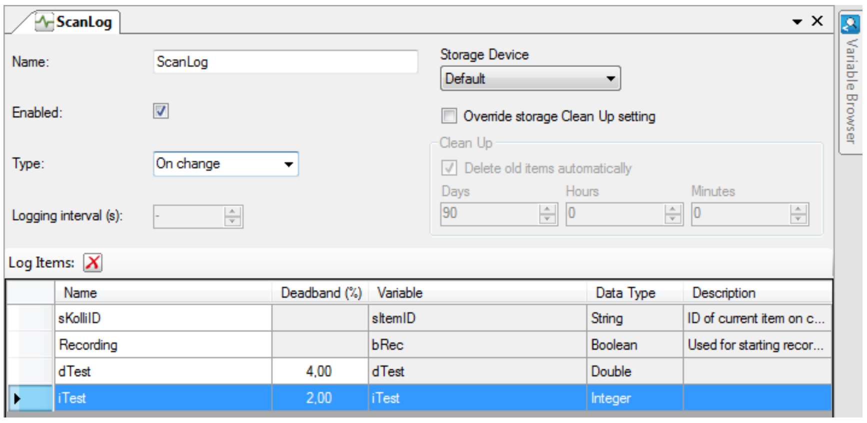


Figure 2.378 Popup menu for a Logger node.

Delete deletes the logger from the server configuration. It will be deleted from both the treeview and the logger list in the panel *Loggers*.

Logger panel

Double-click on a *Logger* node in the treeview will open a panel that displays an overview of the settings for the logger and which variables that are part of the logger.


ScanLog

Name: ScanLog

Enabled: ✓

Type: On change

Logging interval (s): -

Storage Device: Default

Override storage Clean Up setting

Clean Up

✓ Delete old items automatically

Days: 90 Hours: 0 Minutes: 0

Log Items:

	Name	Deadband (%)	Variable	Data Type	Description
	sKolliID		sItemID	String	ID of current item on c...
	Recording		bRec	Boolean	Used for starting recor...
	dTest	4,00	dTest	Double	
▸	iTest	2,00	iTest	Integer	

Figure 2.379 The panel Logger.

Logger panel general settings

At the top of the panel, there are the same settings as in the logger list described above. Here also are settings about where to store the logged values and how this data shall automatically be cleaned up.

Storage device determines which one of the storage devices that shall be used for storing this loggers logged values.

Override storage Clean Up setting should be ticked if you don't want to use the general settings that are defined for all storages.

Clean Up

Delete old items automatically should be ticked if you want automatic clean up.

Days, Hours & Minutes are used for setting how old data can be before it is deleted automatically.

Logger panel toolbar

Figure 2.380 The toolbar in the panel Logger.

Delete selected Log item(s)

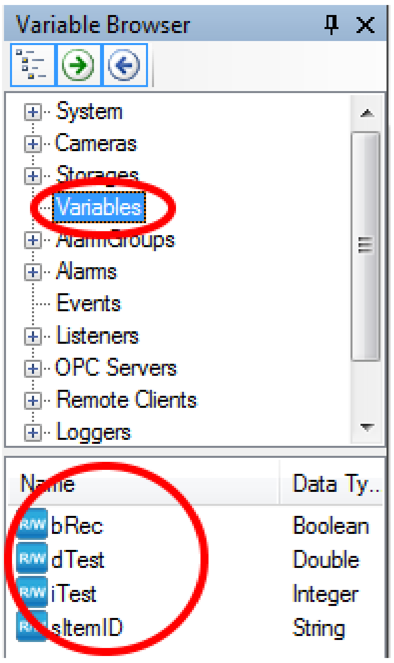
Use this button to delete selected log items from the configuration. You can select more than one item by using the *Ctrl*-key and/or the *Shift*-key.

Logger panel log item list

The only way to add desired variables to a logger is to drag them from the *Variable Browser* window that is docked to the right in this panel. Select one or several variables in the browser and then drag them to the log item list.

Only user-defined variables can be logged.

A current limitation is that only so-called user-defined variables can be logged. These are the variables in the node *Variables* in the *Variable Browser* window. See *Figure 2.381*.



Name	Data Ty..
bRec	Boolean
dTest	Double
iTest	Integer
sItemID	String

Figure 2.381 The Variable Browser in the Logger panel.

Still, it's easy to overcome this limitation if you want to log any of the other variables in Ethiris Server's data store. Just create a user-defined variable with the same data type and then simply copy the value by using a script.

E.g., if we want to log the current frame rate for the camera *Door* in our user-defined variable *dTest,* we can write a row of a script like this:

```
dTest = Door.CurrentPictureRate;
```

The log item list consists of several columns. All information in the list except for *Name* and *Deadband* is read-only.

Name is used for representing the log item.

Deadband can be used for variables of the data types *Integer* or *Double* as long as *Min* and *Max* are specified for the underlying variable. Deadband is expressed in percent of the actual value range, which is why you have to know min and max for the variable value. If we, for example, have an integer variable with a value range of 0 – 200 and a deadband of 5.00 %, it means that the value has to change at least 10 units for logging to occur, e.g., from 53 -> 63.

Variable presents the name of the underlying variable, i.e., the variable whose value will be logged. This name is automatically created when the variable is dragged from the *Variable Browser* window.

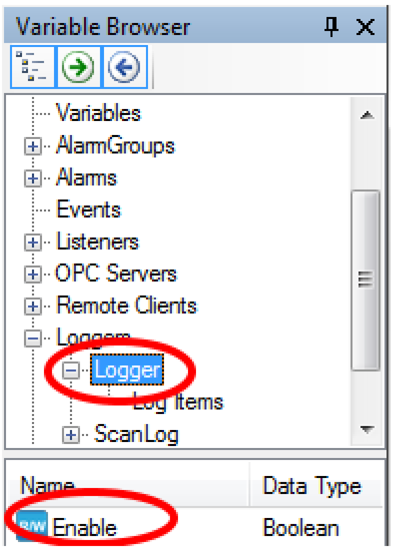
Data type presents the data type of the underlying variable. This information is automatically created when the variable is dragged to the log item list.

Description presents a possible description of the underlying variable.

Logger variables

When a logger is defined, some variables are automatically created that can be used in several different ways, e.g. in a script, send via OPC to other systems or present the information in Ethiris Client-

When you open the *Script* panel in Ethiris Admin, there is a corresponding tool window *Variable Browser,* which is docked to the right in the main window. The *Variable Browser*window contains all available variables in Ethiris Server's data store. See *Figure 2.244* for an example of when a *Logger* is selected and the corresponding variables are displayed in the lower panel (ringed in).



Name	Data Type
Enable	Boolean

Figure 2.382 Variables for a Logger.

We will discuss the Script and variables later in the manual.

Directly under the Logger object, there is only one variable, *Enable*. The variables you want to log is under the *Log Items* object.

Enable is used to enable/disable the logger. If this variable is not used in a script, the setting for *Enable* in the Logger panel described above applies.

<*Log item*> is used for presenting logged values in the player in Ethiris Client. Connect a log item to a label or an LED to present the value. These are found under the *Log Items* object.

2.4.56 Security node

Under the Ethiris Server node in the treeview, there is a *Security* node. This is where you configure common security settings for the whole Ethiris Server.

A number of specific user operations are defined in Ethiris. For each such user operation, you can, if you want, specify that the user who performs the operation must be a member of a certain user group, either in Ethiris or in the Windows user system. This may be either a local user group on the computer running Ethiris Server or a global user group in a domain or Active Directory if the computers and users involved are members of a domain. To be able to specify a user group in the domain, it is necessary for both the computer running Ethiris Server to be a member of the domain and the logon entered by the user to be for an account in the same domain.

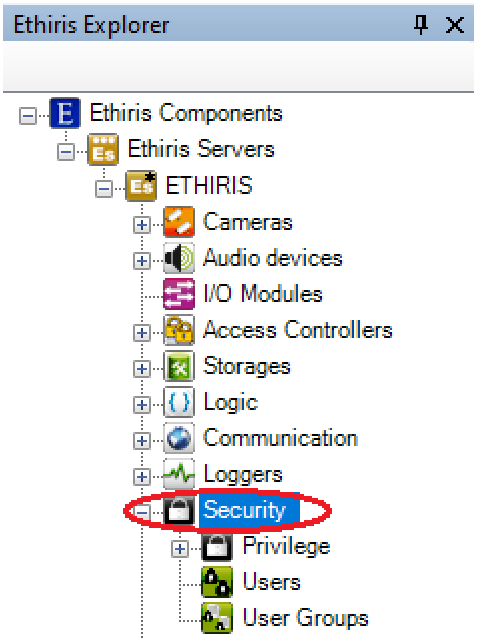


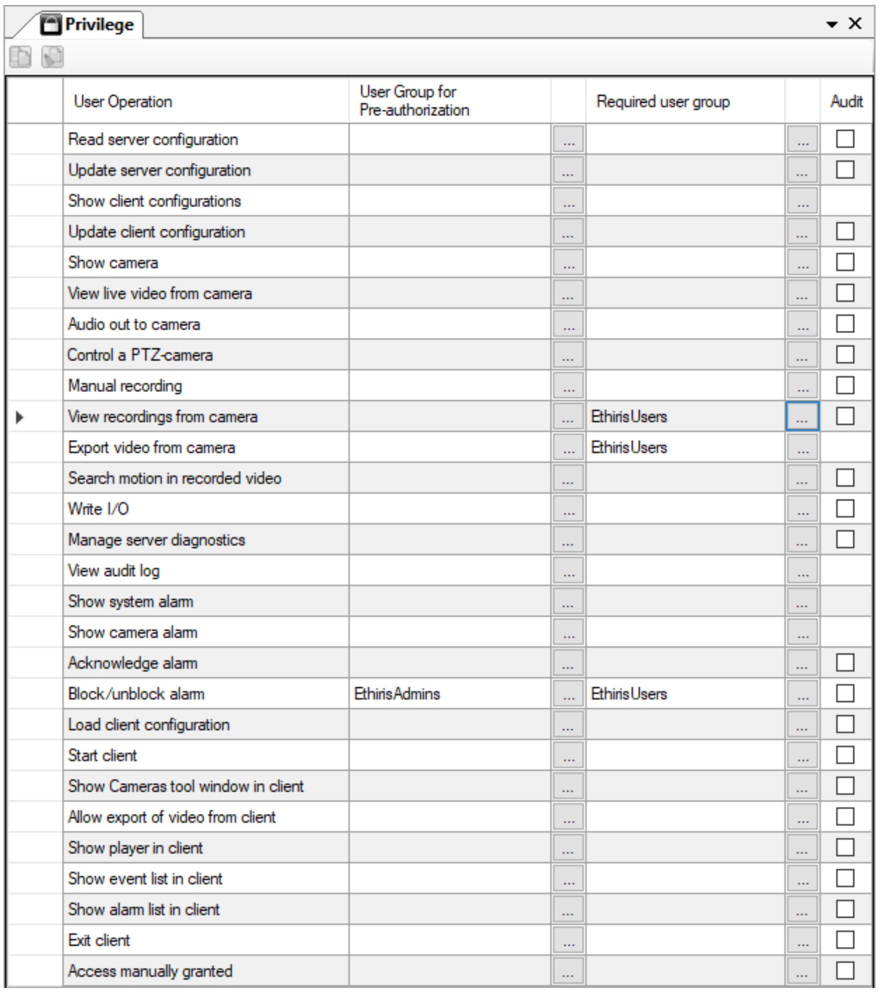
Figure 2.383 The Security node in Ethiris Explorer treeview.

Security panel

The node *Security* has no corresponding panel. For convenience, the panel *Privilege* is opened when the node *Security* is double-clicked.

Privilege panel

Double-clicking the *Privilege* node in the treeview opens the corresponding panel.



Privilege

	User Operation	User Group for Pre-authorization		Required user group		Audit
	Read server configuration		...		...	☐
	Update server configuration		...		...	☐
	Show client configurations		...		...	
	Update client configuration		...		...	☐
	Show camera		...		...	☐
	View live video from camera		...		...	☐
	Audio out to camera		...		...	☐
	Control a PTZ-camera		...		...	☐
	Manual recording		...		...	☐
▶	View recordings from camera		...	EthirisUsers	...	☐
	Export video from camera		...	EthirisUsers	...	
	Search motion in recorded video		...		...	☐
	Write I/O		...		...	☐
	Manage server diagnostics		...		...	☐
	View audit log		...		...	
	Show system alarm		...		...	
	Show camera alarm		...		...	
	Acknowledge alarm		...		...	☐
	Block/unblock alarm	EthirisAdmins	...	EthirisUsers	...	☐
	Load client configuration		...		...	☐
	Start client		...		...	☐
	Show Cameras tool window in client		...		...	☐
	Allow export of video from client		...		...	☐
	Show player in client		...		...	☐
	Show event list in client		...		...	☐
	Show alarm list in client		...		...	☐
	Exit client		...		...	☐
	Access manually granted		...		...	☐

Figure 2.384 The Privilege panel.

This panel consists of a list of all user operations for which you can require that the user logs in for access.

The following user operations exist:

Read server configuration is for displaying the Ethiris Server configuration in Ethiris Admin. If a *Required User Group* is specified, the user has to log in as a member of the specified group to be able to load the Ethiris Server configuration in Ethiris Admin.

Update server configuration is for saving the Ethiris Server configuration from Ethiris Admin.

Show client configurations is for showing client configurations when a compatible client browses an Ethiris Server for available configurations. If this privilege is set, all configurations will be hidden from users not having this privilege. Note that a user lacking this privilege will still be able to load the client configuration.

Update client configuration is for saving/updating the client configurations maintained by the Ethiris Server from Ethiris Admin.

Show Camera is for being able to see cameras in the client. If a *Required User Group* is specified and the user is not logged in, it will seem as if the cameras in the configuration don't exist.

View live video from camera is for viewing live video from cameras in Ethiris Client. To be able to view live video from a camera that is part of the Ethiris Server configuration where this user operation has a *Required User Group* specified, you have to log in to Ethiris Client.

Audio out to camera is for allowing an operator to talk out to a camera (that has some kind of speaker connected to it).

Control a PTZ-camera is for being able to control a PTZ camera, i.e., pan, tilt, and zoom.

Manual recording is for being able to start event recording from Ethiris Client by clicking the button for manual recording in live.

View recordings from camera is for view recorded video in Ethiris Client.

Export video from camera is for exporting recorded video from Ethiris Client. This privilege controls the authority for the export.

Search motion in recorded video is for being able to search for motion in recorded video from Ethiris Client.

Write I/O is for being able to write values to variables from Ethiris Client. E g setting a Boolean value via a button in Ethiris Client.

Manage server diagnostics is for managing logfiles from the Ethiris Server.

View audit log is for being able to see audit trail information in the *Event list* in Ethiris Client.

Show system alarm is for viewing system alarms in the *Alarm list*. Alla alarms without an associated camera count as a system alarm.

Show camera alarm is for viewing camera alarms in the *Alarm list*. Alla alarms without at least one associated camera count as a camera alarm.

Acknowledge alarm is for acknowledging alarms in the *Alarm list* in Ethiris Client.

Block/unblock alarm is for blocking/unblocking alarms in the *Alarm list* in Ethiris Client.

Load client configuration is to be able to load the client configuration in the compatible client. Client configurations having this privilege set will be hidden for users lacking this privilege when browsing available client configurations.

Start client is for starting Ethiris Client. If this operation requires log in, a log in dialog is automatically displayed when starting Ethiris Client.

Show Cameras tool window in client is for displaying the tool window Cameras in Ethiris Client. If you do not have the privilege of this, the tool window Cameras not be visible.

Allow export of video from client is for exporting video from the Player in Ethiris Client. To be able to export video, you have to have both this privilege and the *View recordings from camera* privilege. This privilege controls the availability of the various export controls in Ethiris Client, such as menus and buttons.

Show player in client is for displaying the Player panel in Ethiris Client. If you do not have the privilege of this, the Player panel will not be visible.

Show event list in client is for displaying the Event list panel in Ethiris Client. If you do not have the privilege of this, the Event list panel will not be visible.

Show alarm list in client is for displaying the Alarm list panel in Ethiris Client. If you do not have the privilege of this, the Alarm list panel will not be visible.

Exit client is for closing Ethiris Client.

Access manually granted is for allowing an operator to manually open a door via a connected Access Controller.

Privilege panel toolbar

Figure 2.385 The toolbar in the Privilege panel.

Copy row/cell	Use this button to copy the content of the currently selected row/cell.
Paste selected rows/cells	Use this button to paste into the selected row(s). You can select more than one operation by using the *Ctrl*-button and/or the *Shift*-button.

Privilege panel operation list

The operation list consists of several columns.

User Operation is the name of the user operation for which you can define a required user group

User Group for Pre-authorization is the name of the Windows user group that will be used for pre-authorization. This group can only be entered provided that *Required User Group* has been defined. The purpose of pre-authorization is to require a so-called double log in. This means that before the regular user can log in, a user that is a member of the specified pre-authorization user group has to log in first.

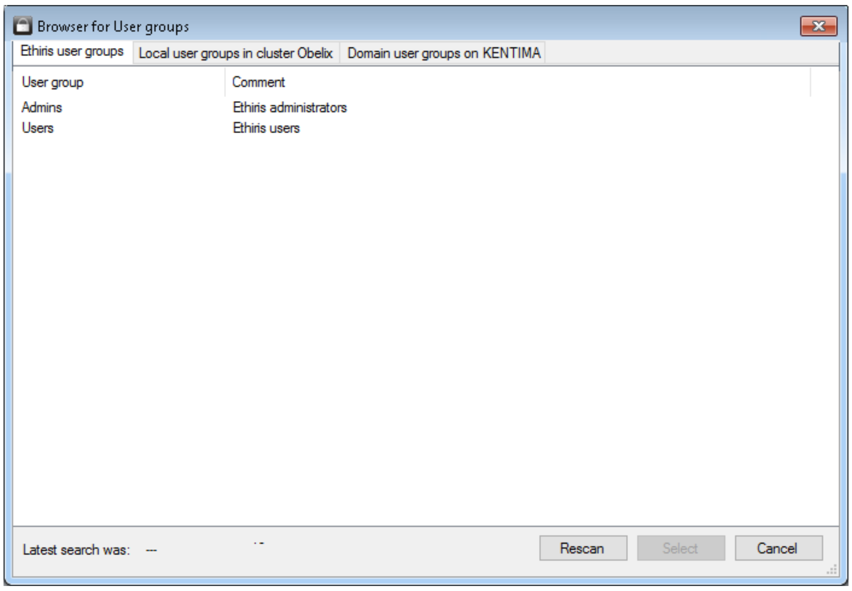
Browse button can be used to browse for available user groups. A dialog will appear to show the available options.

Required User Group is the name of the Windows user group that the logged in user has to be a member of to get access to the operation.

Browse button can be used to browse for available user groups.

Audit shall be checked if you want to log each time a certain user operation is performed. The audit log is available in the *Event list* in Ethiris Client.

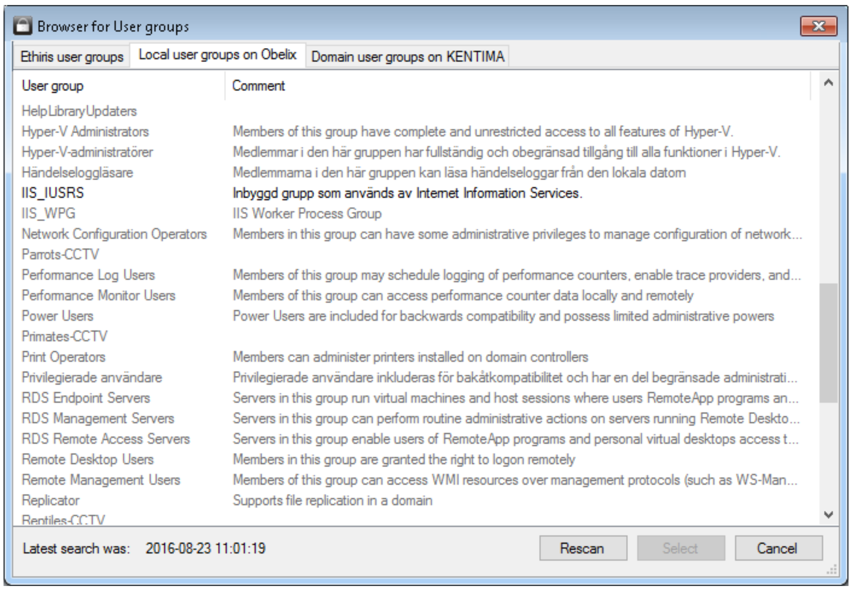
The dialog displays different tabs depending on the configurations and whether the server/ cluster is a member of a domain.



User group	Comment
Admins	Ethiris administrators
Users	Ethiris users

Figure 2.386 An example of the dialog "Browser for User groups", the tab Ethiris user groups.

The tab *Ethiris user groups* list the user groups that are defined in the Ethiris Server configuration. If none are defined, this tab is hidden.

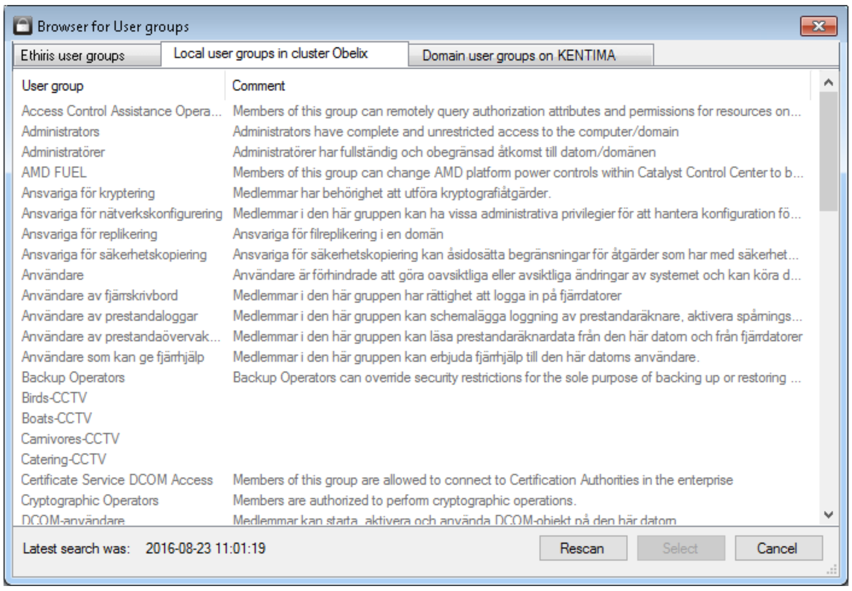


User group	Comment
HelpLibraryUpdaters	
Hyper-V Administrators	Members of this group have complete and unrestricted access to all features of Hyper-V.
Hyper-V-administratörer	Medlemmar i den här gruppen har fullständig och obegränsad tillgång till alla funktioner i Hyper-V.
Händelseloggläsare	Medlemmarna i den här gruppen kan läsa händelseloggar från den lokala datorn
IIS_IUSRS	Inbyggd grupp som används av Internet Information Services.
IIS_WPG	IIS Worker Process Group
Network Configuration Operators	Members in this group can have some administrative privileges to manage configuration of network...
Parrots-CCTV	
Performance Log Users	Members of this group may schedule logging of performance counters, enable trace providers, and...
Performance Monitor Users	Members of this group can access performance counter data locally and remotely
Power Users	Power Users are included for backwards compatibility and possess limited administrative powers
Primates-CCTV	
Print Operators	Members can administer printers installed on domain controllers
Privilegierade användare	Privilegierade användare inkluderas för bakåtkompatibilitet och har en del begränsade administrati...
RDS Endpoint Servers	Servers in this group run virtual machines and host sessions where users RemoteApp programs an...
RDS Management Servers	Servers in this group can perform routine administrative actions on servers running Remote Deskto...
RDS Remote Access Servers	Servers in this group enable users of RemoteApp programs and personal virtual desktops access t...
Remote Desktop Users	Members in this group are granted the right to logon remotely
Remote Management Users	Members of this group can access WMI resources over management protocols (such as WS-Man...
Replicator	Supports file replication in a domain
Reptiles-CCTV	

Latest search was: 2016-08-23 11:01:19

Figure 2.387 An example of the dialog "Browser for User groups", the tab Local user groups for an Ethiris Server.

The tab *Local user groups on <Name>* list the local Windows user groups defined on the computer running Ethiris Server.


Browser for User groups
Ethiris user groups
Local user groups in cluster Obelix
Domain user groups on KENTIMA

User group	Comment
Access Control Assistance Opera...	Members of this group can remotely query authorization attributes and permissions for resources on...
Administrators	Administrators have complete and unrestricted access to the computer/domain
Administratörer	Administratörer har fullständig och obegränsad åtkomst till datorn/domänen
AMD FUEL	Members of this group can change AMD platform power controls within Catalyst Control Center to b...
Ansvariga för kryptering	Medlemmar har behörighet att utföra kryptografiåtgärder.
Ansvariga för nätverkskonfigurering	Medlemmar i den här gruppen kan ha vissa administrativa privilegier för att hantera konfiguration fö...
Ansvariga för replikering	Ansvariga för filreplikering i en domän
Ansvariga för säkerhetskopiering	Ansvariga för säkerhetskopiering kan åsidosätta begränsningar för åtgärder som har med säkerhet...
Användare	Användare är förhindrade att göra oavsiktliga eller avsiktliga ändringar av systemet och kan köra d...
Användare av fjärrskrivbord	Medlemmar i den här gruppen har rättighet att logga in på fjärrdatorer
Användare av prestandaloggar	Medlemmar i den här gruppen kan schemalägga loggning av prestandaräknare, aktivera spårnings...
Användare av prestandaövervak...	Medlemmar i den här gruppen kan läsa prestandaräknardata från den här datorn och från fjärrdatorer
Användare som kan ge fjärrhjälp	Medlemmar i den här gruppen kan erbjuda fjärrhjälp till den här datorns användare.
Backup Operators	Backup Operators can override security restrictions for the sole purpose of backing up or restoring ...
Birds-CCTV	
Boats-CCTV	
Carnivores-CCTV	
Catering-CCTV	
Certificate Service DCOM Access	Members of this group are allowed to connect to Certification Authorities in the enterprise
Cryptographic Operators	Members are authorized to perform cryptographic operations.
DCOM-användare	Medlemmar kan starta, aktivera och använda DCOM-objekt på den här datorn

Latest search was: 2016-08-23 11:01:19

Figure 2.388 An example of the dialog "Browser for User groups", the tab Ethiris user groups.

The tab *Local user groups in cluster <Name>* list the local Windows user groups that are defined on the computers that are members of the cluster. Any user groups that are not defined on all members of the cluster are displayed grayed-out and cannot be selected.

The tab *Domain user groups on <Name>* list the user groups available on the domain where the server/cluster members are a member.

2.4.57 Client type Privilege node

Under the node Privilege in the treeview, there is a *Client type Privilege* node.

The purpose of this node is to provide the opportunity to enter specific security settings for different types of clients.

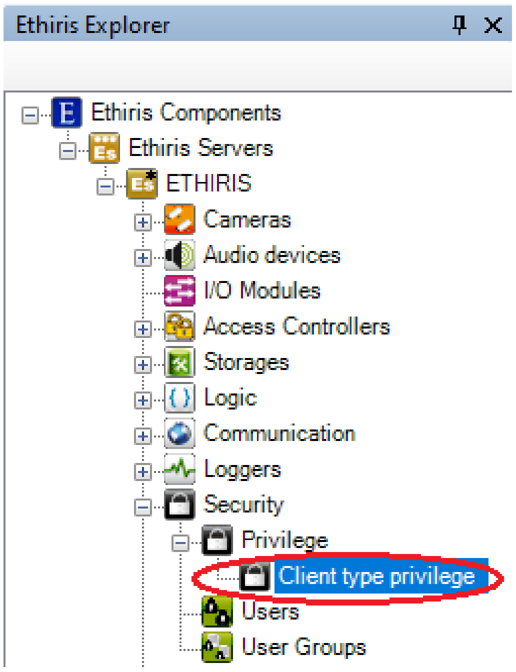


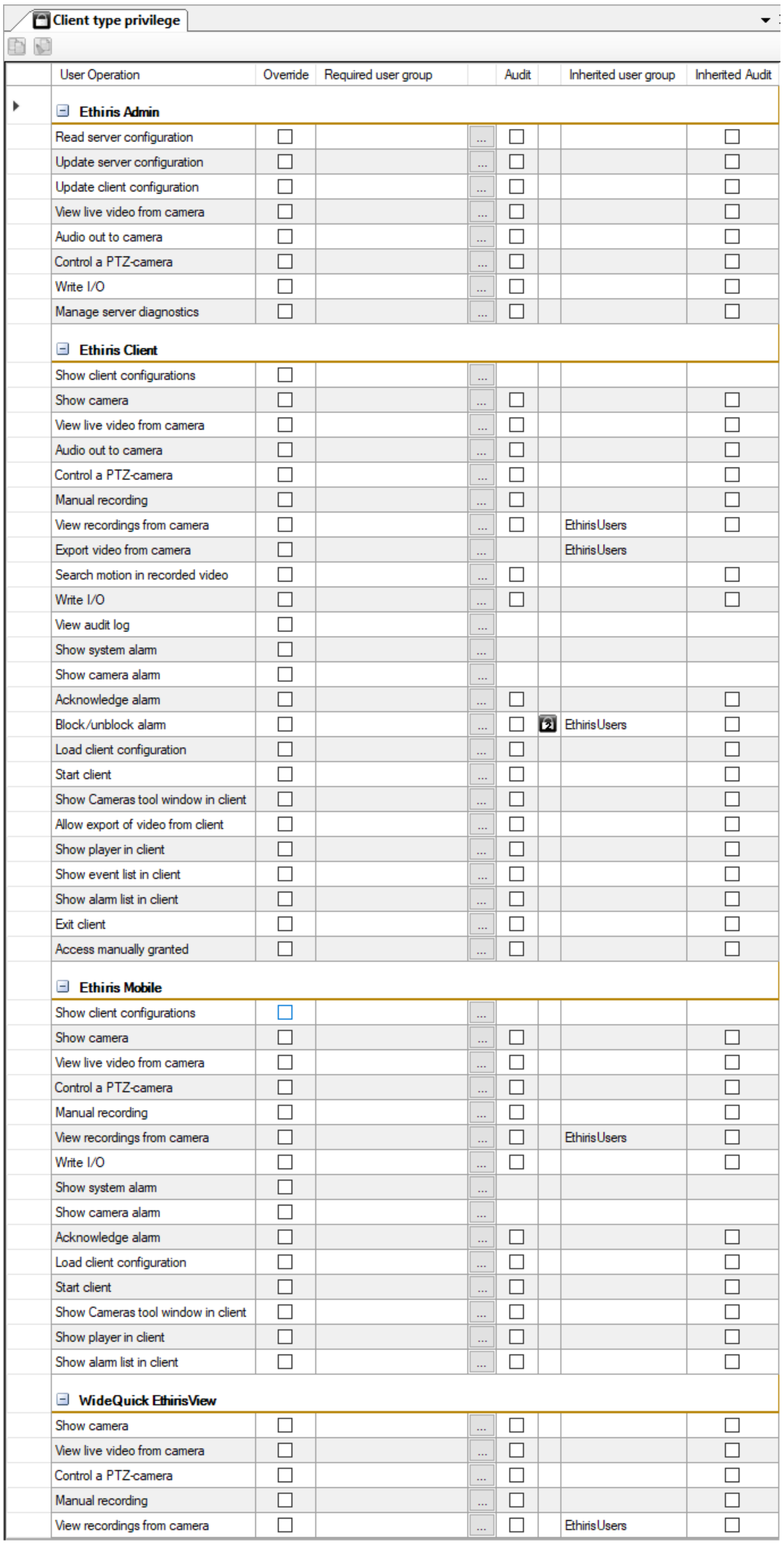
Figure 2.389 The node Client type Security in the Ethiris Explorer treeview.

Client type privilege popup menu

There is no popup menu for this node.

Client type privilege panel

Double-click on the *Client type privilege* node in the treeview will open the corresponding panel.



Client type privilege

User Operation	Override	Required user group		Audit		Inherited user group	Inherited Audit
Ethiris Admin							
Read server configuration	☐		...	☐			☐
Update server configuration	☐		...	☐			☐
Update client configuration	☐		...	☐			☐
View live video from camera	☐		...	☐			☐
Audio out to camera	☐		...	☐			☐
Control a PTZ-camera	☐		...	☐			☐
Write I/O	☐		...	☐			☐
Manage server diagnostics	☐		...	☐			☐
Ethiris Client							
Show client configurations	☐		...				
Show camera	☐		...	☐			☐
View live video from camera	☐		...	☐			☐
Audio out to camera	☐		...	☐			☐
Control a PTZ-camera	☐		...	☐			☐
Manual recording	☐		...	☐			☐
View recordings from camera	☐		...	☐		EthirisUsers	☐
Export video from camera	☐		...			EthirisUsers	
Search motion in recorded video	☐		...	☐			☐
Write I/O	☐		...	☐			☐
View audit log	☐		...				
Show system alarm	☐		...				
Show camera alarm	☐		...				
Acknowledge alarm	☐		...	☐			☐
Block/unblock alarm	☐		...	☐	▣	EthirisUsers	☐
Load client configuration	☐		...	☐			☐
Start client	☐		...	☐			☐
Show Cameras tool window in client	☐		...	☐			☐
Allow export of video from client	☐		...	☐			☐
Show player in client	☐		...	☐			☐
Show event list in client	☐		...	☐			☐
Show alarm list in client	☐		...	☐			☐
Exit client	☐		...	☐			☐
Access manually granted	☐		...	☐			☐
Ethiris Mobile							
Show client configurations	☐		...				
Show camera	☐		...	☐			☐
View live video from camera	☐		...	☐			☐
Control a PTZ-camera	☐		...	☐			☐
Manual recording	☐		...	☐			☐
View recordings from camera	☐		...	☐		EthirisUsers	☐
Write I/O	☐		...	☐			☐
Show system alarm	☐		...				
Show camera alarm	☐		...				
Acknowledge alarm	☐		...	☐			☐
Load client configuration	☐		...	☐			☐
Start client	☐		...	☐			☐
Show Cameras tool window in client	☐		...	☐			☐
Show player in client	☐		...	☐			☐
Show alarm list in client	☐		...	☐			☐
WideQuick EthirisView							
Show camera	☐		...	☐			☐
View live video from camera	☐		...	☐			☐
Control a PTZ-camera	☐		...	☐			☐
Manual recording	☐		...	☐			☐
View recordings from camera	☐		...	☐		EthirisUsers	☐

Figure 2.390 The panel Client type Security.

KENTIMA
Automation and Security Products

In the list, there are just about the same columns and the same operations as for security and some additional columns. The column for *pre-authorization* is missing but is indicated with a small lock if pre-authorization is defined at the Ethiris Server level. Another difference is that the operations are divided between the different types of clients; *Ethiris Admin, Ethiris Client*, *Ethiris Mobile,* and *WideQuick EthirisView*. See the last section for an explanation of the various operations.

For each type of client, the relevant operations are listed.

Override has to be explicitly checked to be able to change the *Required User Group*. A Security setting can be set at the Ethiris Server level, meaning that all types of clients have the same security settings unless they are specifically overridden in this panel by checking *Override*.

Required User Group specifies what user group the user has to be a member of, to access this function. A blank field means that no log in is required. You can browse for available user groups by clicking the browse button to the right of this column.

Audit can be checked if you want to log when a certain operation is executed. For each operation for which *Audit* is specified, the system will save information on the time, the operation performed, who performed it, the client computer from which it was performed, and any other available information, depending on the type of operation performed.

It is permitted to specify that the system must log an operation without, at the same time, making any requirement that the user must be a member of a specific group. The operation will be logged regardless of this, but in these cases, there may be no information on who performed the operation if the user has not logged on. Other available information is logged as usual.

The audit log can be viewed in the *Events* panel in Ethiris Client.

Inherited User Group is just information on the current setting on the Ethiris Server level. To override this setting, check the *Override* checkbox.

Inherited Audit is just information on the current setting on the Ethiris Server level. To override this setting, check the *Override* checkbox

2.4.58 Users node

Under the node *Security* in the treeview, there is a node *Users*.

The purpose of this node is to add user accounts to the system. These user accounts can be used to log in to Ethiris and thereby grant or deny the user privilege for a number of user operations. They are also used to specify recipients of mail and SMS that are sent from the Ethiris Server.

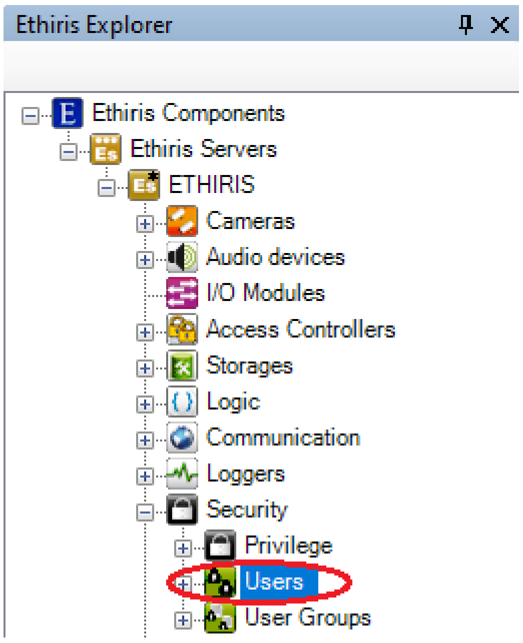


Figure 2.391 The node Usersin EthirisExplorer treeview.

Users popup menu

Right-click on this node opens a menu.

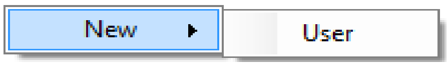
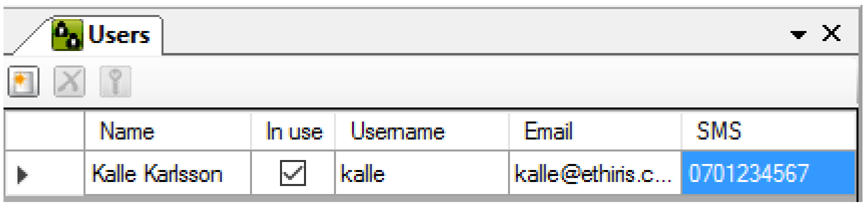


Figure 2.392 Popup menu for the node Users.

New->User will add a new user to the server configuration. It will immediately be available as a new user node in the treeview below the node *Users*. If the panel *Users* is open, it will show up there also..

Users panel

Double click on the node Users in the treeview will open the corresponding panel.



	Name	In use	Username	Email	SMS
▶	Kalle Karlsson	✓	kalle	kalle@ethiris.c...	0701234567

Figure 2.393 The Users panel.

This panel holds a list of all users defined in the server configuration.

At the top of the panel is a toolbar.

User panel toolbar

Figure 2.394 Toolbar in the panel Users.

Add a new user

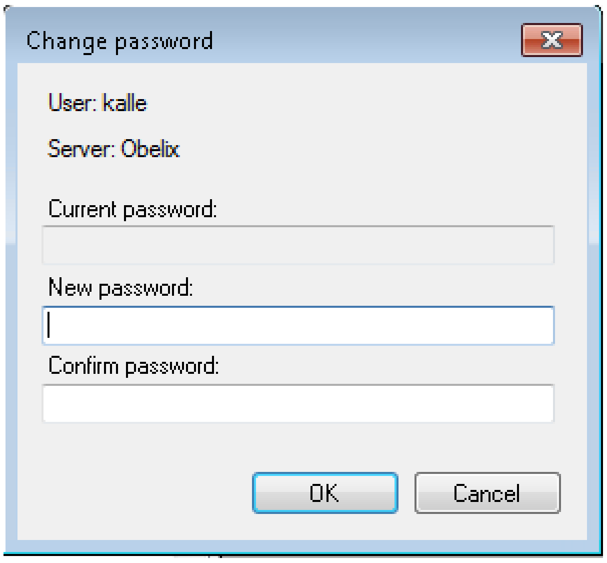
Use this button to create a new user. This is the same as selecting the menu New->User in the popup menu, as described above. A new user is added to the server configuration immediately.

Delete selected user(s)

Use this button to delete users from the configuration. You can select several users by holding down the *Ctrl*- or *Shift* key.

Change password of selected user

Use this button to specify a new password for the selected user.


Change password

User: kalle

Server: Obelix

Current password:

New password:

Confirm password:

Figure 2.395 Dialogen Byt lösenord

If the user exists in the configuration (not being defined now), the new password will be saved immediately when you click OK. If the user is being defined, the password will be saved when the configuration is saved.

Users panel list of users

The user's list has several columns.

Name is the full name of the user. This name must be unique in the configuration since it’s used to identify the user in *Contact lists*, *E-mail messages,* and *SMS-messages*.

In use should be checked if you want the user to be able to log in to the system. When *In Use* is checked for a newly created user, the *Change password* dialog is displayed, and a password must be specified. If no password is specified, the user cannot log in to Ethiris. If *In Use* is unchecked, the user can still be used as a target for E-mail and/or SMS messages. Any existing password is retained.

User name is the name the user enters when logging in to Ethiris.

Email is the e-mail address of the user.

SMS is the mobile phone number to the user when sending SMS.

2.4.59 User node

Under the node *Users* in the treeview, there is a node for each defined user.

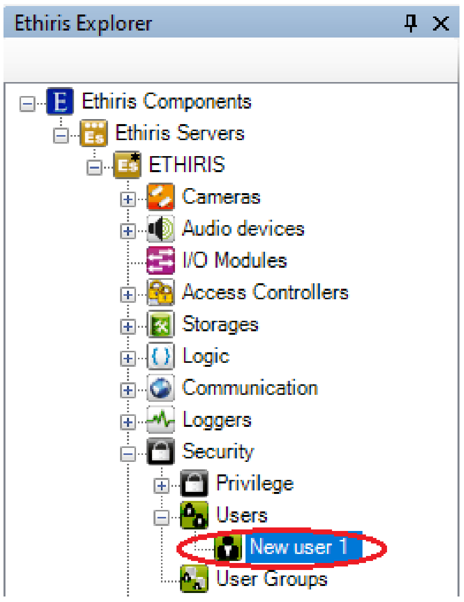


Figure 2.396 New user added.

User popup menu

Right-clicking such a node opens a context menu.

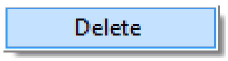


Figure 2.397 The popup menu for a user.

Delete will remove the user's definition from the configuration. The user will be removed immediately both from the treeview and the user's list in the panel *Users.*

User node panel

There is no panel for the user nodes. The panel Users is opened instead on double-click on a user node.

2.4.60 User Groups node

Under the node *Security* in the treeview, there is a node *User Groups*.

This node holds the defined user groups in the system. These are used to define which users have specific privileges. You can, for example, say that a user must be a member of the user group *EthirisAdmins* to be allowed to update the server configuration.

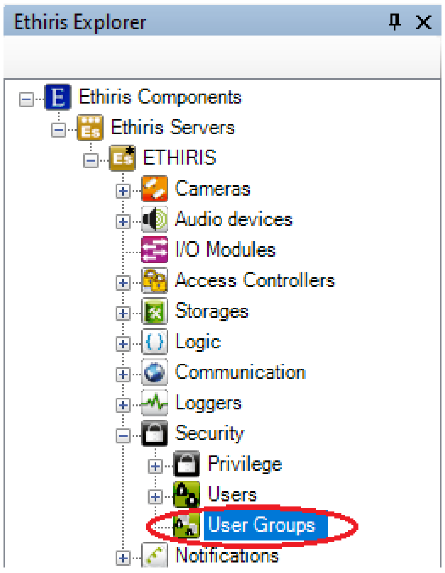


Figure 2.398 The node User Groups in Ethiris Explorer.

User groups popup menu

Right-clicking such a node opens a context menu.

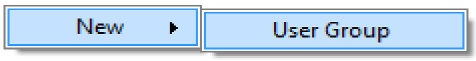


Figure 2.399 The popup menu for the node User groups.

New->User group will add a new user group to the server configuration. It will become immediately available as a new node below the *User Groups* node in the treeview. If the panel *User Groups* is open, it will also be visible there.

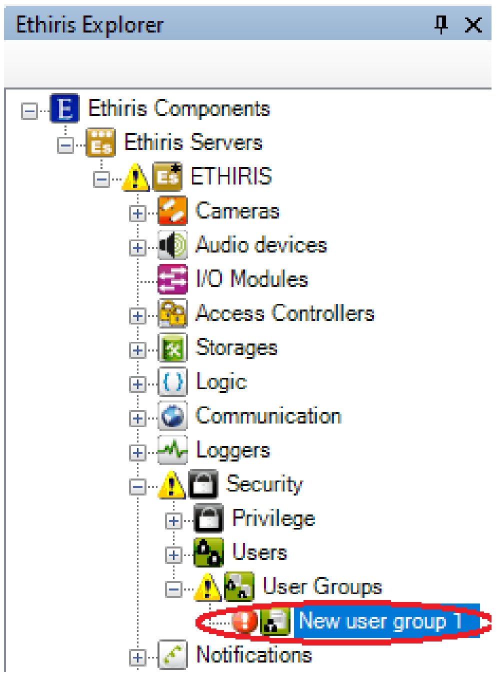
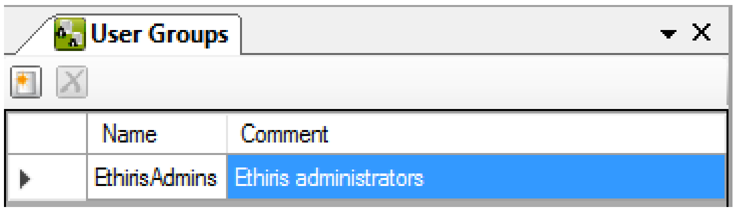


Figure 2.400 New user group added.

The error icon is displayed for a newly added user group because there are no users defined in that group yet.

User Groups panel

Double click on the node *User Groups* in the treeview will open the corresponding panel.



	Name	Comment
▶	EthirisAdmins	Ethiris administrators

Figure 2.401 The panel User group.

This panel comprises a list of all user groups defined in the server configuration.

At the top, there is a toolbar.

User Groups panel toolbar

Figure 2.402 The toolbar in the panel User Groups.

Add a new user group

Use this button to create a new user group. This is the same as choosing the menu alternative New User group in the popup menu as described above. A new user group is added immediately to the server configuration.

Delete selected user group(s)

Use this button to delete the user groups that are selected in the list. You can select more than one user group by holding down the *Ctrl*- or *Shift*-key.

User Groups panel list of user groups

The list of user groups has several columns.

Name is the user group name. This name must be unique in the configuration. It is used to identify the group in *Privilege*.

Comment is exactly what it sounds like, a comment. It’s displayed in the dialog *Browse for user groups*.

2.4.61 User group node

Under the node *User Groups* in the treeview, there is a node for each defined user group.

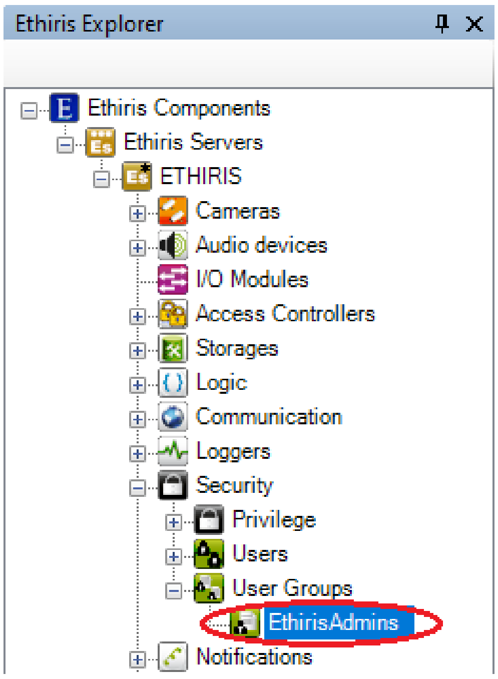


Figure 2.403 A User group in the Ethiris Explorer treeview.

User group popup menu

Right-clicking such a node opens a context menu.

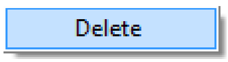
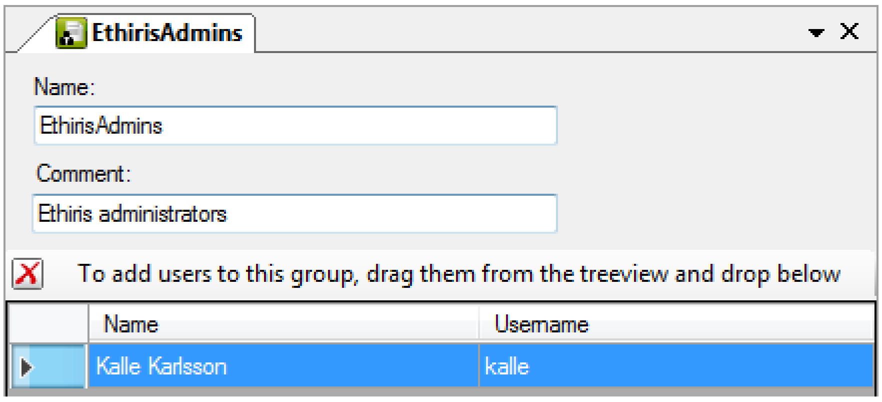


Figure 2.404 The popup menu for a user group.

Delete removes the user group from the server configuration. The user group is deleted immediately both from the treeview and from the list of user groups in the panel *User groups*.

User group panel

Double click on a *User group* node in the treeview opens the corresponding panel.



EthirisAdmins

Name:
EthirisAdmins

Comment:
Ethiris administrators

To add users to this group, drag them from the treeview and drop below

	Name	Username
▶	Kalle Karlsson	kalle

Figure 2.405 The panel for a user group.

This panel holds a list of users that belong to the user group. At the top, there are fields where you can modify the name and comment of the user group.

The panel has the following fields:

Name is the name of the user group. The name is used to identify the user group in the panel *Privilege*.

Comment gives an opportunity to describe the user group. It’s displayed in the dialog *Browse for User groups*.

User group panel toolbar

Figure 2.406 Toolbar in the panel for a User group.

Remove selected user(s) from user group.

Use this button to delete the users that are selected in the list of user group members from the group. You can select more than one user by holding down the *Ctrl-* or *Shift*-key.

User group panel list of users

The list of users that are members of a user group has several columns.

Name is the name of the user.

User name is the user's log-in name.

To add users to a user group, simply drag the user from the treeview and drop them on the list of members. Open the panel for the user group to which you wish to add a member. In the treeview, you click and hold down the left mouse button on the user you wish to add to the user group. Drag the user and release the mouse button when the user is on top of the list of members in the user group.

2.4.62 Notifications node

Under the Ethiris Server node in the treeview, there is a *Notifications* node. This is a container node for *Contacts*, *Mails,* and *SMSs* that are defined in the server.

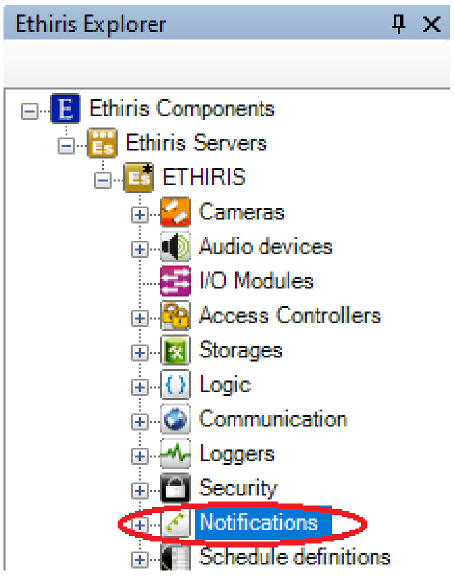


Figure 2.407 The Notifications node in Ethiris Explorer treeview.

The Notifications node is about Email and SMS. This is where you define all the recipients of emails and SMSs. You define the emails and SMSs themselves here as well.

Notifications popup menu

Right-clicking this node brings up a context menu.

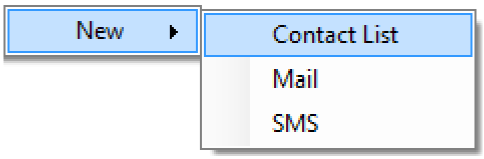


Figure 2.408 The popup menu for the Notifications node.

New->Contact list adds a new contact list to the server configuration. It is immediately visible in the treeview as a new contact list node under the *Contact Lists* node. Should you have opened the *Contact Lists* panel, the new contact list would be added there too.

New->Mail adds a new mail to the server configuration. It is immediately visible in the treeview as a new mail node under the *Mails* node. Should you have opened the *Mails* panel, the new mail would be added there too.

New->SMS adds a new SMS to the server configuration. It is immediately visible in the treeview as a new SMS node under the *SMSs* node. Should you have opened the *SMSs* panel, the new SMS would be added there too.

Notifications panel

There is no panel for the *Notifications* node. Instead, each sub-node has a panel.

2.4.63 Contact Lists node

Under the *Notifications* in the treeview, there is also a *Contact Lists* node. This is a container node for all contact lists that are defined in the server.

The purpose of contact lists is to gather several contacts in a list. The contact list can then be used as a recipient in mails and SMSs.

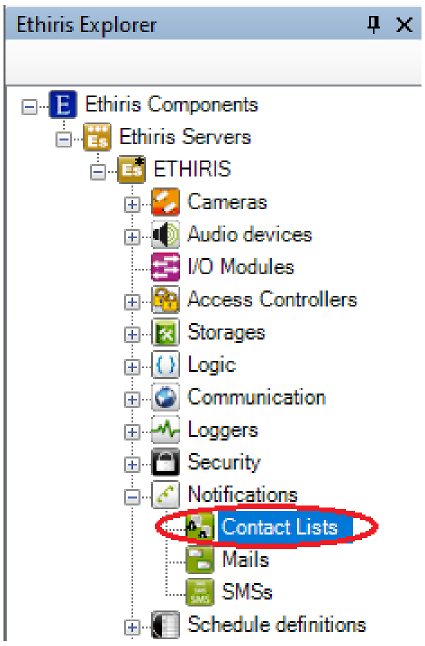


Figure 2.409 The Contact Lists node in Ethiris Explorer treeview.

Contact Lists popup menu

Right-clicking this node brings up a context menu.

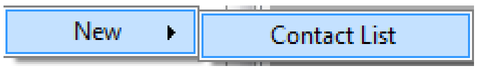
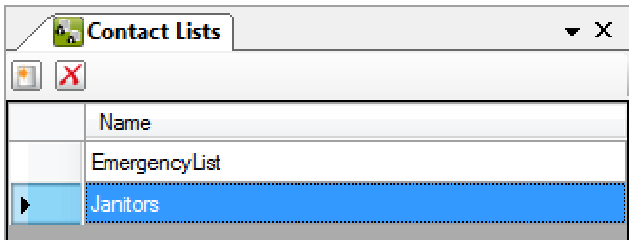


Figure 2.410 The popup menu for the Contacts node.

New->Contact List adds a new contact list to the server configuration. It is the same as the *New->Contact List* in the Notifications popup menu above.

Contact Lists panel

Double-clicking the *Contact Lists* node in the treeview opens the corresponding panel.



Name
EmergencyList
Janitors

Figure 2.411 The Contact Lists panel.

This panel consists of a list of all contact lists currently part of the server configuration.

At the top of the panel, there is a toolbar.

Contact Lists panel toolbar

Figure 2.412 The toolbar in the Contact Lists panel.

Add a new contact list

Use this button to create a new contact list. This is the same as *New->Contact List* in the popup menu for the Notifications node described above. A new contact list is immediately added to the server configuration.

Delete selected contact list(s)

Use this button to delete the selected contact list(s) from the configuration. You can select more than one contact list by using the *Ctrl*-button and/or the *Shift*-button.

Contact Lists panel contact list

The contact list consists of one column only.

Name is the desired name of the contact list. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the contact list in the list indicating the problem. This name is used for identifying the contact list in mails and SMSs.

2.4.64 Contact List node

Under the Contact Lists node in the treeview, there may be some *Contact List* nodes. You have to open the Contact List panel to add contacts to the list.

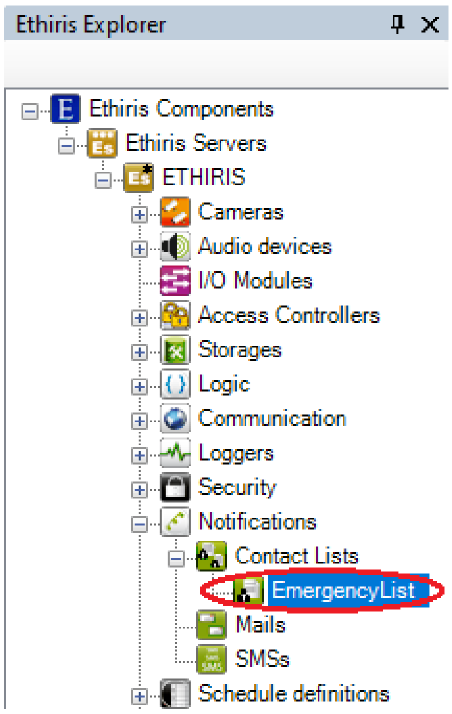


Figure 2.413 A Contact List node in Ethiris Explorer treeview.

Contact List popup menu

Right-clicking this node brings up a context menu.

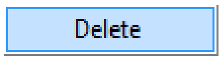
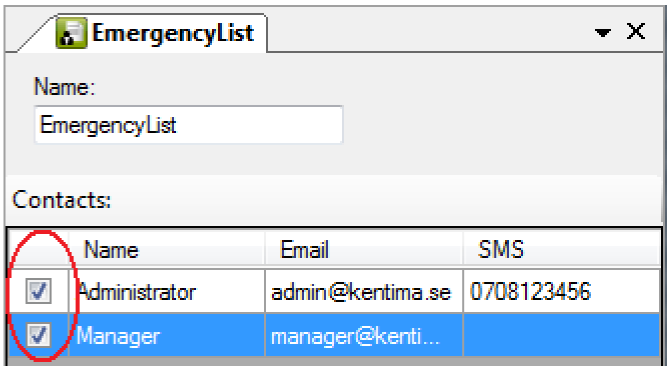


Figure 2.414 The popup menu for a Contact List node.

Delete Removes the contact list from the server configuration. It is immediately removed from both the treeview and the contact list in the *Contact Lists* panel.

Contact List panel

Double-clicking a *Contact List* node in the treeview opens a panel for the specific contact list.



Name:
EmergencyList

Contacts:

	Name	Email	SMS
✓	Administrator	admin@kentima.se	0708123456
✓	Manager	manager@kenti...	

Figure 2.415 The Contact List panel.

In this panel, you can select which users will be members of the list. All available users are listed.

Contacts just check the checkbox for the desired user.

2.4.65 Mails node

Under the *Notifications* in the treeview, there is also a *Mails* node. This is a container node for all mails that are defined in the server.

The purpose of Mails is to create email definitions that later can be sent on various events in the system.

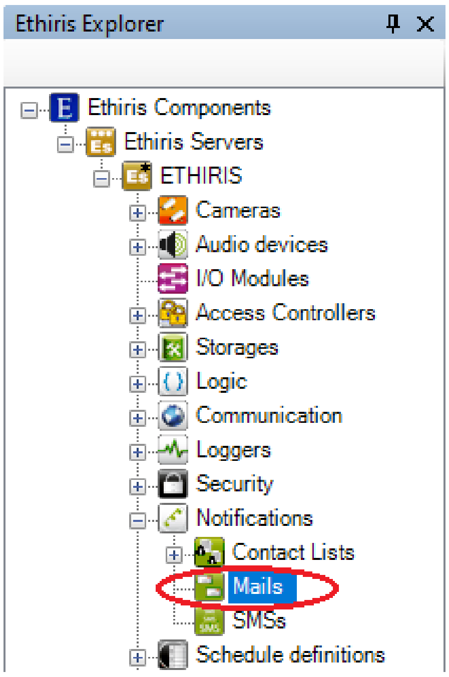


Figure 2.416 The Mails node in Ethiris Explorer treeview.

Mails popup menu

Right-clicking this node brings up a context menu.

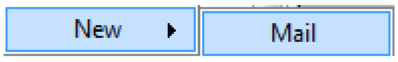
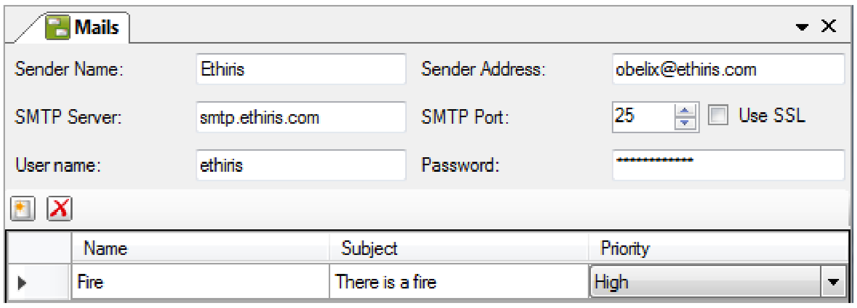


Figure 2.417 The popup menu for the Mails node.

New->Mail adds a new mail to the server configuration. It is the same as the *New->Mail* in the Notifications popup menu above.

Mails panel

Double-clicking the *Mails* node in the treeview opens the corresponding panel.



Mails

Sender Name:	Ethiris	Sender Address:	obelix@ethiris.com
SMTP Server:	smtp.ethiris.com	SMTP Port:	25 ☐ Use SSL
User name:	ethiris	Password:	*************

Name	Subject	Priority
Fire	There is a fire	High

Figure 2.418 The Mails panel.

This panel consists of some general email settings and then a list of all currently defined email messages in the server configuration.

Mails panel General settings

Sender Name is the name indicated as the sender in email messages.

Sender address is the email address indicated as the sender address in email messages.

SMTP Server is the name of the email server that manages outgoing emails.

SMTP Port is normally 25. This is the standard port for SMTP services. Some mail servers may use different ports like 465 och 587. Check which port the mail server is using.

Use SSL should be checked if the SMTP server requires an encrypted connection.

User name is used if the email server requires a login to send an email, you can enter the user name here.

Password is used if the email server requires a login to send an email, you can enter the password here.

The Ethiris mail module supports both *SSL* and *TLS* as well as unencrypted communication.

After the general settings, there is a toolbar.

Mails panel toolbar

Figure 2.419 The toolbar in the Mails panel.

Add a new Mail	Use this button to create a new mail. This is the same as *New->Mail* in the popup menu for the Notifications node described above. A new mail is immediately added to the server configuration.
Delete selected mail(s)	Use this button to delete the selected mail(s) from the configuration. You can select more than one mail by using the *Ctrl*-button and/or the *Shift*-button.

Mails panel mail list

The mail list consists of several columns.

Name is the desired name of the email. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the contact list in the list indicating the problem. This name is used for identifying the mail object in, e.g., the *Variable Browser*.

Subject is used as the subject in the email message.

Priority is used as the priority when sending the email message.

2.4.66 Mail node

Under the Mails node in the treeview, there may be some *Mail* nodes. You have to open the Mail panel to add *Receivers* and possibly *Attachments*.

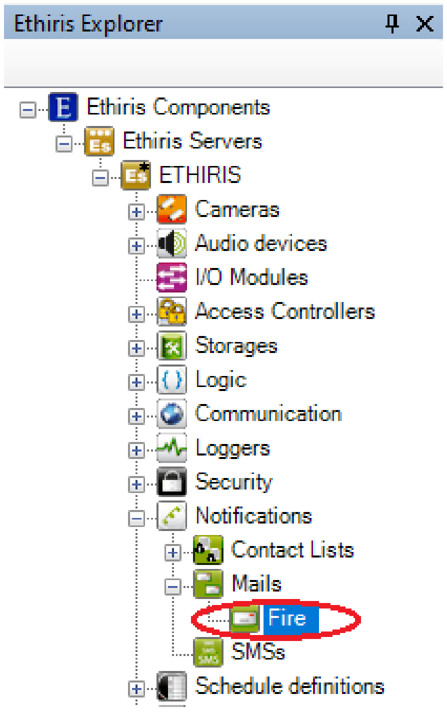


Figure 2.420 A Mail node in Ethiris Explorer treeview.

Mail popup menu

Right-clicking this node brings up a context menu.

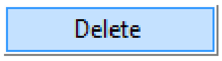
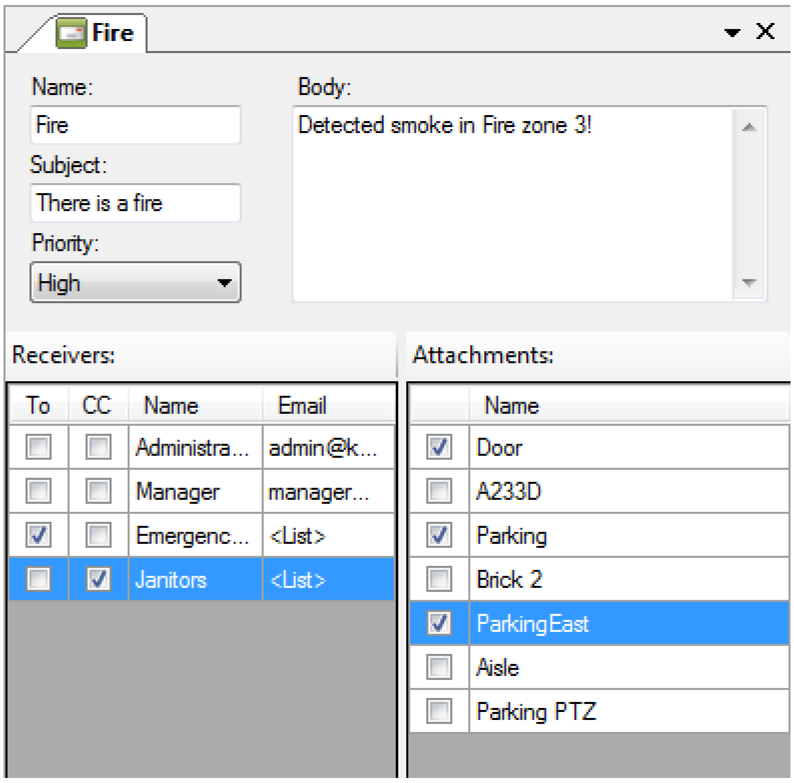


Figure 2.421 The popup menu for a Mail node.

Delete Removes the mail from the server configuration. It is immediately removed from both the treeview and the mail list in the Mails panel.

Mail panel

Double-clicking a *Mail* node in the treeview opens a panel for the specific mail.



Fire

Name: Fire

Subject: There is a fire

Priority: High

Body: Detected smoke in Fire zone 3!

Receivers:

To	CC	Name	Email
☐	☐	Administra...	admin@k...
☐	☐	Manager	manager...
✓	☐	Emergenc...	<List>
☐	✓	Janitors	<List>

Attachments:

	Name
✓	Door
☐	A233D
✓	Parking
☐	Brick 2
✓	ParkingEast
☐	Aisle
☐	Parking PTZ

Figure 2.422 The Mail panel.

In this panel, you can select receivers of the mail. All available contacts and contact lists are listed. You can also select cameras for attaching an image from desired cameras.

Name, **Subject,** and **Priority** are the same as above in the *Mails panel mail list*.

Body is the body of the email.

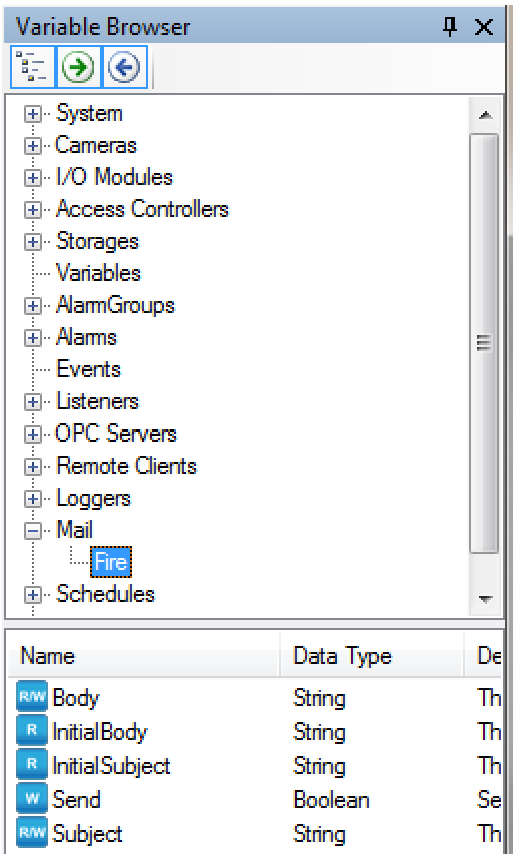
Receivers just check desired checkboxes for *To* and *CC* for selecting receivers.

Attachments just check desired checkboxes for the cameras you want to send an image from in the email.

Mail variables

When defining a *Mail*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The *Variable Browser* panel contains all available variables in the Ethiris Server data store. See *Figure 2.423* for an example where a mail is selected, and the variables belonging to the mail are presented in the lower pane.



Name	Data Type	De
Body	String	Th
InitialBody	String	Th
InitialSubject	String	Th
Send	Boolean	Se
Subject	String	Th

Figure 2.423 Variables for a Mail.

Body is a variable holding the body of the message that will get sent. This means that you can modify the message body from a script before sending it, and so you might not have to create different messages for different purposes.

InitialBody is a read-only variable that holds the message body originally defined in Ethiris Admin. It can be used if you want to reset the message after sending a modified message.

InitialSubject is a read-only variable that holds the message subject originally defined in Ethiris Admin. It can be used if you want to reset the message after sending a modified message.

Send is a writable variable that, when it changes from *false* to *true,* triggers sending of the mail message with the subject and body that are specified at the moment.

Subject is a variable holding the subject of the message that will get sent. This means that you can modify the message body from a script before sending it, and so you might not have to create different messages for different purposes.

2.4.67 SMSs node

Under the *Notifications* in the treeview, there is also an *SMSs* node. This is a container node for all *SMSs* that are defined in the server.

The purpose of SMSs is to create SMS definitions that later can be sent on various events in the system.

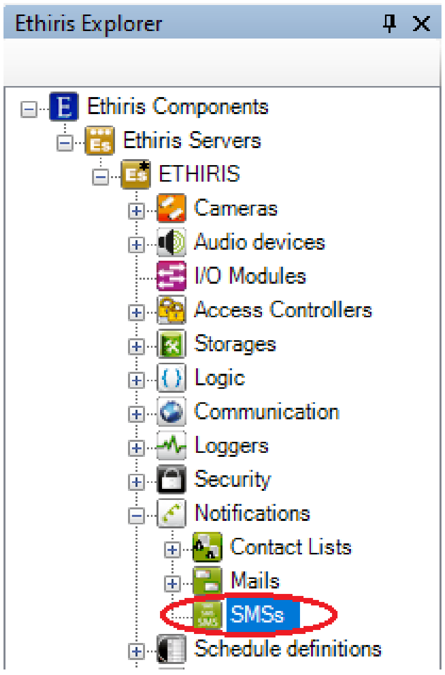


Figure 2.424 The SMSs node in Ethiris Explorer treeview.

SMSs popup menu

Right-clicking this node brings up a context menu.

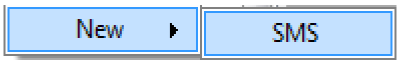
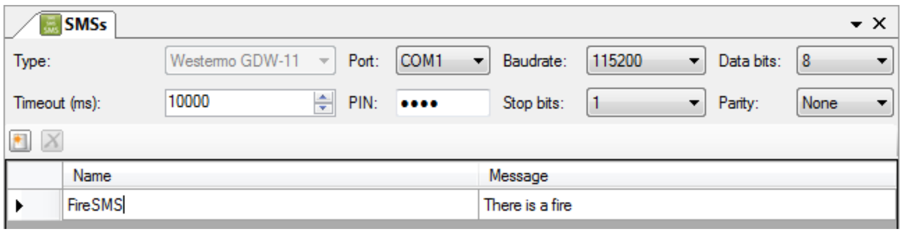


Figure 2.425 The popup menu for the SMSs node.

New->SMS adds a new SMS to the server configuration. It is the same as the *New->SMS* in the *Notifications* popup menu above.

SMSs panel

Double-clicking the *SMSs* node in the treeview opens the corresponding panel.



Type: Westermo GDW-11 Port: COM1 Baudrate: 115200 Data bits: 8
Timeout (ms): 10000 PIN: •••• Stop bits: 1 Parity: None

Name	Message
FireSMS	There is a fire

Figure 2.426 The SMSs panel.

This panel consists of some general SMS settings and then a list of all currently defined SMS messages in the server configuration.

SMSs panel General settings

Type is the type of equipment to use when sending SMSs. For the time being, only *Westermo GDW-11* is supported.

Port indicates the serial port on the computer to which the modem is connected.

Baud rate indicates the communication speed on the serial port.

Data bits are the number of data bits for communication with the modem must be *7* or *8*, depending on the modem settings.

Timeout indicates how long Ethiris waits for a response from the modem before we regard the communication as having failed.

PIN indicates any PIN that the SIM card requires for logon.

Stop bits is the number of stop bits for communication with the modem; *1* or *2*, depending on the modem settings.

Parity indicates the parity check for communication with the modem; *N (None)*, *O (Odd)*, *E (Even)*, *M (Mark),* or *S (Space)*, depending on the modem settings.

After the general settings, there is a toolbar.

SMSs panel toolbar

Figure 2.427 The toolbar in the SMSs panel.

Add a new SMS	Use this button to create a new SMS. This is the same as *New->SMS* in the popup menu for the Notifications node described above. A new SMS is immediately added to the server configuration.
Delete selected SMS(s)	Use this button to delete the selected SMS(s) from the configuration. You can select more than one SMS by using the *Ctrl*-button and/or the *Shift*-button.

SMSs panel SMS list

The SMS list consists of several columns.

Name is the desired name of the SMS. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the SMS in the list indicating the problem. This name is used for identifying the SMS object in, e.g., the *Variable Browser*.

Message is the text in the SMS that is sent.

2.4.68 SMS node

Under the SMSs node in the treeview, there may be some *SMS* nodes. You have to open the SMS's panel to add *Receivers*.

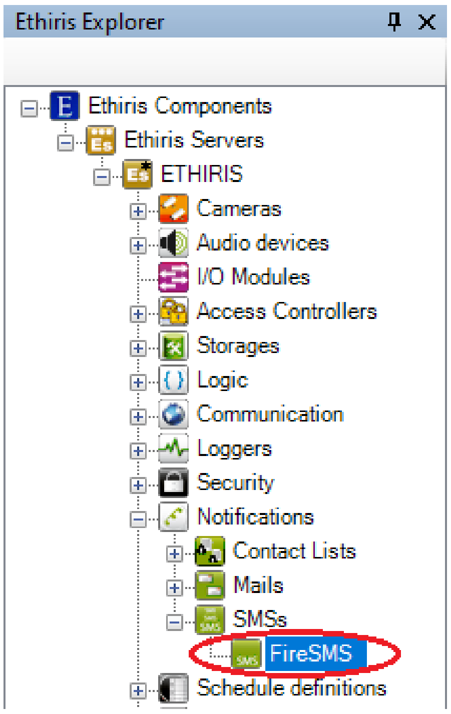


Figure 2.428 A SMS node in the Ethiris Explorer treeview.

SMS popup menu

Right-clicking this node brings up a context menu.

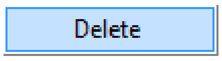
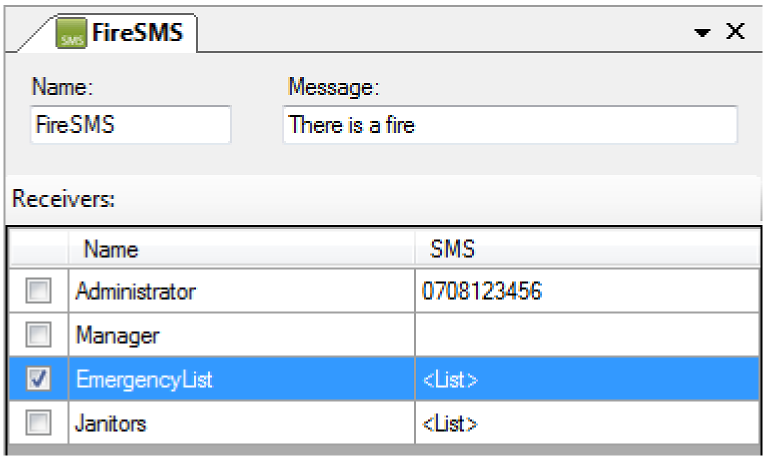


Figure 2.429 The popup menu for an SMS node.

Delete Removes the SMS from the server configuration. It is immediately removed from both the treeview and the SMS list in the SMSs panel.

SMS panel

Double-clicking an *SMS* node in the treeview opens a panel for the specific SMS.



FireSMS

Name: FireSMS

Message: There is a fire

Receivers:

	Name	SMS
☐	Administrator	0708123456
☐	Manager	
✓	EmergencyList	<List>
☐	Janitors	<List>

Figure 2.430 The SMS panel.

In this panel, you can select receivers of the SMS. All available contacts and contact lists are listed.

Name and **Message** are the same as above in the *SMSs panel SMS list*.

Receivers just check desired checkboxes for selecting receivers.

2.4.69 Schedule definitions node

Under the Ethiris Server node in the treeview, there is a *Schedule definitions* node. This is a container node for *Templates*, *Schedules,* and *Deviations* that are defined in the server.

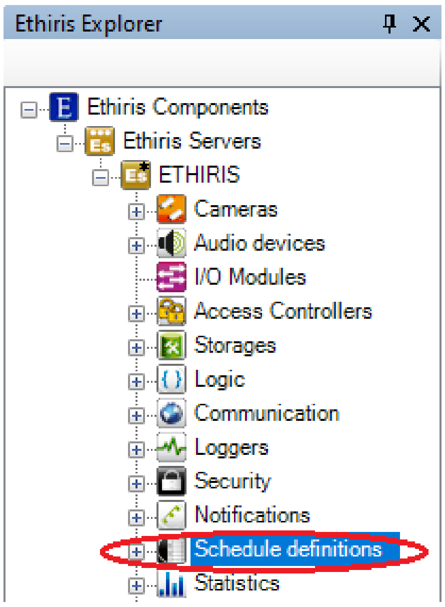


Figure 2.431 The Schedule definitions node in Ethiris Explorer treeview.

The Schedule definitions node is about schedules. This is where you define all schedule templates, schedules, and possibly deviations to the schedules.

Schedules are used in logical expressions in scripts to define the times at which various functions in Ethiris Server are to be active. For example, they can be used to define the times at which it is permitted to retrieve and display frames from connected cameras or to define the times at which it is permitted to store frames in connection with events.

Schedules consist of two parts, Schedule templates and Schedules. A schedule template defines a set of on and off times over a period of 24 hours or a week. A schedule is based on a schedule template together with a (possibly empty) list of deviations for certain weeks or dates. In this way, for example, you can create a schedule that, for most days, is based on the ”Normal” weekly schedule except for week 15, in which instead you use the ”Easter” weekly schedule and 24/12 when you use the ”Christmas Eve” daily schedule.

Each schedule template defines several on and off times during a week or 24 hours. In principle, you can define any number of schedule templates. As you only define the on and off times, the time up to the first on/off in a schedule is of unspecific status. Therefore, you can define the initial status of the schedule template.

In the configuration tool, a weekly schedule is displayed as 7 fields, one for each day of the week, while a daily schedule consists of one field. For each day, the times at which the schedule is active are indicated in green, and the times at which the schedule is inactive are indicated in grey. If the first on/off does not occur at 00.00 on Monday, Monday begins with the state determined by the initial status defined. For each day, you can define any number of different on and off times. On times are indicated with green markers, while off times are indicated with red markers.

The on and off times can be changed by dragging the marker with the left mouse button. For more precise adjustment of the on-time, you can click the marker with the right mouse button and select ”Set time…” from the popup menu. To remove an on or off time, you can click it with the right mouse button and select ”Remove transition” from the menu.

To add new on or off times, you can click with the right mouse button in an active or inactive period in the diagram. If you click on an inactive period, you get a menu with items including ”Add a new active period within day” and ”Add a new active period”. Both options add an on time and off time. The difference between the two items is that ”Add a new active period” spreads the range of the new period over the period in which you clicked even if this should extend over several days, while ”Add a new active period within day” is limited to the period in the day in which you clicked. If instead, you click with the right mouse button in an active period, you get a menu with items including ”Add a new inactive period within day” and ”Add a new inactive period”. Regardless of the period in which you click with the right mouse button, the menu also contains the ”Add transition ON” and ”Add transition OFF” items to add individual on and off times.

You can also use the *Copy* and *Paste* tool buttons to copy the transitions from one day to one or several other days.

Schedule definitions popup menu

Right-clicking this node brings up a context menu.

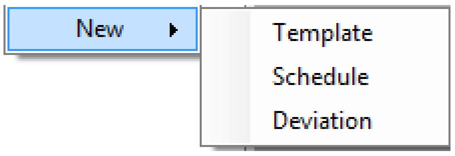


Figure 2.432 The popup menu for the Schedule definitions node.

New->Template adds a new template to the server configuration. It is immediately visible in the treeview as a new template node under the *Templates* node. Should you have opened the *Templates* panel, the new template would be added there too.

New->Schedule adds a new schedule list to the server configuration. It is immediately visible in the treeview as a new schedule node under the *Schedules* node. Should you have opened the *Schedules* panel, the new schedule would be added there too.

New->Deviation adds a new deviation to the server configuration. It is immediately visible in the treeview as a new deviation node under the *Deviations* node. Should you have opened the *Deviations* panel, the new deviation would be added there too.

Schedule definitions panel

There is no panel for the *Schedule definitions* node. Instead, each sub-node has a panel.

2.4.70 Templates node

Under the *Schedule definitions* node in the treeview, there is a *Templates* node. This is a container node for all schedule templates that are defined in the server.

The purpose of schedule templates is to create template definitions that specify a number of on and off times over a period of 24 hours or a week. One template can be used in one or more schedules.

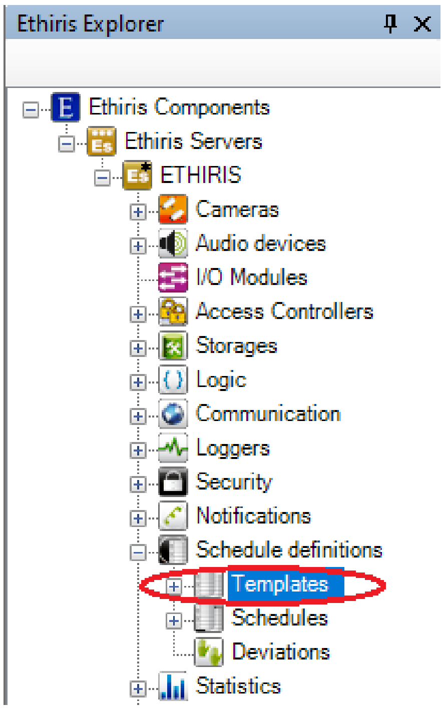


Figure 2.433 The Templates node in Ethiris Explorer treeview.

Templates popup menu

Right-clicking this node brings up a context menu.

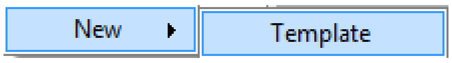
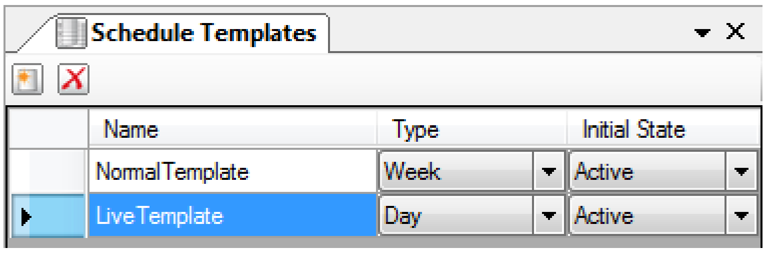


Figure 2.434 The popup menu for the Templates node.

New->Template adds a new schedule template to the server configuration. It is the same as the *New->Template* in the Schedule definitions popup menu above.

Templates panel

Double-clicking the *Templates* node in the treeview opens the corresponding panel.



Name	Type	Initial State
NormalTemplate	Week	Active
LiveTemplate	Day	Active

Figure 2.435 The Templates panel.

This panel consists of a list of all currently defined schedule templates in the server configuration.

At the top of the panel, there is a toolbar.

Templates panel toolbar

Figure 2.436 The toolbar in the Templates panel.

Add a new template	Use this button to create a new schedule template. This is the same as *New->Template* in the popup menu for the Schedule definitions node described above. A new template is immediately added to the server configuration.
Delete selected template(s)	Use this button to delete the selected template(s) from the configuration. You can select more than one template by using the *Ctrl*-button and/or the *Shift*-button.

Templates panel template list

The template list consists of several columns.

Name is the desired name of the template. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the template in the list indicating the problem. This name is used for identifying the template in e g schedules.

Type specifies if the schedule covers a whole week or a single day.

Initial state defines if the schedule should be active or inactive during the period before the first defined transition.

2.4.71 Template node

Under the **Templates** node in the treeview, there may be some *Template* nodes. You have to open the Template's panel to define the *on/off* times by adding *transitions*.

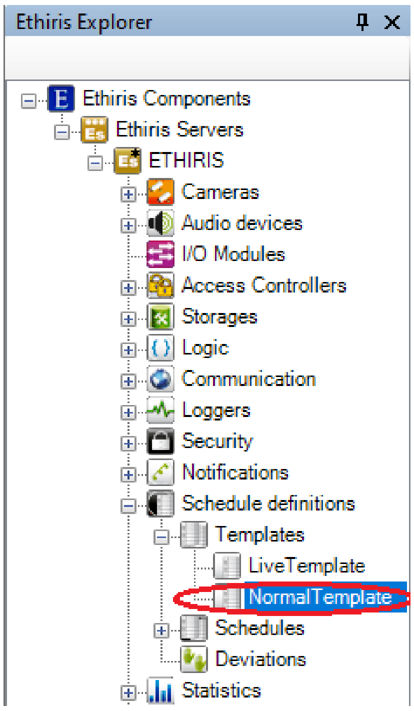


Figure 2.437 A Template node in Ethiris Explorer treeview.

Template popup menu

Right-clicking this node brings up a context menu.

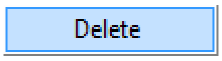


Figure 2.438 The popup menu for a Template node.

Delete Removes the template from the server configuration. It is immediately removed from both the treeview and the template list in the Templates panel. Note that you can't delete the template if it is used in a schedule.

Template panel

Double-clicking a *Template* node in the treeview opens a panel for the specific template.

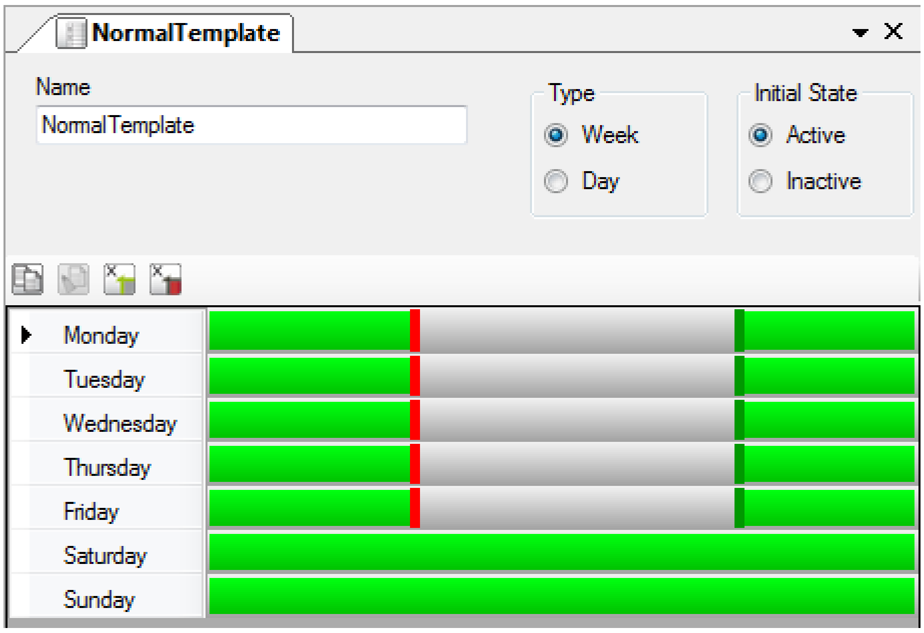


Figure 2.439 The Template panel.

In this panel, you define the on and off times for the schedule template.

Name, **Type,** and **Initial state** are the same as above in the *Templates panel template list*.

After these settings, there is a toolbar.

Template panel toolbar

Figure 2.440 The toolbar in the Template panel.

Copy selected day	Use this button to copy the content of the selected day. The selected day is indicated with an arrow to the far left. In the example above, *Monday* is selected. Click on the left column (with the name of the day) to select a day.
Paste	Use this button to paste the copied content into the currently selected day.
Clear selected day	Use this button to remove all transitions for the currently selected day. This means that the last state from the previous day is used for the whole day.
Clear all days	Use this button to remove all transitions for all days. This means that all days will have the *Initial state*.

Monday - Sunday represents the times from 00:00 to 23:59 during one day of the week. Green fields represent active periods, and gray fields represent inactive periods.

Template panel popup menus

There are three different popup menus in the Templates panel. Depending on where you right-click, different menus popup.

Right-clicking in a day field during an inactive period (white field) displays the following popup menu.

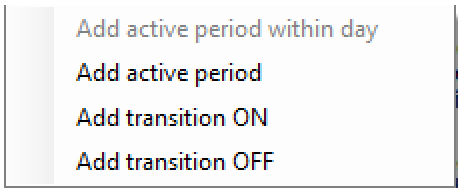


Figure 2.441 The popup menu for an inactive day period.

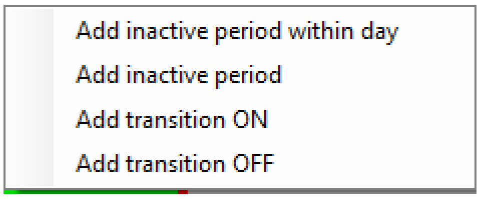


Figure 2.442 The popup menu for an active day period.

Add active period within day adds an on time and off time so that they create a new active period. The times for these are distributed evenly during the period in which you clicked during the day selected.

Add active period adds an on time and off time so that they create a new active period. The times for these are distributed evenly during the period in which you clicked (even if it extends over several days).

Add inactive period within day adds an off time and an on time so that they create a new inactive period. The times for these are distributed evenly during the period in which you clicked during the day selected.

Add inactive period adds an off time and an on time so that they create a new inactive period. The times for these are distributed evenly during the period in which you clicked (even if it extends over several days).

Add transition ON adds an *on* time at the point at which you clicked.

Add transition OFF adds an *off* time at the point at which you clicked.

If you right-click a transition, this popup menu is displayed:

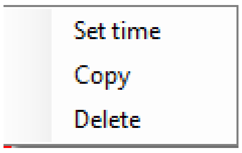


Figure 2.443 The popup menu for a transition.

Set time opens a dialog where you can set the transition time on a specific minute.

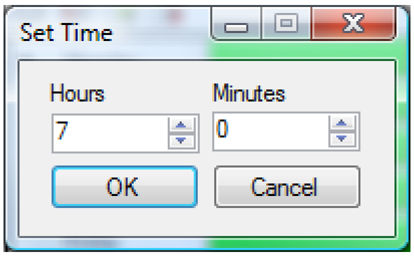


Figure 2.444 The Set time dialog for a transition.

Copy copies the transition, which you then can paste to another day by clicking the *Paste* tool button.

Delete deletes the transition.

2.4.72 Schedules node

Under the Schedule definitions node in the treeview, there is also a *Schedules* node. This is a container node for all schedules that are defined in the server.

The purpose of schedules is to create schedules using the schedule templates you have defined earlier. A schedule can then be used in various contexts for controlling when, e.g., recording of video should take place.

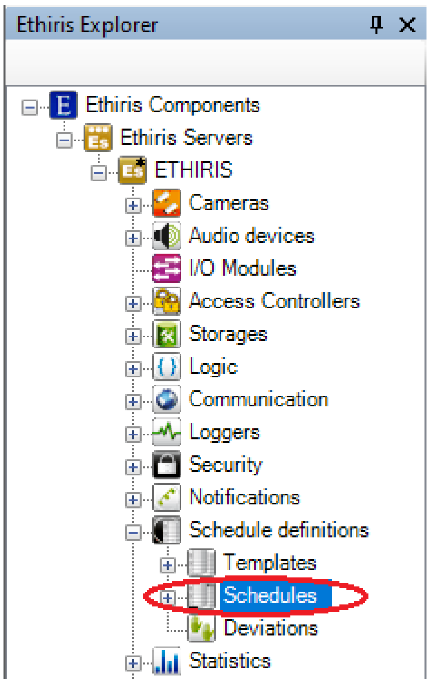


Figure 2.445 The Schedules node in Ethiris Explorer treeview.

Schedules popup menu

Right-clicking this node brings up a context menu.

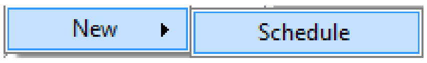
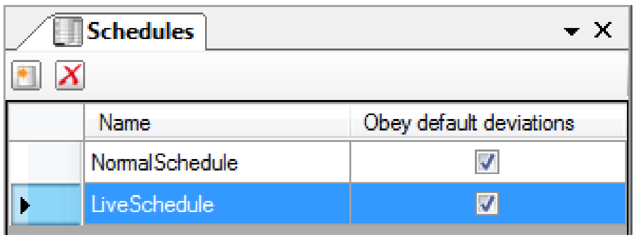


Figure 2.446 The popup menu for the Schedules node.

New->Schedule adds a new schedule to the server configuration. It is the same as the *New->Schedule* in the Schedule definitions popup menu above.

Schedules panel

Double-clicking the *Schedules* node in the treeview opens the corresponding panel.



	Name	Obey default deviations
	NormalSchedule	✓
▶	LiveSchedule	✓

Figure 2.447 The Schedules panel.

This panel consists of a list of all currently defined schedules in the server configuration.

At the top of the panel, there is a toolbar.

Schedules panel toolbar

Figure 2.448 The toolbar in the Schedules panel.

Add a new schedule	Use this button to create a new schedule. This is the same as *New->Schedule* in the popup menu for the Schedules node described above. A new schedule is immediately added to the server configuration.
Delete selected schedule(s)	Use this button to delete the selected schedule(s) from the configuration. You can select more than one schedule by using the *Ctrl*-button and/or the *Shift*-button.

Schedules panel schedule list

The schedule list consists of several columns.

Name is the desired name of the schedule. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the template in the list indicating the problem. This name is used for identifying the schedule in, e.g., script.

Obey default deviations shall be checked if this schedule shall take account of the global deviations defined under *Schedule definitions/Deviations* in the treeview in addition to the deviations defined locally for the schedule.

2.4.73 Schedule node

Under the *Schedules* node in the treeview, there may be some *Schedule* nodes. You have to open the Schedule's panel to define which *template* to use.

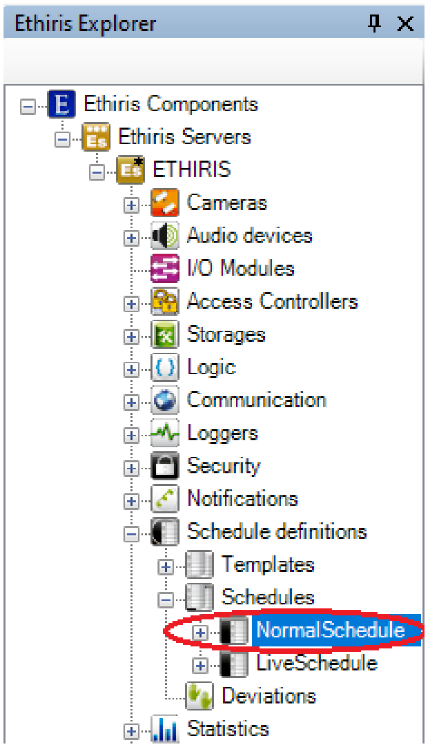


Figure 2.449 A Schedule node in Ethiris Explorer treeview.

Schedule popup menu

Right-clicking this node brings up a context menu.

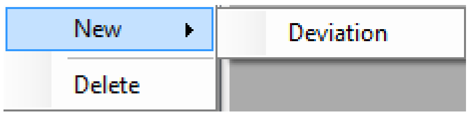


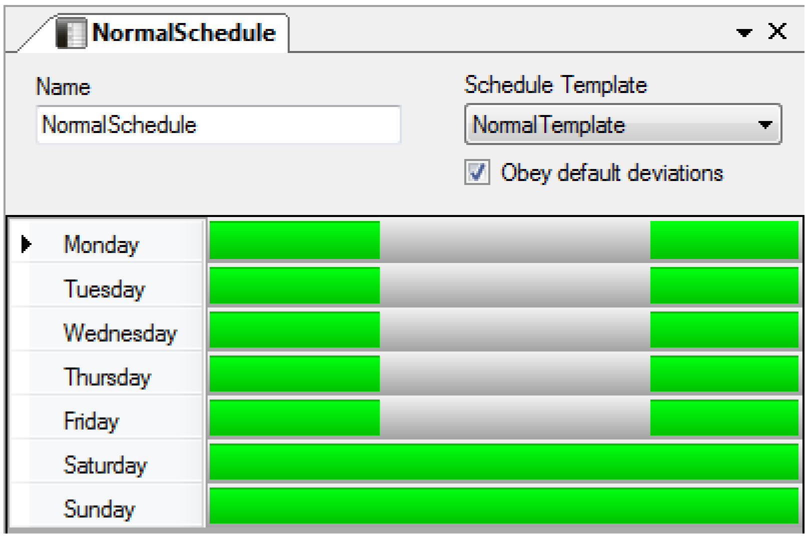
Figure 2.450 The popup menu for a Schedule node.

New->Deviation adds a *deviation* to this specific schedule. It is immediately added to both the treeview and the *Deviations* panel under the schedule.

Delete Removes the schedule from the server configuration. It is immediately removed from both the treeview and the schedule list in the Schedules panel.

Schedule panel

Double-clicking a *Schedule* node in the treeview opens a panel for the specific schedule.


NormalSchedule

Name: NormalSchedule

Schedule Template: NormalTemplate

Obey default deviations ✓

Figure 2.451 The Schedule panel.

In this panel, you define which schedule template to use for the schedule.

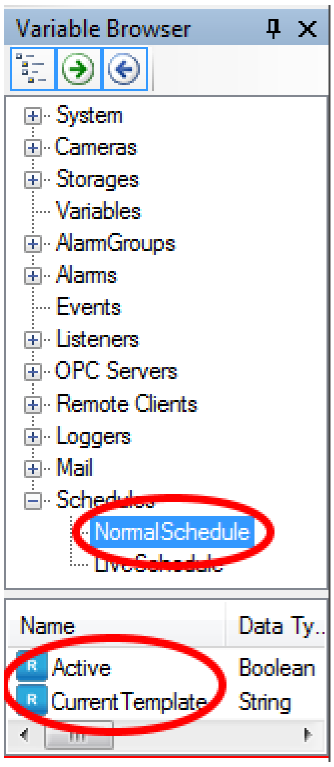
Name and **Obey default deviations** are the same as above in the *Schedules panel schedule list*.

Schedule template displays a list with all available schedule templates that have been defined in the server. Choose the one to use for this schedule.

Schedule variables

When defining a *schedule*, some *variables* are automatically created that you can use in a whole lot of ways, e.g. in a script, send via OPC to other systems, or presenting the information in Ethiris Client.

When you open the *Script* panel in Ethiris Admin, there is a corresponding *Variable Browser* tool window that is docked to the right of the main frame. The Variable Browser panel contains all available variables in the Ethiris Server data store. See *Figure 2.452* for an example where a *schedule* is selected, and the variables belonging to the schedule are presented in the lower pane (ringed in).



Name	Data Ty...
Active	Boolean
CurrentTemplate	String

Figure 2.452 Variables for a schedule.

Active is a read-only variable, that is *true* if the schedule is active (green color) for the moment. This is by far the most used variable for a schedule.

CurrentTemplate is a read-only string variable that contains the name of the schedule template used.

2.4.74 Deviations node

Under a *Schedule* node in the treeview, there is a *Deviations* node. This is a container node for all deviations defined for this schedule.

The purpose of deviations is to be able to make exceptions from the normal schedule. We can define, e.g., that on Christmas, we follow this or that template instead of the one we have defined in the schedule.

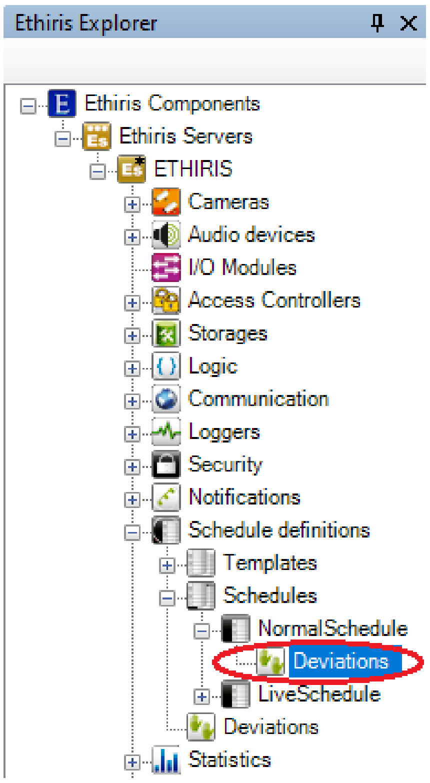


Figure 2.453 The Deviations node in Ethiris Explorer treeview.

Deviations popup menu

Right-clicking this node brings up a context menu.

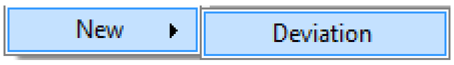


Figure 2.454 The popup menu for the Schedule deviations node.

New->Deviation adds a new deviation to the schedule. It is the same as the *New->Deviation* in the Schedule popup menu above.

Deviations panel

Double-clicking the *Deviations* node in the treeview opens the corresponding panel.

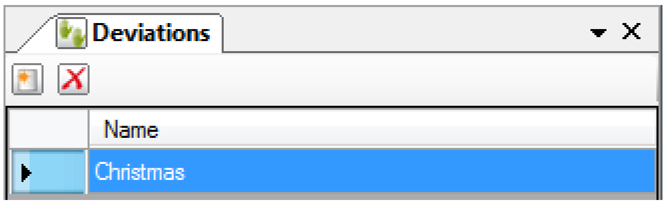


Figure 2.455 The Schedule deviations panel.

This panel consists of a list of all currently defined deviations for this schedule.

At the top of the panel, there is a toolbar.

Deviations panel toolbar

Figure 2.456 The toolbar in the Deviations panel.

Add a new deviation	Use this button to create a new deviation for the schedule. This is the same as *New->Deviation* in the popup menu for the Schedule node described above. A new deviation is immediately added to the server configuration.
Delete selected deviation(s)	Use this button to delete the selected deviation(s) from the configuration. You can select more than one deviation by using the *Ctrl*-button and/or the *Shift*-button.

Deviations panel deviation list

The deviation list consists of only one column.

Name is the desired name of the deviation. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the deviation in the list indicating the problem.

2.4.75 Deviation node

Under the Deviations node in the treeview, there may be some *Deviation* nodes. You have to open the Deviation's panel to define how and when the deviation will be active.

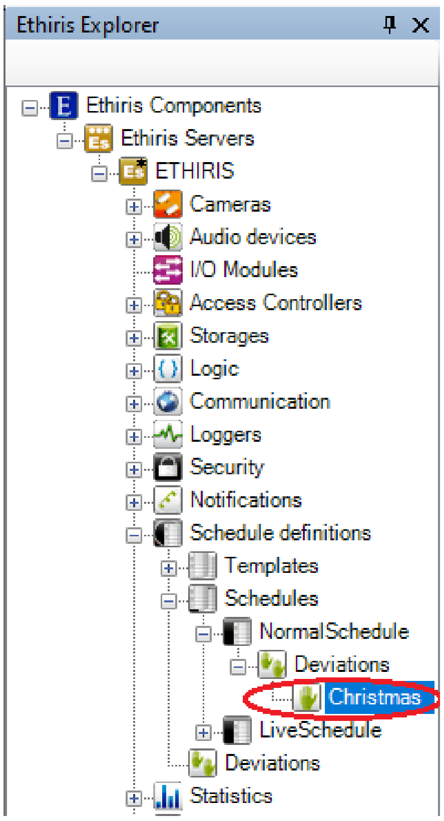


Figure 2.457 A Deviation node in Ethiris Explorer treeview.

Deviation popup menu

Right-clicking this node brings up a context menu.

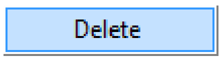
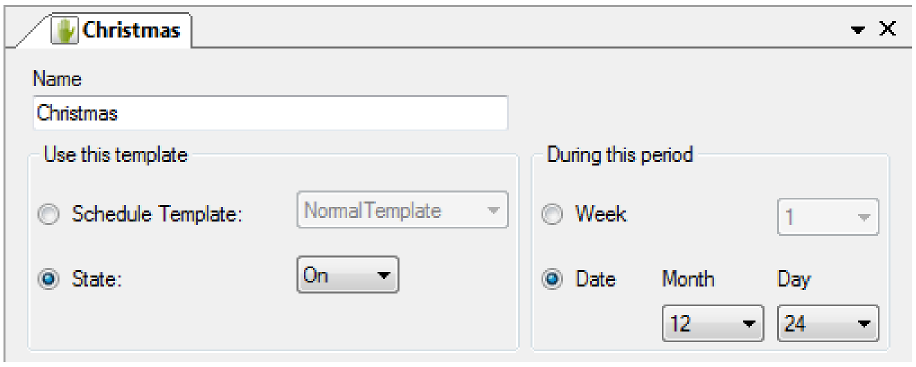


Figure 2.458 The popup menu for a Schedule deviation node.

Delete Removes the deviation from the schedule in the server configuration. It is immediately removed from both the treeview and the deviation list in the Schedule deviations panel.

Deviation panel

Double-clicking a *Schedule deviation* node in the treeview opens a panel for the specific deviation.


Christmas

Name: Christmas

Use this template
- Schedule Template: NormalTemplate
- [✓ State:] On

During this period
- Week: 1
- [✓ Date] Month: 12 Day: 24

Figure 2.459 The Deviation panel.

In this panel, you define how and when the deviation will be active.

Name is the same as above in the *Deviations panel deviation list*.

Use this template You can either use a schedule template, or you can set a state *on* or *off* for the deviation.

During this period You can either select a specific week or date.

2.4.76 Deviations node

Under the Schedule definitions node in the treeview, there is also a *Deviations* node. This is a container node for all default deviations that are defined in the server.

The purpose of default deviations is to be able to define exceptions from the normal schedules that can be used by all schedules in the system.

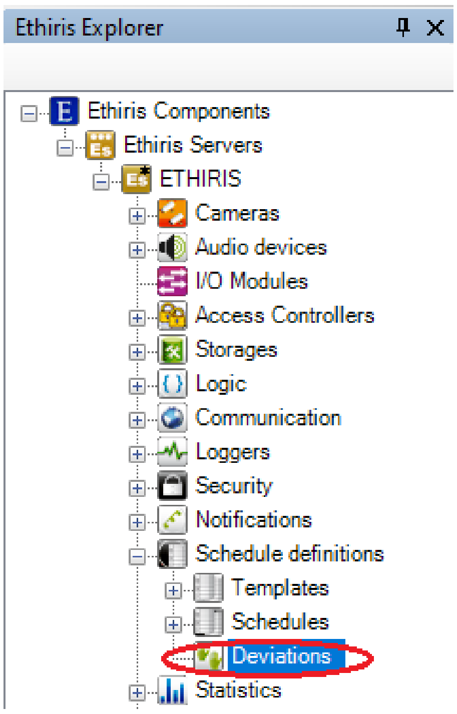


Figure 2.460 The Deviations node in Ethiris Explorer treeview.

Deviations popup menu

Right-clicking this node brings up a context menu.

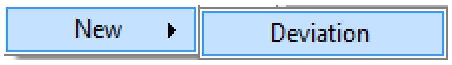
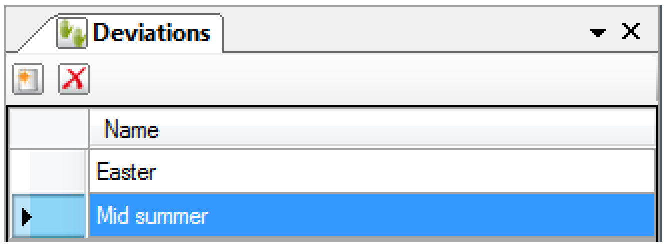


Figure 2.461 The popup menu for the Deviations node.

New->Deviation adds a new default deviation to the server configuration. It is the same as the *New->Deviation* in the Schedule definitions popup menu above.

Deviations panel

Double-clicking the *Deviations* node in the treeview opens the corresponding panel.



	Name
	Easter
▶	Mid summer

Figure 2.462 The Deviations panel.

This panel consists of a list of all currently defined default deviations in the server configuration.

At the top of the panel, there is a toolbar.

Deviations panel toolbar

Figure 2.463 The toolbar in the Deviations panel.

Add a new deviation

Use this button to create a new default deviation. This is the same as *New->Deviation* in the popup menu for the Deviations node described above. A new deviation is immediately added to the server configuration.

Delete selected deviation(s)

Use this button to delete the selected deviation(s) from the configuration. You can select more than one deviation by using the *Ctrl*-button and/or the *Shift*-button.

Deviations panel deviation list

The deviation list consists of one column only.

Name is the desired name of the deviation. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the deviation in the list indicating the problem.

2.4.77 Deviation node

Under the Deviations node in the treeview, there may be some *Deviation* nodes. You have to open the Deviation's panel to define how and when the deviation will be active.

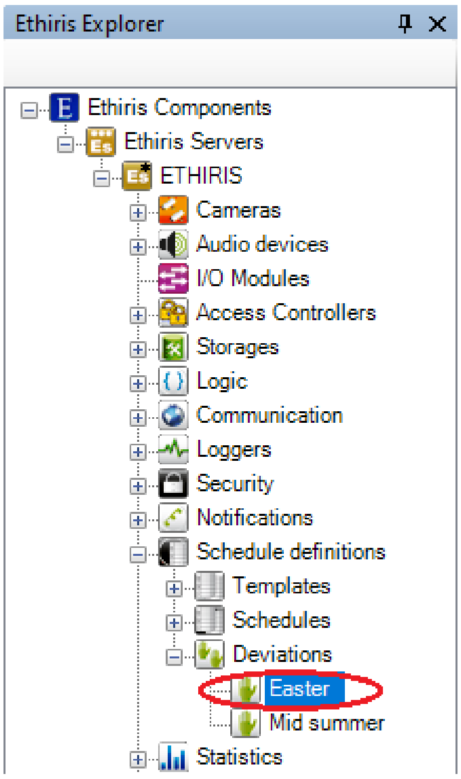


Figure 2.464 A Deviation node in Ethiris Explorer treeview.

Template popup menu

Right-clicking this node brings up a context menu.

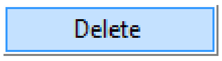
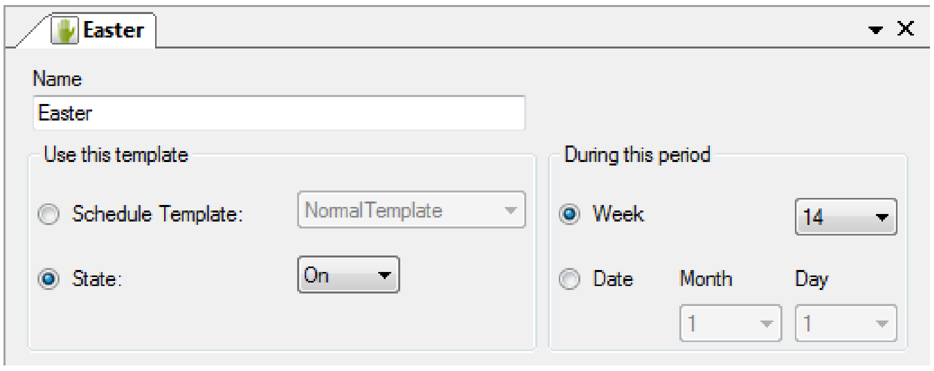


Figure 2.465 The popup menu for a Deviation node.

Delete Removes the deviation from the server configuration. It is immediately removed from both the treeview and the deviation list in the Deviations panel.

Deviation panel

Double-clicking a *Deviation* node in the treeview opens a panel for the specific deviation.



Name: Easter

Schedule Template: NormalTemplate

State: On [✓]

Week [✓] 14

Date: 1 1

Figure 2.466 The Deviation panel.

In this panel, you define how and when the deviation will be active.

Name is the same as above in the *Deviations panel deviation list*.

Use this template You can either use a schedule template, or you can set a state *on* or *off* for the deviation.

During this period You can either select a specific week or a specific date.

2.4.78 Statistics node

Under each Ethiris Server in the treeview, there is a *Statistics* node. This is a collection node for various types of statistics. As of today, there are statistics for cameras.

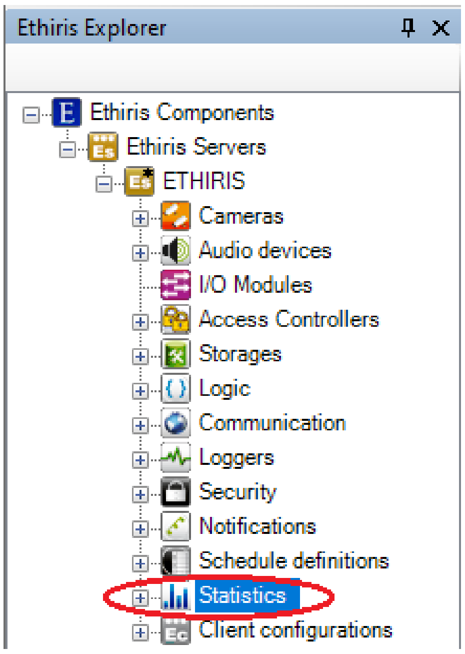


Figure 2.467 Statistics node in Ethiris Explorer treeview.

Statistics popup menu

This node has no popup menu.

Statistics panel

This node has no panel.

2.4.79 Cameras node

Under the *Statistics* node in the treeview, there is a *Cameras* node. The purpose of this node is to display statistics for all the cameras that are connected to the Ethiris Server.

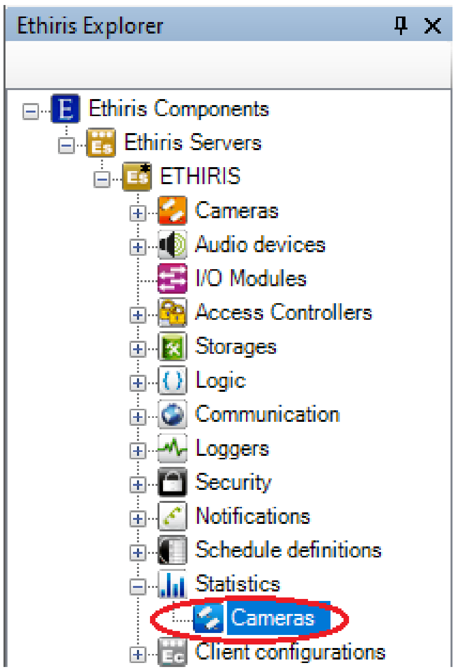


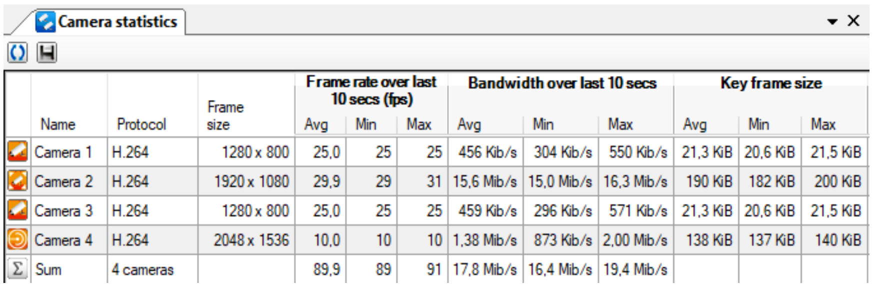
Figure 2.468 The Cameras node under Statistics in Ethiris Explorer treeview.

Cameras popup menu

This node has no popup menu.

Cameras panel

Double-clicking the *Cameras* node in the treeview opens the corresponding panel.



			Frame rate over last 10 secs (fps)			Bandwidth over last 10 secs			Key frame size		
Name	Protocol	Frame size	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
Camera 1	H.264	1280 x 800	25,0	25	25	456 Kib/s	304 Kib/s	550 Kib/s	21,3 KiB	20,6 KiB	21,5 KiB
Camera 2	H.264	1920 x 1080	29,9	29	31	15,6 Mib/s	15,0 Mib/s	16,3 Mib/s	190 KiB	182 KiB	200 KiB
Camera 3	H.264	1280 x 800	25,0	25	25	459 Kib/s	296 Kib/s	571 Kib/s	21,3 KiB	20,6 KiB	21,5 KiB
Camera 4	H.264	2048 x 1536	10,0	10	10	1,38 Mib/s	873 Kib/s	2,00 Mib/s	138 KiB	137 KiB	140 KiB
Sum	4 cameras		89,9	89	91	17,8 Mib/s	16,4 Mib/s	19,4 Mib/s			

Figure 2.469 Camera statistics panel.

This panel consists of a list of all the cameras that are currently part of the server configuration.

Cameras panel camera list

The camera list is comprised of several columns. All columns are read-only, i.e., you can't change any values in the list. The bottom row is a Sum row that sums up certain statistics for all available cameras, for instance, Avg bandwidth over the last 10 seconds, to give a quick overview of the capacity that is used. If using a cluster, each member gets a summary row. This makes it easier to see the load of each member.

Some of these statistics are available for usage in scripts and via OPC thru the subgroup *Statistics* on each camera, see the *Variable browser* in *Ethiris Admin*.

Click the button, *Refresh statistics information*, to update the figures in the list.

The statistical information is not dynamically updated.

Name is the name of the camera. It can't be changed here but is merely used for presentation. The icon to the left of the name indicates the current status of the camera. Orange indicates that the communication with the camera is OK, grey indicates that communication, for some reason, doesn't work.

Protocol shows which protocol the camera uses for sending images to Ethiris. It can be *MJPEG*, *MPEG-4*, *H.264,* or *H.265*.

Frame size shows the current image size in pixels.

Frame rate over last 10 secs (fps)

Avg shows the average frame rate in frames per second during the last 10 seconds (since you clicked *refresh*).

Min shows the lowest frame rate during the last 10 seconds.

Max shows the highest frame rate during the last 10 seconds.

Bandwidth over last 10 secs

There is no fixed unit, but it is adjusted based on current values.

Avg shows the average bandwidth during the last 10 seconds.

Min shows the lowest bandwidth during the last 10 seconds.

Max shows the highest bandwidth during the last 10 seconds.

Key frame size

If the protocol is *MJPEG* all frames are base frames. If the protocol is *MPEG-4* or *H.264* it's the so-called *I-frames* or *key frames* that are base frames. The values are fetched from the last 10 base frames. There is no fixed unit, but it is adjusted based on current values.

Avg shows the average size of a frame during the last 10 base frames.

Min shows the smallest size of a frame during the last 10 base frames.

Max shows the largest size of a frame during the last 10 base frames.

Sub frame size

If the protocol is *MJPEG,* there are no *sub frames*. If the protocol is *MPEG-4* or *H.264* it's the so-called *P-frames* or *B-frames* that are sub frames. The values are fetched from the latest 50 sub frames. There is no fixed unit, but it is adjusted based on current values.

Avg shows the average size of a frame during the last 50 sub frames.

Min shows the smallest size of a frame during the last 50 sub frames.

Max shows the largest size of a frame during the last 50 sub frames.

Recording over last week

There is no fixed unit, but it is adjusted based on current values. Note that the statistics are built up gradually, so if only a short period of time has passed since the Ethiris Server started, the statistics might not be accurate.

button can be clicked to open a bar chart where you can see the recording hour by hour or day by day.

Total shows the amount of video that has been recorded during the last week.

Avg shows the amount of video recorded on average per hour during the last week.

Min shows the amount of recorded video for the hour during the last week that had the smallest amount of recording.

Max shows the amount of recorded video for the hour during the last week that had the largest amount of recording.

Pre alarm over last week

There is no fixed unit, but it is adjusted based on current values. Note that the statistics are built up gradually, so if only a short period of time has passed since the Ethiris Server started, the statistics might not be accurate.

button can be clicked to open a bar chart where you can see the pre alarm buffer hour by hour or day by day.

Avg shows how large the pre alarm buffer has been on average during the last week.

Min shows the smallest size of the pre alarm buffer during the last week.

Max shows the largest size of the pre alarm buffer during the last week.

Last connect shows the last time a successful connection was made to the camera.

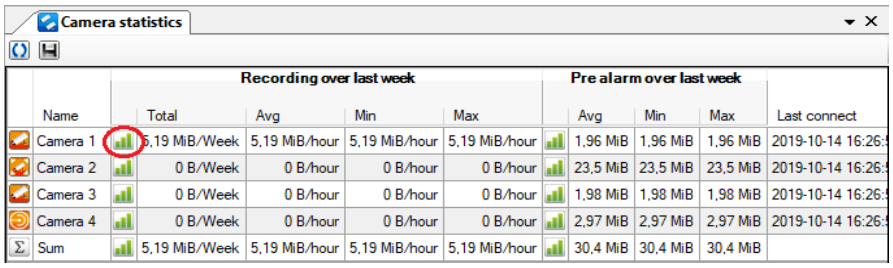
Last disconnect shows the last time there was a disconnection to the camera.

# of connection errors shows how many times there has been a connection error to the camera. Note that there has to be a successful connection between the disconnections for the disconnection to count.

# of dropped frame occurrences shows how many occurrences there have been of dropped frames in the communication with this camera. This is not the number of frames dropped during those occurrences.

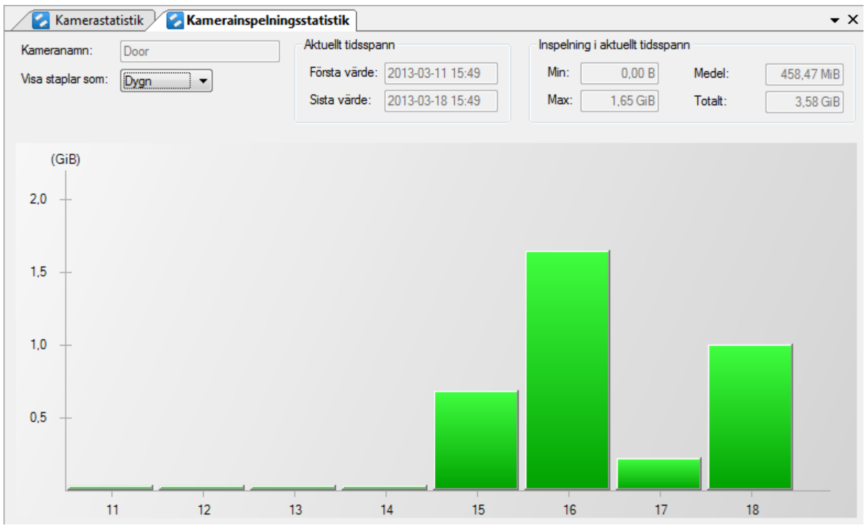
Camera recording statistics panel

By clicking the bar chart button (ringed in on the figure below), you open a panel with statistics over the last week as a bar chart.


Camera statistics

		Recording over last week					Pre alarm over last week			
Name		Total	Avg	Min	Max		Avg	Min	Max	Last connect
Camera 1		5,19 MiB/Week	5,19 MiB/hour	5,19 MiB/hour	5,19 MiB/hour		1,96 MiB	1,96 MiB	1,96 MiB	2019-10-14 16:26:
Camera 2		0 B/Week	0 B/hour	0 B/hour	0 B/hour		23,5 MiB	23,5 MiB	23,5 MiB	2019-10-14 16:26:
Camera 3		0 B/Week	0 B/hour	0 B/hour	0 B/hour		1,98 MiB	1,98 MiB	1,98 MiB	2019-10-14 16:26:
Camera 4		0 B/Week	0 B/hour	0 B/hour	0 B/hour		2,97 MiB	2,97 MiB	2,97 MiB	2019-10-14 16:26:
Sum		5,19 MiB/Week	5,19 MiB/hour	5,19 MiB/hour	5,19 MiB/hour		30,4 MiB	30,4 MiB	30,4 MiB	

Figure 2.470 Button for bar chart is ringed in.


Kamerastatistik / Kamerainspelningsstatistik

Kameranamn: Door
Visa staplar som: Dygn

Aktuellt tidsspann
Första värde: 2013-03-11 15:49
Sista värde: 2013-03-18 15:49

Inspelning i aktuellt tidsspann
Min: 0,00 B
Max: 1,65 GiB
Medel: 458,47 MiB
Totalt: 3,58 GiB

Figure 2.471 Bar chart for recording.

Besides the bars themselves, this panel consists of several fields where all but one are read-only. You can choose to display the bars summarized per *Days* or per *Hours*.

At the bottom, under the bars, there are numbers, in this example, *11 – 18*. These indicate a certain day, in this case, March 11th to March 18th, which also is shown in the *Recording in current time span* box at the top of the panel.

If you hover the mouse pointer over one of the bars, the exact value for the bar is shown in a small information box.

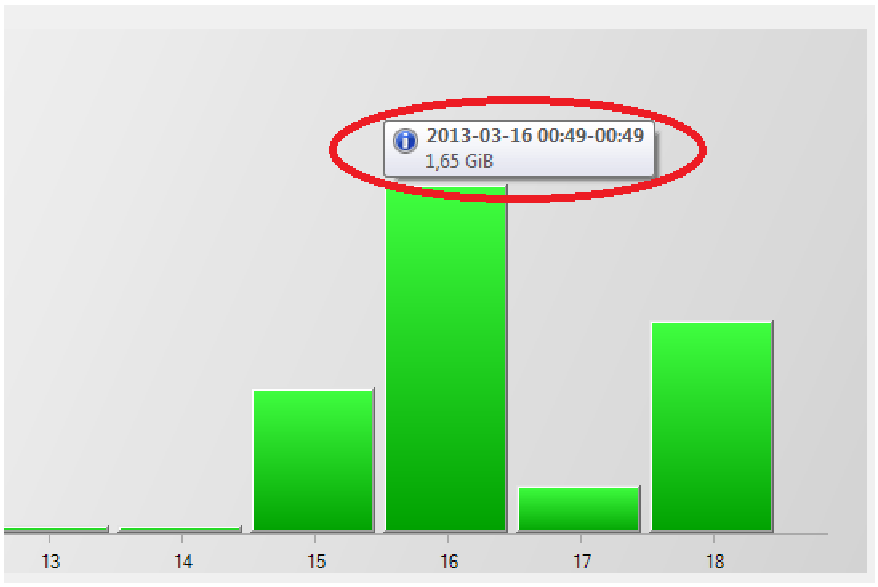
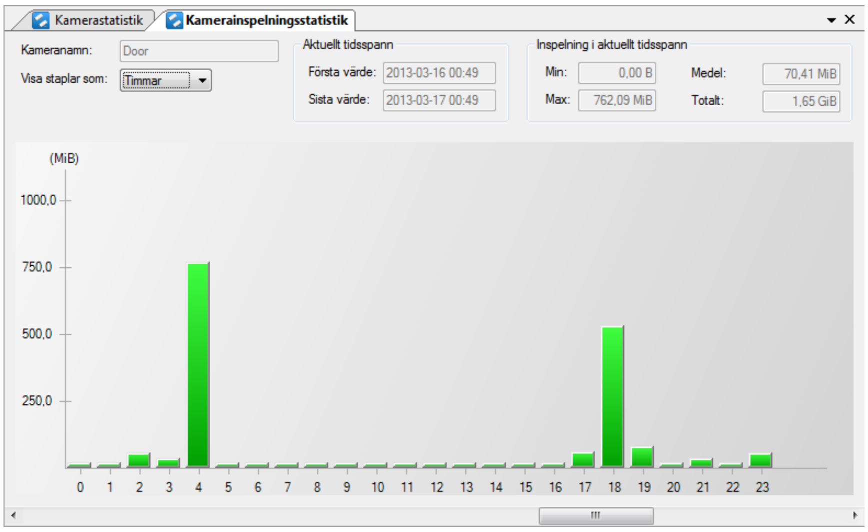


Figure 2.472 Information about a certain bar.

The information in this example tells us that the camera has been recording *1.65 GiB* during March 16. To be accurate, the time span is between *00: 49* March 16 to *00:49* March 17. This has to do with when Ethiris Server was started, it immediately starts collecting statistical data about the cameras, and in this case, it happened to be at the 49th minute at a certain hour.

To view the statistics per hour instead, you can double-click the desired day bar. Alternatively, you can select *Hours* in the list *Display bars as:*



Kameranamn: Door
Visa staplar som: Timmar

Aktuellt tidsspann
Första värde: 2013-03-16 00:49
Sista värde: 2013-03-17 00:49

Inspelning i aktuellt tidsspann
Min: 0,00 B
Max: 762,09 MiB
Medel: 70,41 MiB
Totalt: 1,65 GiB

Figure 2.473 Statistics per hour.

In the chart, all 168 hours are represented, but only 1 day at a time (24 bars) are displayed. Use the scrollbar at the bottom to move forward/backward in time.

Camera pre alarm statistics panel

By clicking the bar chart button for *Pre alarm,* the corresponding panel is opened. It works in the same way as the panel for camera recording described above.

Camera statistics panel toolbar

Figure 2.474 The toolbar in the Camera statistics panel.

Refresh statistics information	Use this button to refresh the statistical values in the panel.
Export statistics information	Use this button to export the statistical values to a *CSV*-file. This file can then be imported to Excel for further processing.

2.4.80 Client configurations node

Under each Ethiris Server in the treeview, there is a *Client configurations* node. Under this node, there can be a number of client configurations that this particular Ethiris Server is handling. Ethiris Server is now handling all client configurations, and Ethiris Client will contact a configuration server (simply an Ethiris Server) to fetch a configuration.

In this way, the handling of client configurations is greatly simplified. Ethiris Client can search the network for Ethiris Servers. For each server, found the available configurations are listed.

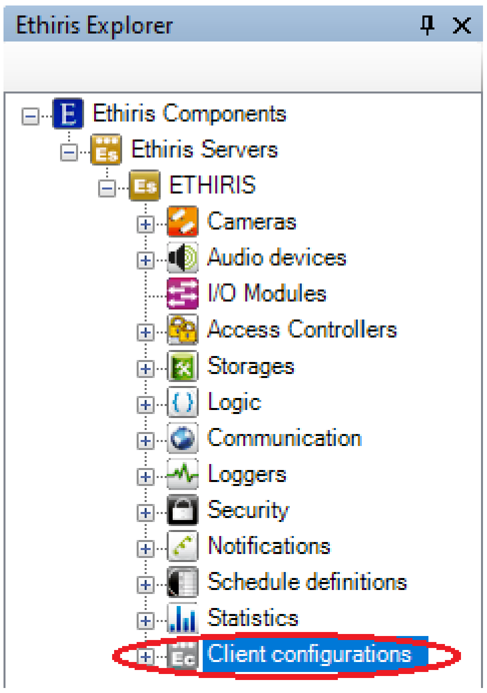


Figure 2.475 The Client configurations node in Ethiris Explorer

Client configurations popup menu

Right-click on this node opens a menu.

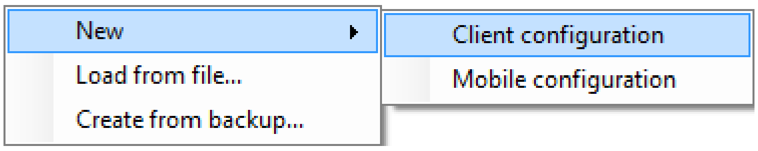


Figure 2.476 Popup menu for the node Client configurations.

New->Client configuration will add a new, empty, configuration for Ethiris Client to this server. It's available immediately in the treeview as a new client configuration node. If the panel *Client configurations* is open, it will show up there as well.

New->Mobile configuration will add a new, empty, configuration for Ethiris Mobile to this server. It's available immediately in the treeview as a new mobile configuration node. If the panel *Client configurations* is open, it will show up there as well. Note that, to be able to use mobile configurations in Ethiris Mobile, version 5.82.x and higher is required.

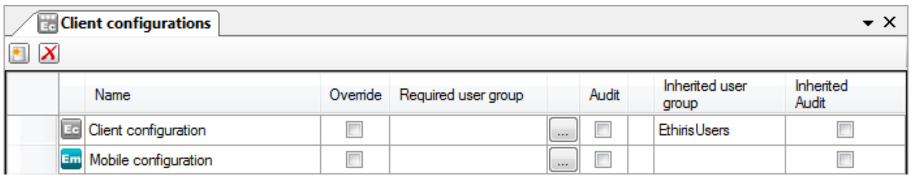
Load from file… will import a client configuration from a file (*.ecc). This is important if you have existing client configurations from earlier Ethiris versions (7.x and earlier) that you would like to use. When upgrading from an earlier version of Ethiris, the client configurations that are already part of the Ethiris Admin project will be imported automatically. If you have several client configurations saved in files or if you have client configuration files stored on another computer/server, they must be imported manually using this function.

After being imported, they will be handled by Ethiris Server, and the *.ecc file will not be required anymore. When importing a client configuration file, the file is not modified.

Create from backup… Restores a client configuration from a backup to this server. Note that this will create a copy of the backed-up configuration with a new id. If you want to restore a backup on an existing configuration, open the configuration in Admin first by double-clicking it, see *Restore configuration… on page 2:320.*

Client configurations panel

Double click on the node *Client configurations* in the treeview opens the corresponding panel. Note that to be able to use mobile configurations, you need version 5.82 or later of Ethiris Mobile.



Name	Override	Required user group		Audit	Inherited user group	Inherited Audit
Client configuration	☐		...	☐	EthirisUsers	☐
Mobile configuration	☐		...	☐		☐

Figure 2.477 The panel Client configurations.

This panel has a list of all client configurations that are handled by this server.

At the top, there is a toolbar.

Client configurations panel toolbar

Figure 2.478 The toolbar in the panel Client configurations

New client configuration

Use this button to create a new client configuration. This is the same as *New->Client configuration* in the popup menu for the *Client configurations* node described above. A new client configuration is immediately added to the server configuration.

Delete selected Client configuration(s)

Use this button to delete the selected client configurations from the server configuration. You can select more than one client configuration by using the *Ctrl*-button and/or the *Shift*-button.

Client configurations panel list of client configuration

The list of client configurations has several columns.

Name is the name of the client configuration. The name is displayed in the title bar in Ethiris Client together with the name of the Ethiris Server that handles the client configuration.

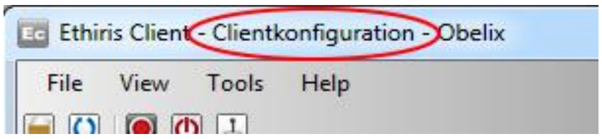


Figure 2.479 The name of the client configuration in the title bar of Ethiris Client.

Override has to be explicitly checked to be able to change the *Required User Group*. A Security setting can be set at the Ethiris Server level, meaning that all client configurations have the same security settings unless they are specifically overridden in this panel by checking *Override*.

Required User Group specifies what user group the user has to be a member of, to access this configuration. A blank field means that no log in is required. You can browse for available user groups by clicking the *browse* button to the right of this column. The user must possess both *Show Client configuration* and *Load Client configuration* for the configuration to be listed when browsed from the client.

To be able to load a client configuration, it is sufficient to have *Load Client configuration* privileges. If *Show Client configuration* privilege is not held, the configuration is not listed when the user is browsing for configurations on the server, but it is still possible to load the configuration.

The browse button can be used to browse for available user groups.

Audit can be checked if you want to log when this configuration is accessed. For each configuration for which *Audit* is specified, the system will save information on the time, configuration name, who performed it, the client computer from which it was performed, and any other available information.

It is permitted to specify that the system must log an operation without, at the same time, making any requirement that the user must be a member of a specific group. The operation will be logged regardless of this, but in these cases, there may be no information on who performed the operation if the user has not logged on. Other available information is logged as usual.

The audit log can be viewed in the *Events* panel in Ethiris Client.

Inherited User Group is just information on the current setting on the Ethiris Server level. To override this setting, check the *Override* checkbox.

Inherited Audit is just information on the current setting on the Ethiris Server level. To override this setting, check the *Override* checkbox

2.4.81 Client configuration node

Under *Client configurations,* there can be one or more *Client configuration* nodes. These configurations are handled by this server.

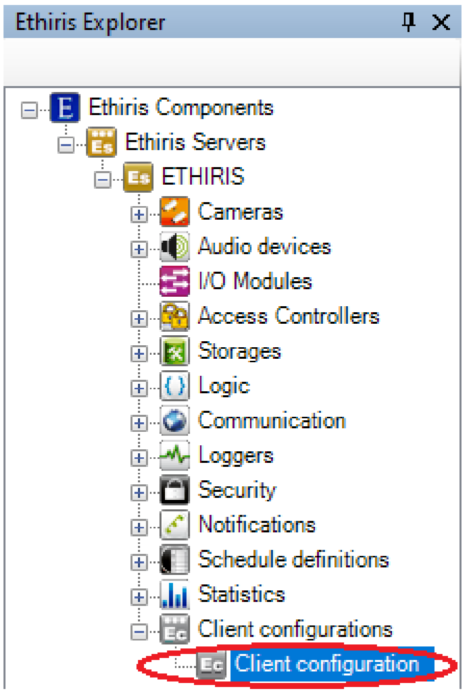


Figure 2.480 A Client configuration node in Ethiris Explorer treeview.

Client configuration node popup menu

Right-click on this node opens a menu.

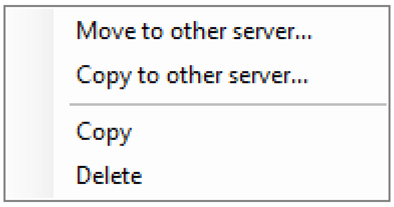


Figure 2.481 The popup menu for a Client configuration node.

Move to other server... Shows a dialog where you can choose another server that should handle this client configuration from now on. The servers to select from must be loaded in Ethiris Admin. Select a server from the list and click the *Select* button. To cancel the dialog, click the X in the upper right corner.

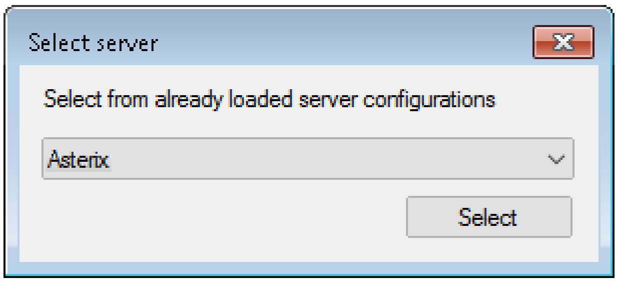


Figure 2.482 The dialog to select a new server for a Client configuration.

Copy to other server… Shows a dialog where you can choose another server that will receive a copy of this configuration. The servers to select from must be loaded in Ethiris Admin. Select a server from the list and click the *Select* button. To cancel the dialog, click the X in the upper right corner. Note that the configurations will get different ids and hence be different configurations from the systems' point of view even though they have the same name.

Copy will simply create a copy of the selected client configuration on the same server but with a new name.

Delete will erase the selected client configuration from the server. Any Ethiris Client that is currently running the deleted configuration will close its configuration and wait for the user to open another one.

Client configuration panel

Double click on a Client configuration node will load the client configuration in Ethiris Admin under the Ethiris Clients node see *2.4.82. Ethiris Clients node.*

2.4.82 Ethiris Clients node

The *Ethiris Clients node* is just a collection node for all Ethiris Clients in the current project. There is neither a popup menu nor a panel associated with this node. There are two different types of client configurations, those for Ethiris Client and those for Ethiris Mobile. Note that at least version 5.82 of Ethiris Mobile is required to be able to use client configurations for Ethiris Mobile.

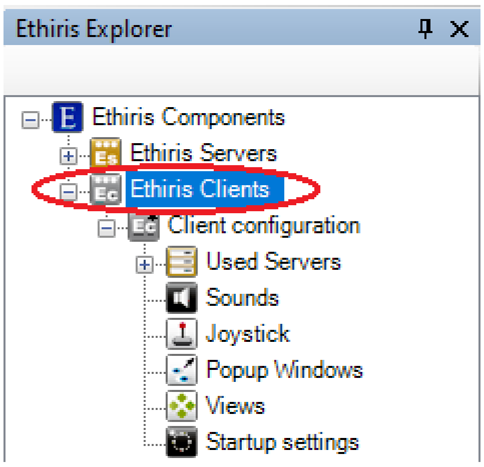


Figure 2.483 The Ethiris Clients node in Ethiris Explorer treeview.

2.4.83 Ethiris Client configuration node

Under the *Ethiris Clients node,* there might be one or several *Ethiris Client* nodes, each one representing an Ethiris Client configuration in the system.

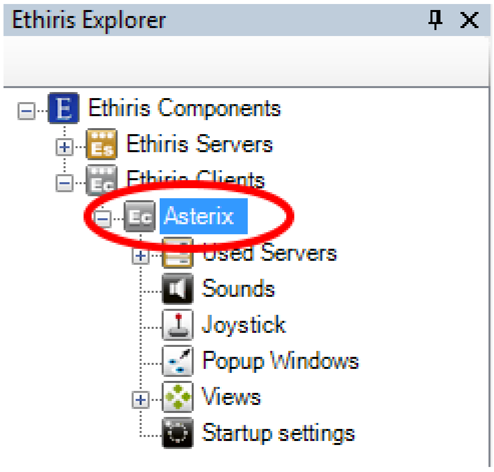


Figure 2.484 The Ethiris Client node in Ethiris Explorer treeview.

Ethiris Client popup menu

Right-clicking this node brings up a context menu.

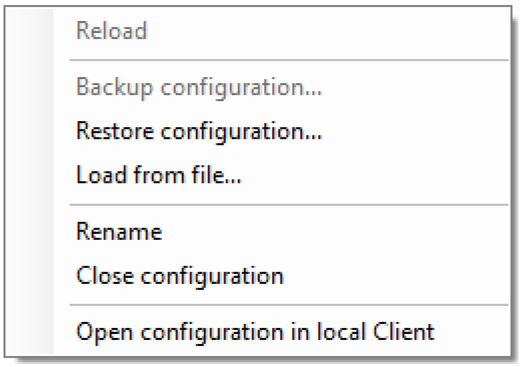
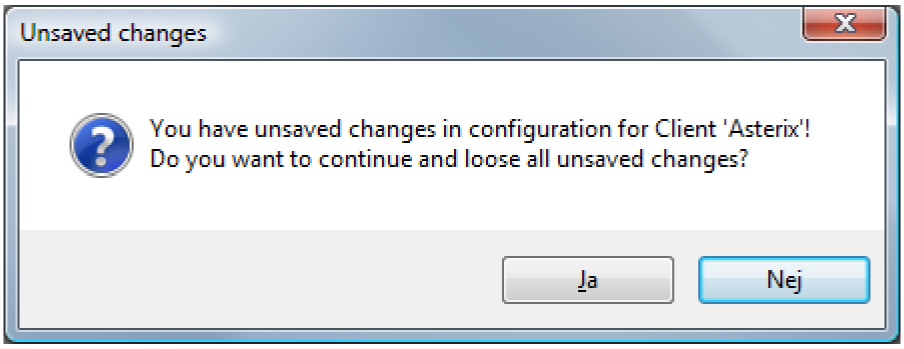


Figure 2.485 The popup menu for the Ethiris Client node.

Reload is only available if you have made changes to the client configuration. *Reload* reads the configuration from the server and loads it into Ethiris Admin. Any open panels belonging to the client will be closed. Before the configuration is reloaded, you will be prompted about unsaved changes and have a chance to change your mind.



You have unsaved changes in configuration for Client 'Asterix'!
Do you want to continue and loose all unsaved changes?

Figure 2.486 Unsaved changes dialog.

Click *Yes* for reloading anyway or click *No* to not reload and get the chance to save your changes first.

Backup configuration… is more or less the same function as described in section *2.4.2 Ethiris components node* on page *2:44*, with the small difference that in this context backup is performed only for the current Ethiris Client.

Restore configuration… is also the same function as described earlier. In this context, it is about restoring the Ethiris Client configuration.

Load from file… imports a client configuration from an earlier version of Ethiris.

Rename sets the node in the treeview in change mode. You can enter a new name directly in the treeview.

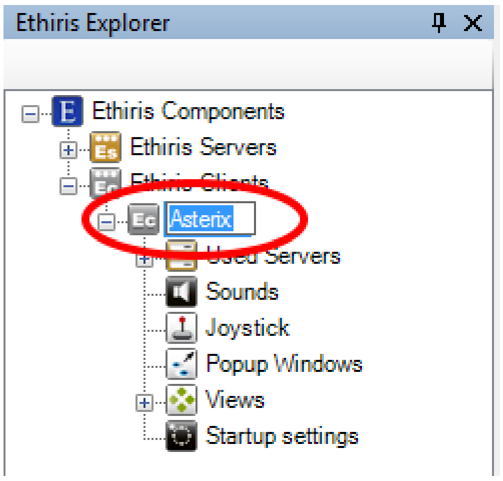


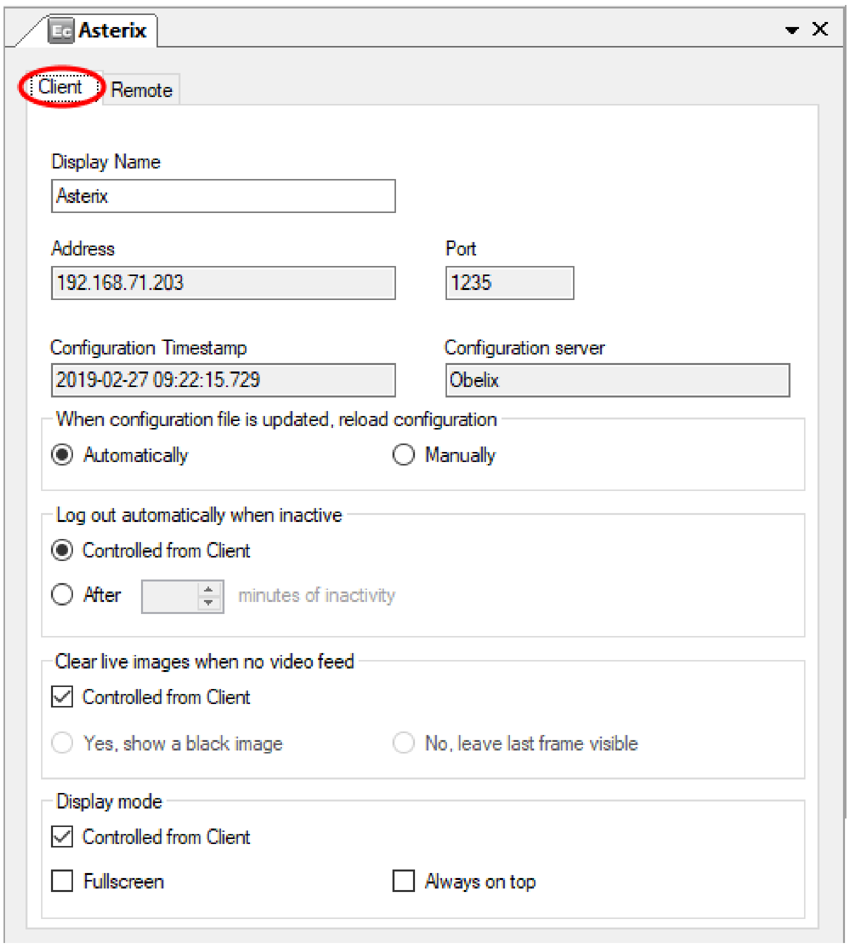
Figure 2.487 Rename client.

Close configuration removes the client configuration from Ethiris Admin. Note that the configuration itself is not changed, it's only the current project in Ethiris Admin that's affected.

Open configuration in local Client opens the client configuration in Ethiris Client on the local computer (if Ethiris Client is installed on the local computer).

Ethiris Client panel

Double-clicking the *Ethiris Client* node in the treeview opens the corresponding panel.



Display Name: Asterix

Address: 192.168.71.203

Port: 1235

Configuration Timestamp: 2019-02-27 09:22:15.729

Configuration server: Obelix

When configuration file is updated, reload configuration
● Automatically ○ Manually

Log out automatically when inactive
● Controlled from Client
○ After [ ] minutes of inactivity

Clear live images when no video feed
☑ Controlled from Client
○ Yes, show a black image ○ No, leave last frame visible

Display mode
☑ Controlled from Client
☐ Fullscreen ☐ Always on top

Figure 2.488 The Ethiris Client panel.

The *Ethiris Client* panel consists of two tabs; *Client* and *Remote*.

Client

Here you find information about the Ethiris Client itself.

Display Name is the name you gave the Ethiris Client component when adding it to the project. It can be changed. As you change the name, it is immediately updated in the treeview. This name is only used for displaying an appropriate name for the client in the treeview.

Adress is the IP-address to the Ethiris Server that is maintaining this client configuration. This field is read-only.

Configuration Timestamp indicates when the client configuration was last saved. This field is read-only.

Configuration server is the name of the Ethiris Server that is maintaining this client configuration. This field is read-only.

When configuration file is updated, reload configuration determines if clients who are started and have this configuration loaded automatically will reload a new version of the configuration when the configuration is updated.

Log out automatically when inactive can be used to automatically log out the logged in user in Ethiris Client. The default setting, *Controlled from Client*, means that the operator self has to log out from the client manually. If you select the setting *After* a certain number of minutes, the log out is performed automatically after the specified time. You can set a time between 1 – 120 minutes.

Clear live images when no video feed determines if the live image in the client should be cleared if/when live video is not streamed. This is valid in all cases if a camera is shown in a camera view in the client, but for some reason, video cannot be streamed, for example, not enough privilege to show live video, the camera is disabled, and so on.

If *Controlled from Client* is checked, the user locally on each client computer can decide how the client should work in this respect.

If you want, you can force a specific functionality on the client, by unchecking the *Controlled from Client* and select *Yes, show a black image* or *No, leave last frame visible.* In this case, the user locally on the client computer cannot change this setting.

Display mode determines if the main window in the client should be displayed in fullscreen, i.e., without title bar and borders. You can also decide if the main window and all other windows should be *Always on top.* This means that no other windows will be placed on top of any window from Ethiris Client. Note that we cannot guarantee that, in all situations, with any other application installed, that no other window can show on top of the client, but the risk of this happening is highly reduced.

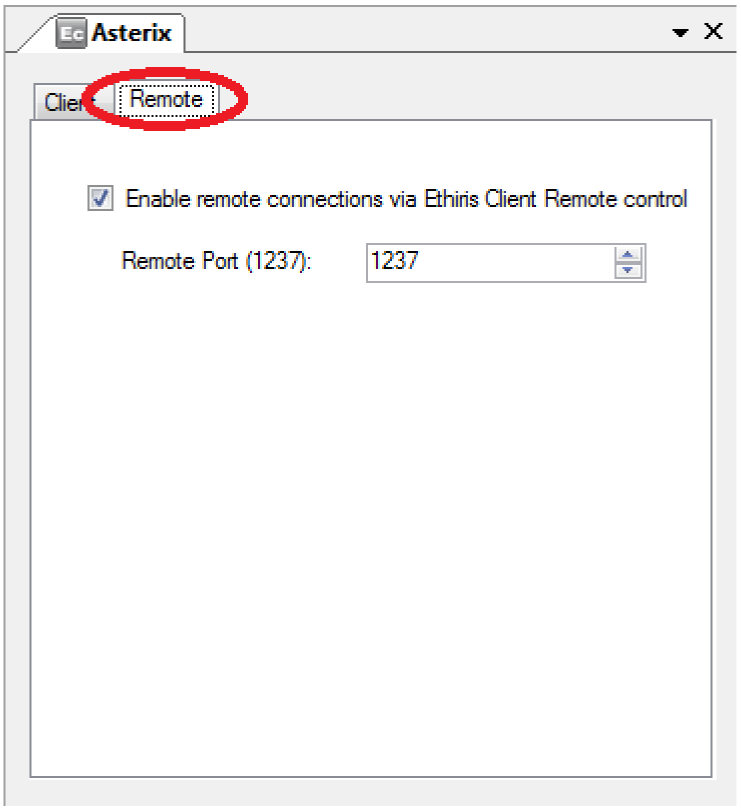
If *Controlled from Client* is checked, the user locally on each client computer can decide how the client should work in this respect.

Check the *Fullscreen* box if you want the client's main window to start in fullscreen mode, i.e., without title bar and borders. If allowed to be controlled by the client, the user can toggle this mode by clicking *F11.*

Check the *Always on top* box if you want the client main windows and all other windows to be on top of windows from other applications. If allowed to be controlled from the client, the user can toggle this mode using the *View* menu in the client.

Remote

In this tab is information about how other applications communicate with this Ethiris Client via the *Remote control of Ethiris Client* interface.



[✓] Enable remote connections via Ethiris Client Remote control

Remote Port (1237): 1237

Figure 2.489 The Remote tab in the Ethiris Client panel.

Enable remote connections via Ethiris Remote control. When checked, Ethiris Client listens to calls from other systems on a specific port. The purpose is to remote control the client and determines what is displayed in Ethiris Client. There is a sample on the Ethiris installation CD with example source code in Visual Basic .NET on how to remote control an Ethiris Client.

Remote port is the port the Ethiris Client listens to for incoming calls. As a default, this port is *1237*.

2.4.84 Used Servers node

Under each Ethiris Client node, there is a *Used Servers* node. This is a collection node for all Ethiris Servers that this client configuration will connect to.

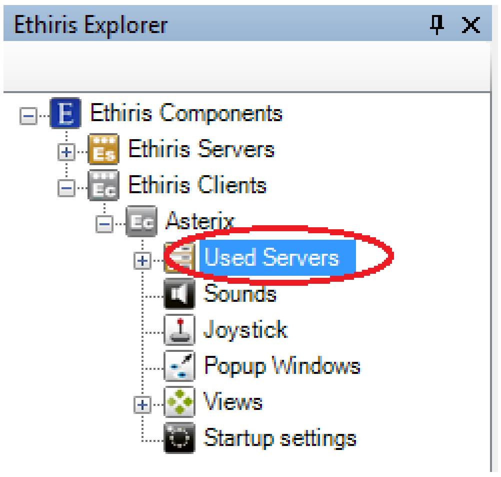


Figure 2.490 The Used Servers node in Ethiris Explorer treeview.

Used Servers popup menu

Right-clicking this node brings up a context menu.

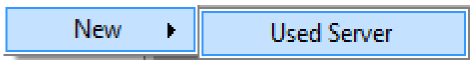


Figure 2.491 The popup menu for the Used Servers node.

New->Used Server brings up a dialog for connecting to an Ethiris Server. The dialog looks different depending on if there are any available Ethiris Servers in the current Admin project. In this case, the Ethiris Server named *Obelix* is available and thus presented in the list of already loaded server configurations.

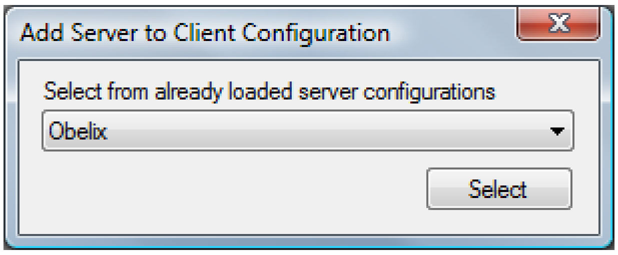
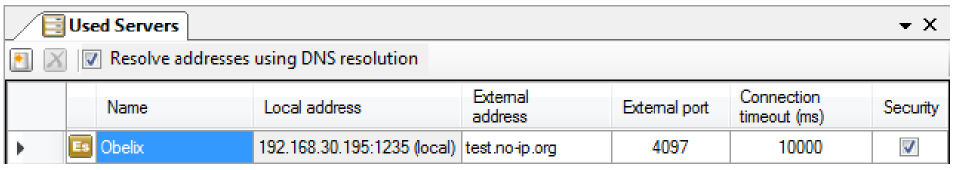

Add Server to Client Configuration
Select from already loaded server configurations
Obelix
Select

Figure 2.492 Add server to client configuration dialog.

Select an Ethiris Server in the list and click *Select* for connecting the server to this client configuration.

Used Servers panel

Double-clicking the *Used Servers* node in the treeview opens the corresponding panel.

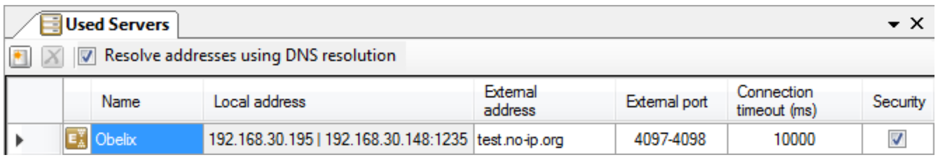


Name	Local address	External address	External port	Connection timeout (ms)	Security
Obelix	192.168.30.195:1235 (local)	test.no-ip.org	4097	10000	✓

Figure 2.493 The Used Servers panel.

This panel consists of a list of all currently selected Ethiris Servers in the client configuration.

Below is an example of how the panel will look when using a cluster of Ethiris Servers. The icon for the server will change, multiple addresses will appear in *Local address,* and multiple external ports will appear in *External port* (if used).



Used Servers

Resolve addresses using DNS resolution

Name	Local address	External address	External port	Connection timeout (ms)	Security
Obelix	192.168.30.195 | 192.168.30.148:1235	test.no-ip.org	4097-4098	10000	✓

Figure 2.494 The UsedServers panel with a cluster of Ethiris Servers.

At the top of the panel, there is a toolbar.

Used Servers panel toolbar

Figure 2.495 The toolbar in the Used Servers panel.

Add a new server

Use this button to add a new Ethiris Server. This is the same as *New->Used Server* in the popup menu for the Used Servers node described above.

Delete selected server(s)

Use this button to delete the selected Ethiris server(s) from the configuration. You can select more than one server by using the *Ctrl*-button and/or the *Shift*-button.

Resolve addresses using DNS resolution is selected as default. This means that Ethiris Client queries a DNS server about the IP addresses for each Ethiris Server in the list and can then figure out if any of the servers run on the local computer (same as where Ethiris Client runs). In that case, the communication can be directed to *localhost,* which is more efficient. In some cases, the IT policy is so strict that it is not permitted to query a DNS server in this fashion, which in turn can lead to long timeouts. Hence the possibility to turn this function off by deselecting the checkbox.

Used Servers panel server list

The server list consists of several columns.

Name is the desired name of the server. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the server in the list indicating the problem. This name is used for identifying the server in various contexts.

Local address is the IP address (and port) the client will use to connect to the server. This field is read-only. It is merely for information on which physical computer the Ethiris Server runs on. The address *127.0.0.1* is an alias for the local computer, i.e., the computer you work on for the moment.

Port is the TCP/IP port that the Ethiris Server listens to for incoming calls from Ethiris Clients. The port has to be determined when the server is connected under *Ethiris Servers* in Ethiris Admin. This is by default *1235,* and there usually is no reason for changing it.

External address is the IP address the Ethiris Client will use to try to connect to the server if the connection on the *local address* fails. This is to be able to use the same client configuration both locally and externally. Here you would enter the publicly accessible IP address of the site where Ethiris Server runs. It could also be a dynamic address (DNS name). If you don't want to use this feature, leave the field empty.

Ethiris Client normally tries to connect using the *Local address*, but if that fails and connection using *External address* and *External port* succeeds, Ethiris Client will, after that, use the E*xternal address* first. If that connection fails, connection using *Local address* will be tried. The client will try the local and external addresses until a connection is made. When a connection is successful, the client will remember if it was the local or external address and try that one first the next time the client is started.

External Port is the port mapped in the router that redirects incoming connections to the correct computer on the local network. Refer to the manual of your router for more information regarding this. If not in use, set value to 0 (will be displayed as a blank field).

Note, if you have a cluster and wish to use external address/port, you still can enter only one port number for the external connection. In that case, Ethiris Client assumes that the highest-ranked server in the cluster can be connected on the (first) port and the next server on the next port and so on, i.e., like 4097-4098 if you enter port 4097 and have a cluster with two members.

Connection timeout (ms) is the number of milliseconds a client waits for the Ethiris Server to respond on calls to the server. *10 000* ms is a default, i.e., *10* seconds.

Security determines which Ethiris Server that acts as a *Security server* for this client configuration. The Security server verifies user privileges that are client-specific. These are the last 6 *Operations* in the *Server Security* panel in the server configuration. The 6 operations are *Start client*, *Allow export of video from client*, *Show player in client*, *Show event list in client*, *Show alarm list in client* & *Exit client*. You can read more about Server Security in section *Security node* on *page 2:255*. Only one server can be the security server. In the case of several used servers in the client configuration, you can check the desired Ethiris server that will act as a Security server.

2.4.85 Used Server node

Under the Used Servers node, all used Ethiris Servers are listed. Each node represents an Ethiris Server.

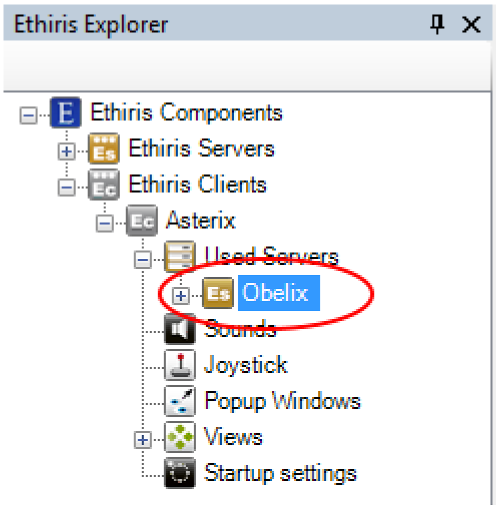


Figure 2.496 A Used Server node in Ethiris Explorer treeview.

If the used Ethiris Server is a cluster, the node will use the icon for a Server Cluster.

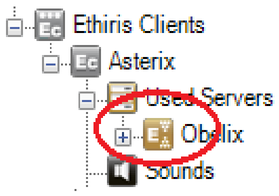


Figure 2.497 The Ethiris Server node represents a cluster node.

Used Server popup menu

Right-clicking this node brings up a context menu.

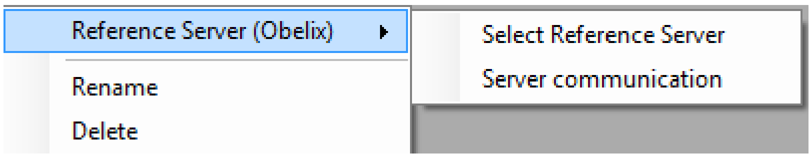


Figure 2.498 The popup menu for a Used Server node.

Reference Server (<Name>)->Select Reference Server brings up a dialog for selecting an Ethiris Server. If there is a *name* in parenthesis, it means that a reference already exists. If no reference exists, it means that Ethiris Admin has no idea of which cameras and I/Os that are available in the Ethiris Server.

When you first add an Ethiris Server to the used servers list, the reference is set automatically. The only reason a reference should not exist is that the client configuration has been opened in Ethiris Admin before the corresponding Ethiris Server configuration has been opened in Ethiris Admin. In this case, you can connect to the Ethiris Server, then select it as a reference for the corresponding used server in the client configuration. You can also use this menu item for changing the referenced server to another Ethiris Server that is loaded into Ethiris Admin.

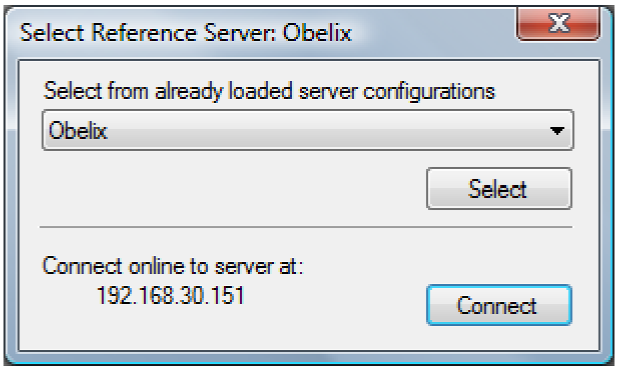

Select Reference Server: Obelix
Select from already loaded server configurations
Obelix
Select
Connect online to server at:
192.168.30.151
Connect

Figure 2.499 Select reference server dialog.

Select the desired server in the list and click *Select* alternatively click *Connect* to select the local Ethiris Server as reference.

Reference Server (<Name>)->Server communication brings up a dialog for selecting which IP address to use when connecting to the Ethiris Server(s). This is important when a server has multiple IP addresses.

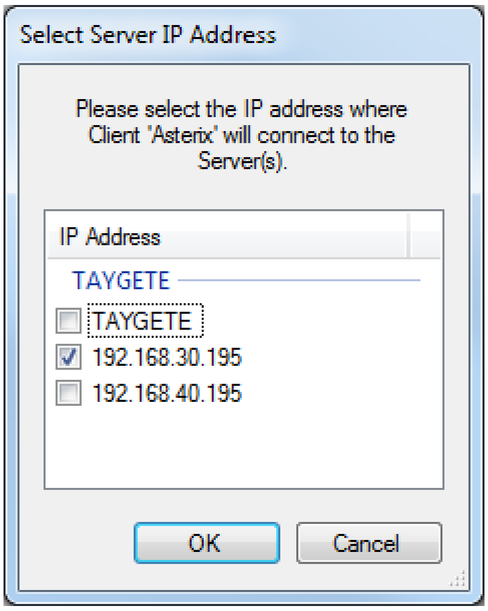
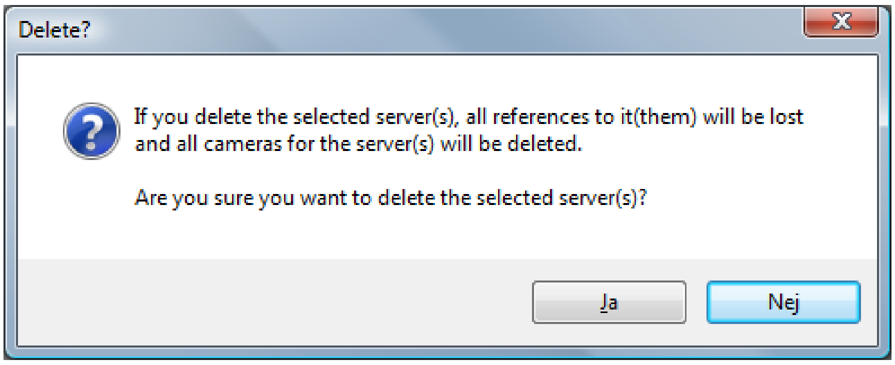


Figure 2.500 The dialog for Server communication."TAYGETE" is the name of the computer which the Ethiris Server is running on.

Rename sets the node in *rename* mode. You can enter a new name for the server directly in the treeview.

Delete removes the Ethiris Server from the used servers list for the client configuration. If you delete the server, all references to cameras and I/Os belonging to that server will disappear from the client configuration. A view, for example, will be kept with the layout intact, but all camera references in the camera views will be deleted. Before deletion, a confirm dialog is displayed.



Delete?

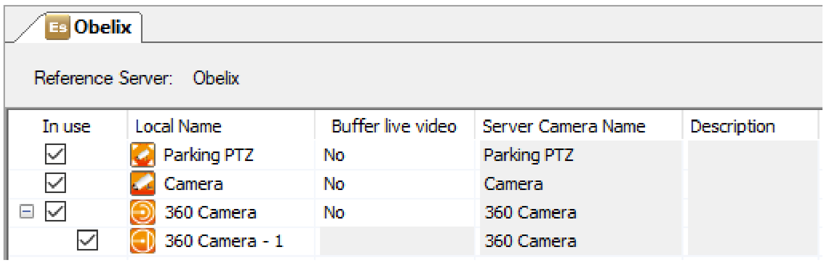
If you delete the selected server(s), all references to it(them) will be lost and all cameras for the server(s) will be deleted.

Are you sure you want to delete the selected server(s)?

Figure 2.501 Dialog for confirming the deletion of a used server.

Used Server panel

Double-clicking the *Used Server* node in the treeview opens the corresponding panel.



Reference Server: Obelix

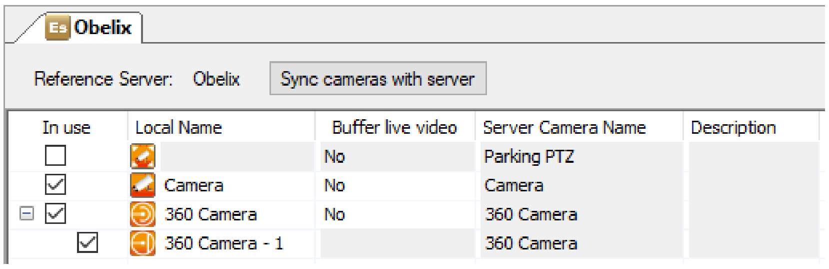
In use	Local Name	Buffer live video	Server Camera Name	Description
✓	Parking PTZ	No	Parking PTZ	
✓	Camera	No	Camera	
⊟ ✓	360 Camera	No	360 Camera	
✓	360 Camera - 1		360 Camera	

Figure 2.502 The Used Server panel.

This panel consists of a list of all available cameras in the referenced Ethiris Server.

At the top of the panel, the name of the currently referenced Ethiris Server is displayed.

A button, ”Sync cameras with server”, will appear if one or more cameras are not checked for usage. If pressed, it will automatically enable all cameras for usage and enter the server’s camera names as the local names.

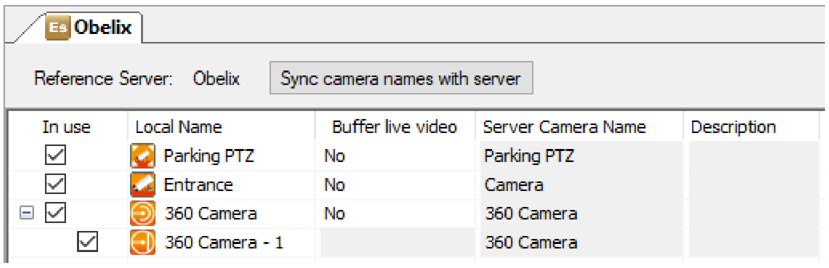


Reference Server: Obelix

In use	Local Name	Buffer live video	Server Camera Name	Description
☐		No	Parking PTZ	
✓	Camera	No	Camera	
⊟ ✓	360 Camera	No	360 Camera	
✓	360 Camera - 1		360 Camera	

Figure 2.503 The Used Server panel with the button ”Sync cameras with server” visible, due to a disabled camera.

If any local camera name does not match the server camera name, another button will appear, ”Sync camera names with server”. When pressed, it will change all local camera names to the corresponding server camera name.


Obelix

Reference Server: Obelix   Sync camera names with server

In use	Local Name	Buffer live video	Server Camera Name	Description
✓	Parking PTZ	No	Parking PTZ	
✓	Entrance	No	Camera	
✓	360 Camera	No	360 Camera	
✓	360 Camera - 1		360 Camera	

Figure 2.504 The Used Server panel with the button "Sync camera names with server" visible, due to a different local name than the server camera name.

Used Server panel camera list

The camera list consists of several columns. You can sort the list by clicking the desired column header.

Use. Check this box to use the corresponding camera in the client configuration. This means that the camera is available for camera views and in the *Cameras panel* in Ethiris Client. When first adding a used server, all cameras are selected. If you add more cameras to the server configuration after the Ethiris Server is referenced in the client configuration, you have to manually select the new cameras by checking the *Use* column.

Be aware that if you deselect a virtual camera in the camera list, it will be deleted from the configuration and you will have to recreate it again if you want it back.

Local name is the name of the camera you want in Ethiris Client. This need not be the same as the name used for the camera in Ethiris Server. The local name has to be unique. The column also contains an icon that indicates which type of camera this is.

Fixed camera

This icon indicates a fixed camera. The icon is also used for cameras that only exist in the client configuration (but don't reference any camera in the server configuration), as it is then not possible to know what type of camera it would be.

PTZ camera

This icon indicates a camera that supports PTZ in some way. This also means that cameras only supporting optical zoom will get this icon.

360 camera

This icon indicates that the camera is a 360 camera that Ethiris supports. This means that Ethiris Client will be able to dewarp the image from the camera, both in live and in the player.

Virtual camera with a predefined dewarping mode and a predefined PTZ position

This icon is shown for virtual cameras that have been created in the client configuration. Starting with a 360 camera, you can create virtual cameras that are a predefined dewarp mode or a predefined PTZ position. When the virtual camera is shown in the client view, it will always start from this predefined mode/position. You can manually change the dewarp mode and PTZ position.
Virtual cameras can be used exactly like any regular camera when creating views in the client.

Buffer live video determines the default setting for live buffering. This setting can be changed in live in Ethiris Client by right-clicking the camera view and select *Buffer live video* in the popup menu. The purpose of live buffering is to make sure that live video runs smoothly at the right place. It can be used when the camera intermittently delivers video, which without buffering renders live video unsmooth. It is only cameras that deliver a so-called RTSP video stream (that contains timestamp information for each frame) where a setting other than *No* can be selected. You can set a fixed time between 50 – 1 000 ms. You can also select *Auto*. Auto refers to the fact that Ethiris automatically tunes how much video has to be buffered for smooth live performance. Note that buffering creates some latency in live. Also, note that a maximum of 30 frames will be buffered regardless of the selected time. This is to avoid memory exhaustion.

Server camera name is the name used for the camera in Ethiris Server. This field is read-only.

Description is an optional description entered for the camera in the Ethiris Server configuration. This field is also read-only.

2.4.86 Camera node

Under each Used Server node in the treeview, all available cameras in the Ethiris Server are presented as treeview nodes.

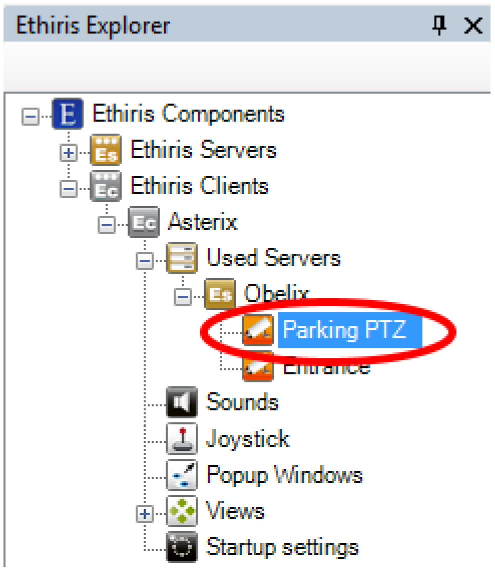


Figure 2.505 A Camera node in Ethiris Explorer treeview.

Camera Change order

The order in which the cameras are presented in the treeview is used in various contexts where a list of cameras is presented, e.g., when you select a camera into a camera view or in the camera list in Ethiris Client.

You can rearrange the order of cameras in the treeview by dragging a camera to another position. Click the desired camera with the left mouse button, hold the button down and, at the same time, move the mouse pointer to another camera in the list, release the mouse button when the mouse pointer is at the desired position in the treeview.

When a camera is moved down in the treeview, the camera will be positioned *after* the camera it was dropped on. If the camera is moved up in the treeview, the camera will be positioned *before* the camera it was dropped on.

Camera popup menu

Right-clicking this node brings up a context menu.

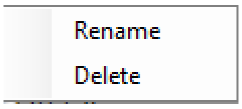


Figure 2.506 The popup menu for a Camera node.

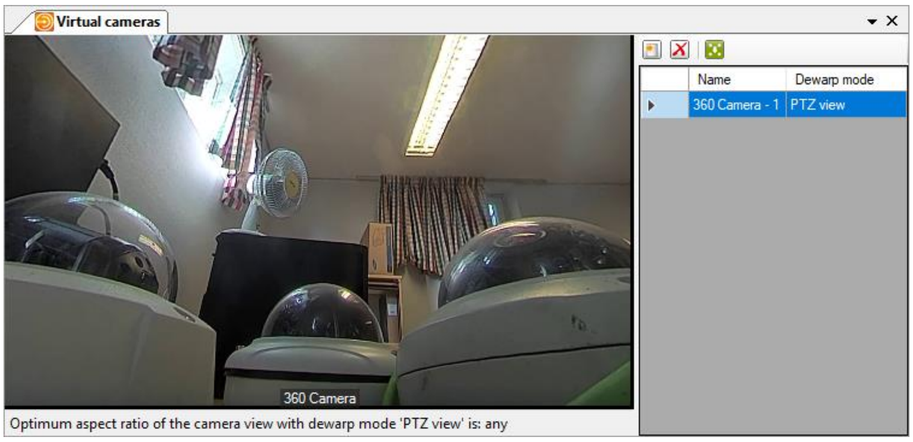
Rename sets the node in rename mode. You can change the camera name directly in the treeview. This is the same as changing the *Local name* in the Used Server camera list.

Delete deselects the camera for this client configuration. This is the same as unchecking the *Use* checkbox in the Used Server Camera list.

Camera panel

Depending on which type of camera you are double-clicking on in the treeview, different panels will open; *Used Server* panel or *Virtual cameras* panel. If you double-click a 360 camera or a virtual camera, the *Virtual cameras* panel will open where you can create and modify virtual cameras with the 360 camera as a source. Note that it is only in the 64-bit version of Admin that the panel *Virtual cameras* will be shown. In the 32.bit version, the panel *Used Server* is always shown, and hence it is not possible to create or modify virtual cameras with this version.

Used Server Virtual cameras panel



Name	Dewarp mode
360 Camera - 1	PTZ view

Optimum aspect ratio of the camera view with dewarp mode 'PTZ view' is: any

Figure 2.507 The Virtual cameras panel.

This panel consists of the image from the 360 camera that is the source of the images in the virtual cameras, dewarped to the current mode. It is possible to create multiple virtual cameras from the same source camera. The virtual cameras can be used like regular cameras when creating views in the client.

In the camera image, you can pan, tilt, and zoom in the image until you get the view you wish for the virtual camera. Note that the view that is shown in the client might differ a little bit from what is shown here due to the camera view in the client might have another aspect ratio.

To change dewarp mode, right-click in the camera image. The following popup menu will be shown. Depending on the mounting position of the camera (*Ceiling, Wall,* or *Ground*), the sub-menu of *Dewarp mode* will contain different options.

The example below shows the menu for a camera that is mounted in position *Ceiling* or *Ground*.

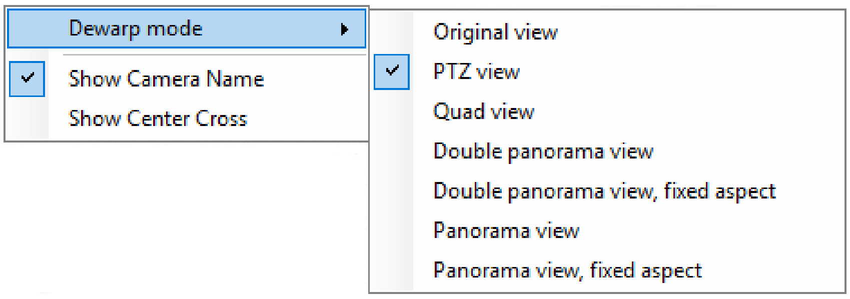
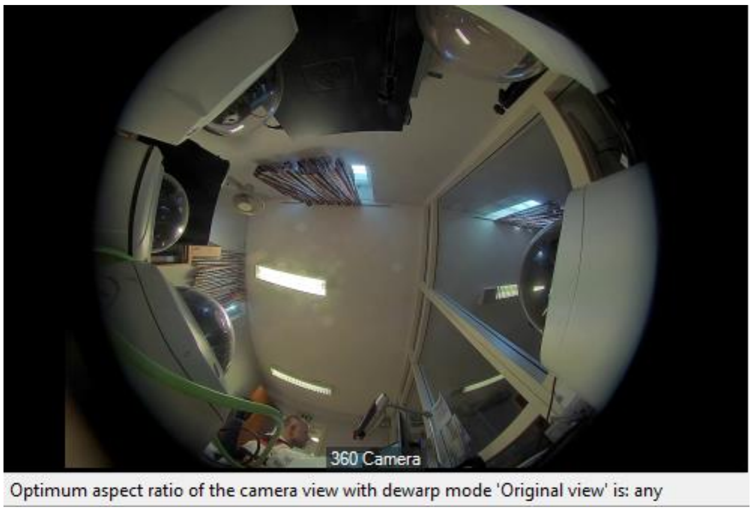


Figure 2.508 The popup menu in the Virtual cameras panel.

Dewarp mode -> Original view shows the original 360 image from the camera.

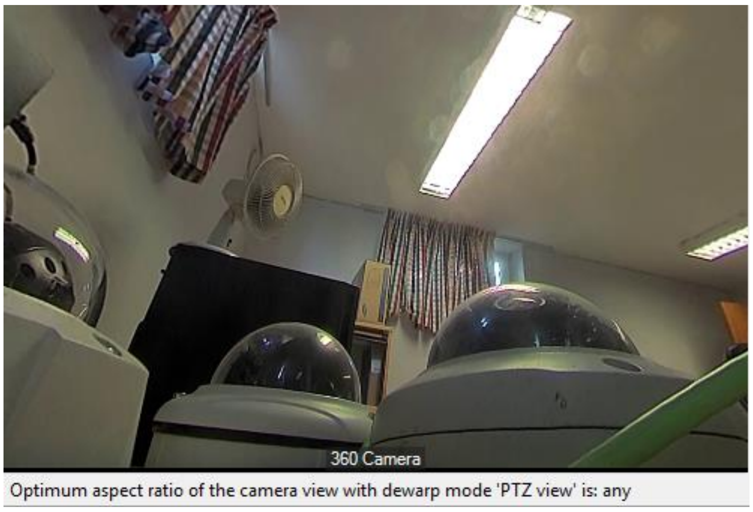


Optimum aspect ratio of the camera view with dewarp mode 'Original view' is: any

Figure 2.509 Original view from a 360 camera.

That the optimum aspect ratio for this dewarp mode is *any* means that the view will look good and that the aspect ratio in the image will remain the same regardless of what aspect ratio the destination camera view in the client has.

Dewarp mode -> PTZ view shows a dewarped view from the camera. It is possible to pan, tilt, and zoom in the image.

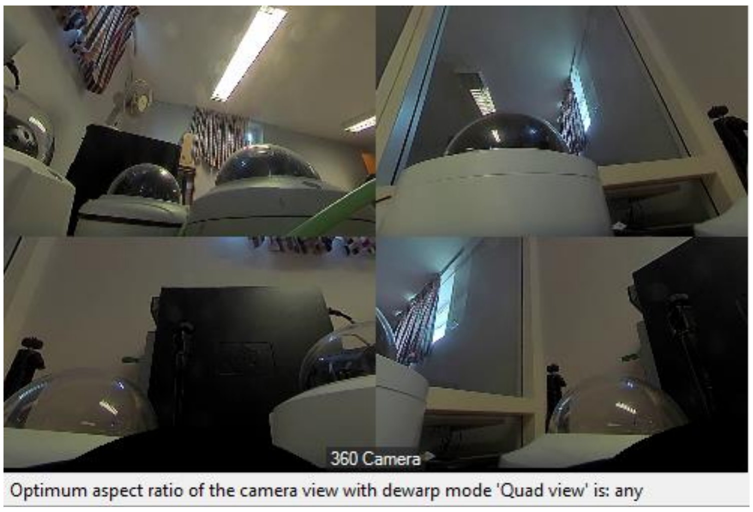


Optimum aspect ratio of the camera view with dewarp mode 'PTZ view' is: any

Figure 2.510 PTZ view from a 360 camera.

That the optimum aspect ratio for this dewarp mode is *any* means that the view will look good and that the aspect ratio in the image will remain the same regardless of what aspect ratio the destination camera view in the client has. It might be, though, that the field of view will differ slightly due to the aspect ratio in the client.

Dewarp mode -> Quad view shows four dewarped views from the camera in the same image. In each and one of the four views, you can pan, tilt, and zoom independently of the other views.

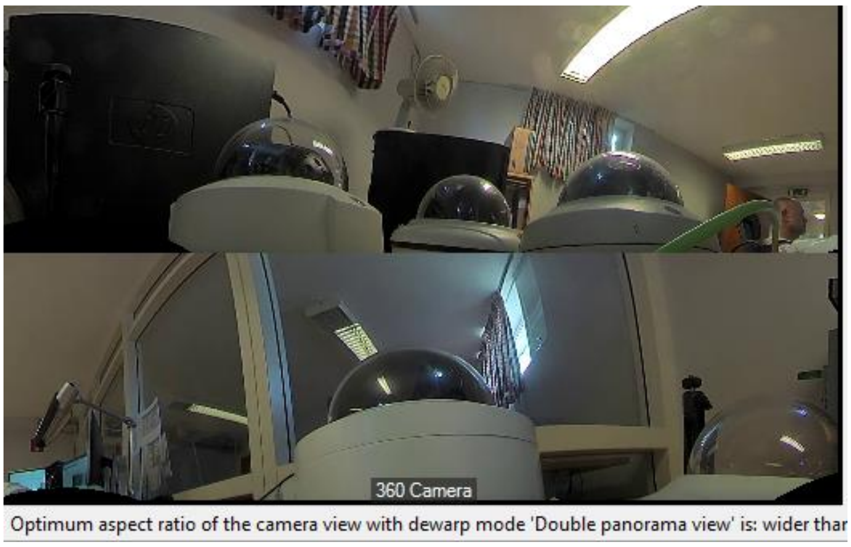


Optimum aspect ratio of the camera view with dewarp mode 'Quad view' is: any

Figure 2.511 Quad view from a 360 camera.

That the optimum aspect ratio for this dewarp mode is *any* means that the view will look good and that the aspect ratio in the image will remain the same regardless of what aspect ratio the destination camera view in the client has. It might be, though, that the field of view will differ slightly due to the aspect ratio in the client.

Dewarp mode -> Double panorama view shows two 180 degree images above each other in one image from the camera. You can pan in the image.

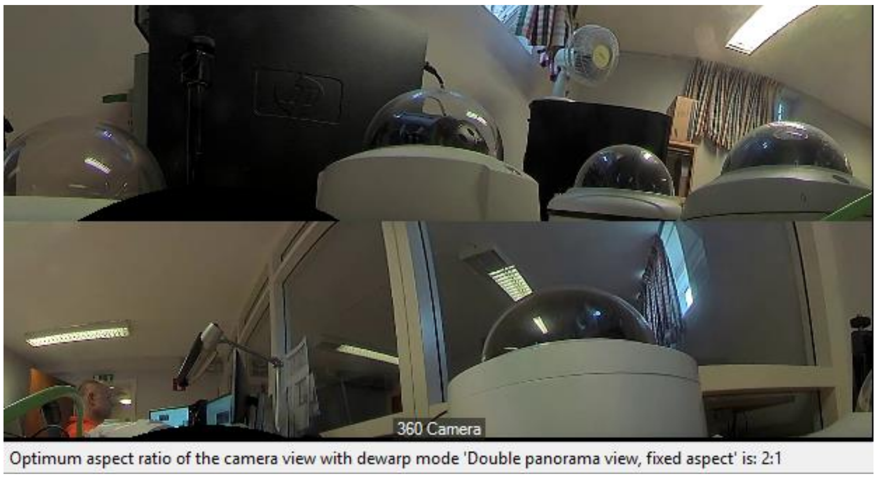


Optimum aspect ratio of the camera view with dewarp mode 'Double panorama view' is: wider thar

Figure 2.512 Double panorama view from a 360 camera.

The optimum aspect ratio for this dewarp mode is *wider than 4:3*. This is to be able to maintain a reasonable correct aspect ratio in the dewarped image.

Dewarp mode -> Double panorama view, fixes aspect shows two 180 degree images above each other in one image from the camera. You can pan in the image.

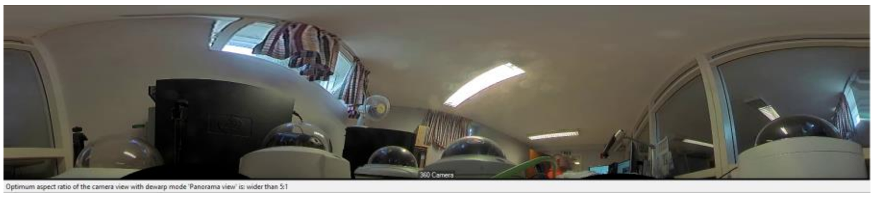


Optimum aspect ratio of the camera view with dewarp mode 'Double panorama view, fixed aspect' is: 2:1

Figure 2.513 Double panorama view, fixed aspect from a 360 camera.

The optimum aspect ratio for this dewarp mode is *2:1*. If the camera view has another aspect ratio, there will be black bars above and under or to the left and right of the image.

Dewarp mode -> Panorama view shows one 360 degree image from the camera. You can pan, tilt, and zoom in the image.

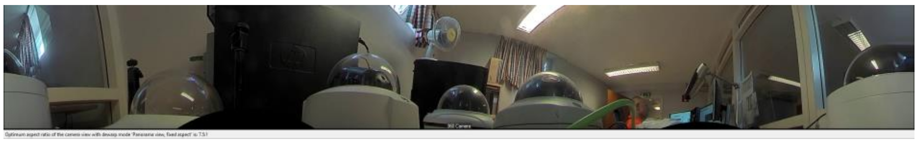


Optimum aspect ratio of the camera view with dewarp mode 'Panorama view' is: wider than 5:1

Figure 2.514 Panorama view from a 360 camera.

The optimum aspect ratio for this dewarp mode is *wider than 5:1*. This is to be able to maintain a reasonable correct aspect ratio in the dewarped image.

Dewarp mode -> Panorama view, fixed aspect shows one 360 degree image from the camera. You can pan, tilt, and zoom in the image.



Optimum aspect ratio of the camera view with dewarp mode 'Panorama view, fixed aspect' is: 7.5:1

Figure 2.515 Panorama view, fixed aspect from a 360 camera.

The optimum aspect ratio for this dewarp mode is *7.5:1*. If the camera view has another aspect ratio, there will be black bars above and under or to the left and right of the image.

The example below shows the menu for a camera that is mounted in position *Wall*.

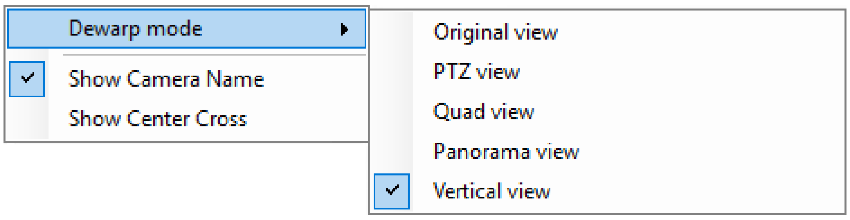


Figure 2.516 The popup menu in the Virtual cameras panel for a wall-mounted camera.

Dewarp mode -> Vertical view shows a standing 180 degree image from the camera. You can pan, tilt, and zoom in the image.

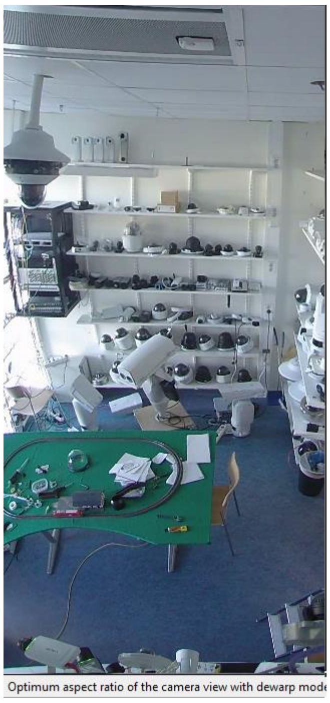


Figure 2.517 Vertical view from a 360 camera.

The optimum aspect ratio for this dewarp mode is *1:2.* If the camera view has another aspect ratio, there will be black bars above and under or to the left and right of the image.

The other dewarp modes of a wall-mounted camera are the same as for a ceiling och ground-mounted camera.

You can easily create views in Ethiris that matches all these types of images.

Virtual cameras panel toolbar

Figure 2.518 The toolbar in the panel Virtual cameras.

Add a new virtual camera	Use this button to add a new virtual camera. Start by right-clicking the camera image and select the dewarp mode you wish. Use the mouse to pan, tilt and zoom the image until you are satisfied. Then create the virtual camera by clicking this toolbar button. The virtual camera will be created with the current dewarp mode and the current PTZ position.
Delete selected virtual camera(s)	Use this button to delete the selected virtual cameras.
Update the virtual camera with the current position	Use this button to change the dewarp mode and/or PTZ position for an existing virtual camera.

2.4.87 Sounds node

Under the node Ethiris Client, there is a *Sounds* node. This node is used for connecting sound files to various events in Ethiris.

Each sound shall have a unique name. Each sound file is connected to an *Activation Signal* and possibly to an *Inactivation signal*. When the activation signal is activated, Ethiris Client plays back the sound file. In this way, any event in Ethiris can be associated with a certain sound.

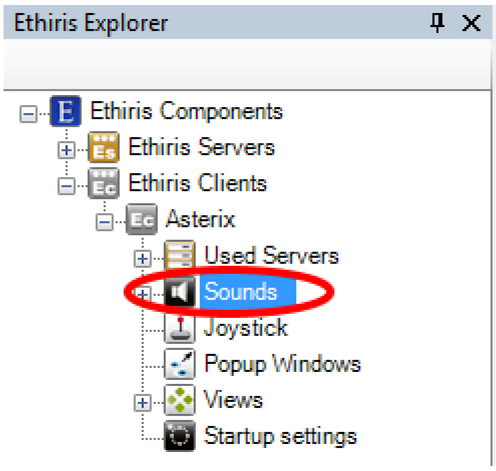


Figure 2.519 The Sounds node in Ethiris Explorer treeview.

Sounds popup menu

Right-click on this node opens a menu.

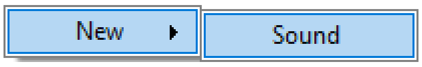


Figure 2.520 The popup menu for the Sounds node in Ethiris Explorer treeview.

New->Sound adds a new sound to the client configuration. It will be immediately visible in the treeview as a new nod. If the panel *Sounds* is open, it will be displayed there also.

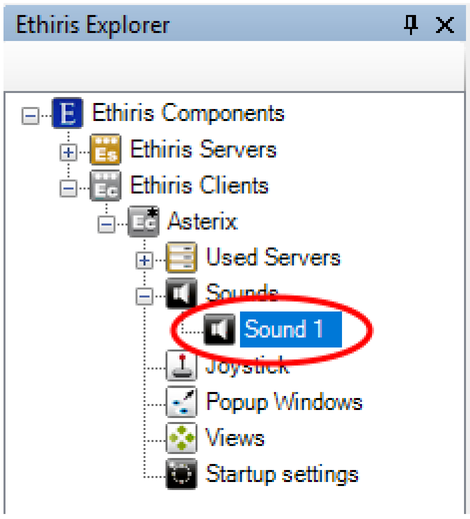
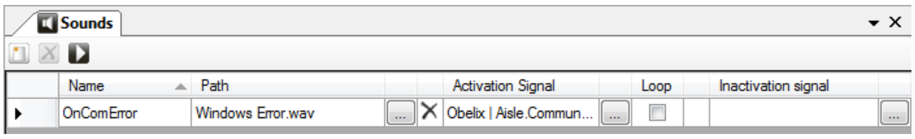


Figure 2.521 New sound added.

Sounds panel

Double-clicking the *Sounds* node in the treeview opens the corresponding panel.



	Name	Path			Activation Signal		Loop	Inactivation signal	
▸	OnComError	Windows Error.wav	...	✕	Obelix | Aisle.Commun...	...	☐		...

Figure 2.522 The Sounds panel.

This panel consists of a list of all defined sounds.

At the top of the panel, there is a toolbar.

Sounds panel toolbar

Figure 2.523 The toolbar in the Sounds panel.

Add a new sound — Use this button to add a new sound.

Delete selected sound(s) — Use this button to delete the selected sound(s) from the configuration. You can select more than one sound by using the *Ctrl*-button and/or the *Shift*-button.

Play selected sound — Use this button to test/play the selected sound file.

Stop all sounds — Use this button to stop the playback of all sound files.

Sounds panel sound list

The sound list consists of several columns.

Name is the name of the sound. The name is only a reference to the sound and is not used for anything in particular for the moment.

Path determines the name and path to the sound file. There is support for sound files of the types *.wav*.

Path Browse button is used for opening a dialog for browsing for sound files.

Activation signal specifies which signal in Ethiris Server's data store will start playback of the sound. This can be any *Boolean*-signal such as *Motion* for a motion detector or *CommunicationError* for a camera or a user-defined signal. When the signal goes active (from 0 -> 1) the playback of the sound is started. Delete the signal by clicking the ✕ button to the left of the signal.

Loop determines if the sound will be played over and over again until the *Inactivation signal* is activated. If *Loop* is not ticked, the sound will be played once, and if loop is selected, the sound will be played repeatedly until the inactivation signal goes from 0->1.

Inactivation signal is not needed if you have not selected *Loop*. If loop is selected, there has to be an inactivation signal for being able to stop playback of the sound. Any *Boolean* in Ethiris Server's data store is available. Delete the signal by clicking the ✕ button to the left of the signal.

2.4.88 Sound node

Under the node *Sounds,* there might be some *Sound* nodes. This node is used for connecting a sound file to events in Ethiris.

Each sound file is connected to an *Activation Signal* and possibly to an *Inactivation signal*. When the activation signal is activated, Ethiris Client plays back the sound file. In this way, any event in Ethiris can be associated with a certain sound.

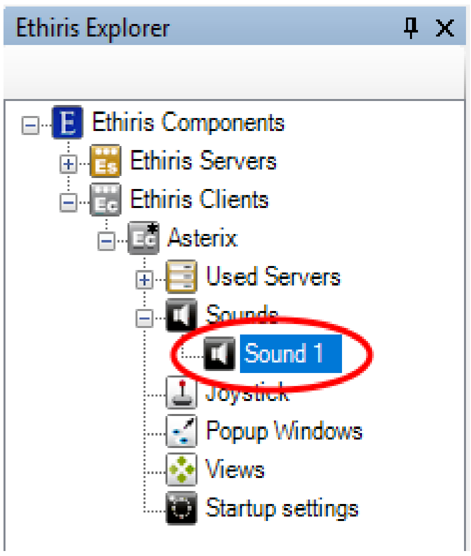


Figure 2.524 A Sound node in Ethiris Explorer treeview.

Sound popup menu

Right-click on this node opens a menu.

Figure 2.525 The popup menu for a Sound node in the Ethiris Explorer treeview.

Delete deletes the sound from the client configuration. It will immediately disappear in the treeview and in the list of sounds in the panel *Sounds*.

Sound panel

Double-clicking the *Sound* node in the treeview opens a panel that is the same as the panel *Sounds*.

2.4.89 Joystick node

Under the Ethiris Client node, there is a *Joystick* node. This node is used for configuring a joystick that is connected to the client computer.

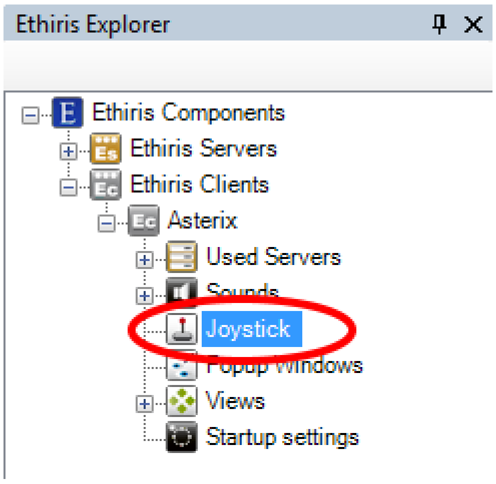


Figure 2.526 The joystick node in Ethiris Explorer treeview.

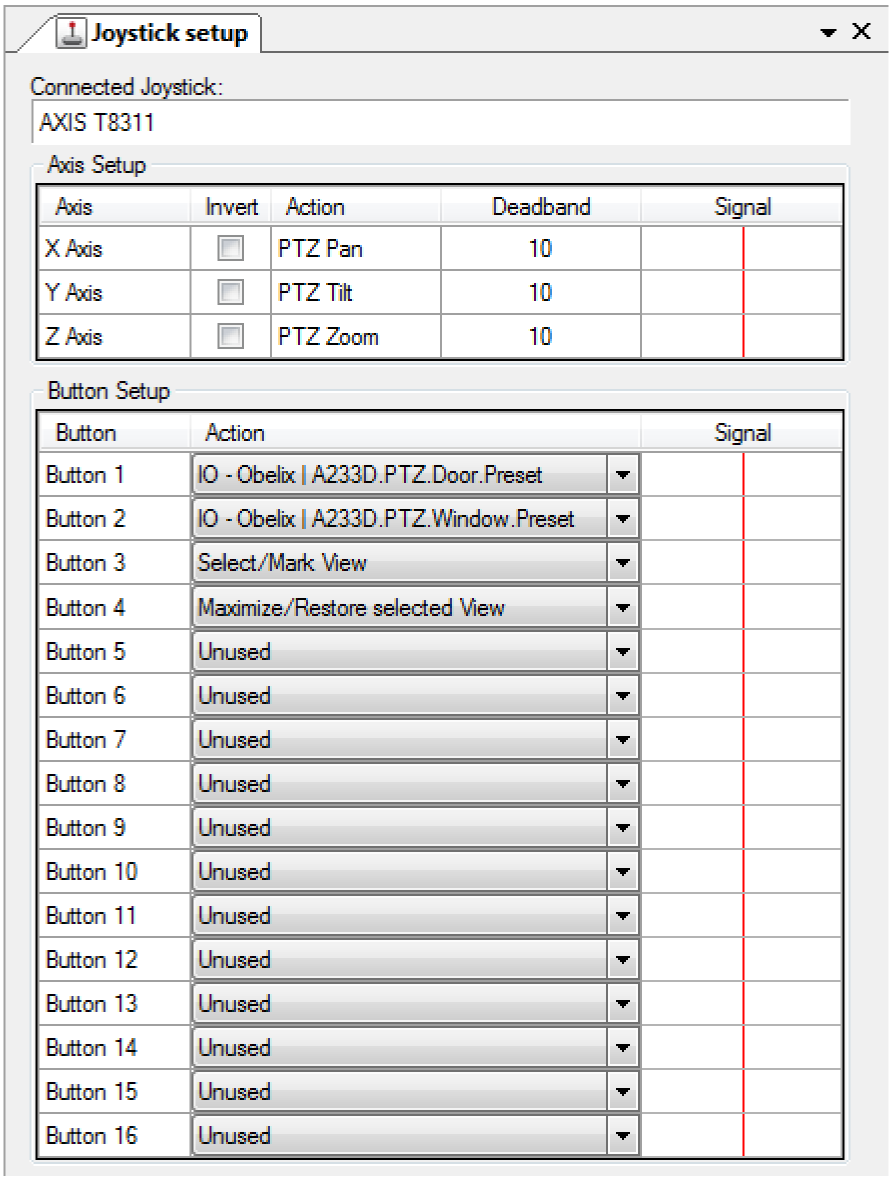
Joystick popup menu

There is no popup menu for this node.

Joystick panel

Double-clicking the *Joystick* node in the treeview opens the corresponding panel.

Note that if no joystick is connected to the current computer, no panel is opened. The available settings depend on the type of joystick to use. In the example below, an Axis T8311 joystick is used.



Connected Joystick: AXIS T8311

Axis Setup

Axis	Invert	Action	Deadband	Signal
X Axis	☐	PTZ Pan	10	
Y Axis	☐	PTZ Tilt	10	
Z Axis	☐	PTZ Zoom	10	

Button Setup

Button	Action	Signal
Button 1	IO - Obelix | A233D.PTZ.Door.Preset	
Button 2	IO - Obelix | A233D.PTZ.Window.Preset	
Button 3	Select/Mark View	
Button 4	Maximize/Restore selected View	
Button 5	Unused	
Button 6	Unused	
Button 7	Unused	
Button 8	Unused	
Button 9	Unused	
Button 10	Unused	
Button 11	Unused	
Button 12	Unused	
Button 13	Unused	
Button 14	Unused	
Button 15	Unused	
Button 16	Unused	

Figure 2.527 The Joystick panel.

This panel consists of two main sections; *Axis setup* and *Button setup*.

At the top of the panel, the model name of the currently connected joystick is displayed.

Joystick Axes setup

The purpose of this section is to adjust the joystick axes deadband and possibly invert them.

The list contains the three axes for *X*, *Y* & *Z*. There are several columns for each axis.

Axis indicates which axis it is. The X axis is used for pan, Y axis for tilt, and Z axis for zoom.

Invert can be used for inverting the axis. If, for example, the X axis is inverted, a camera would go right when the joystick is pulled to the left. This can be convenient if the camera is mounted upside down.

Action just informs about what PTZ function the axis is connected to.

Deadband is a value between 0 – 100. The default is 10. It determines when the joystick is considered to be in the center, i.e., the current value of an axis is 0. Some joysticks are not completely stable, meaning that even if you don't pull the joystick at all, it may still give a small value for some of the axes. The effect will then be that the PTZ camera currently in focus will move even if you don't move the joystick. To avoid this problem, you can set a deadband high enough to simply ignore small values from the joystick. The higher deadband value, the more you have to pull the joystick to begin moving the camera.

Signal indicates the current value of the axes. This is helpful both for learning which axes do what and for setting the deadband value. When the joystick is untouched, there should be no signal on any axis.

Joystick Button setup

The purpose of this section is to connect buttons on the joystick unit to various Ethiris functions.

The list contains 128 buttons. Not all buttons may be present on the actual joystick. Axis T8311, for example, has 6 buttons. There are three columns for each button.

Button indicates which button it is.

Action determines what will happen when the corresponding button is pressed. There are several options:

Unused – The button is not used by Ethiris.

Select/Mark view – This is the same as pressing the *Space bar* in Ethiris Client. It is a toggle function where the camera view in focus will toggle between *selected* and *marked*. When a camera view is marked (indicated by a dotted green frame), you can move focus to another camera view in the same view by pulling the joystick left/right/up/down. You can also use the arrow keys on the keyboard to move focus. When a camera view is marked, you can select it by pressing this joystick button or the space bar on the keyboard. A selected camera view is indicated by a green frame. When a camera view is selected, you can maneuver (Pan, Tilt & Zoom) the camera with the joystick, or by the arrow keys on the keyboard.

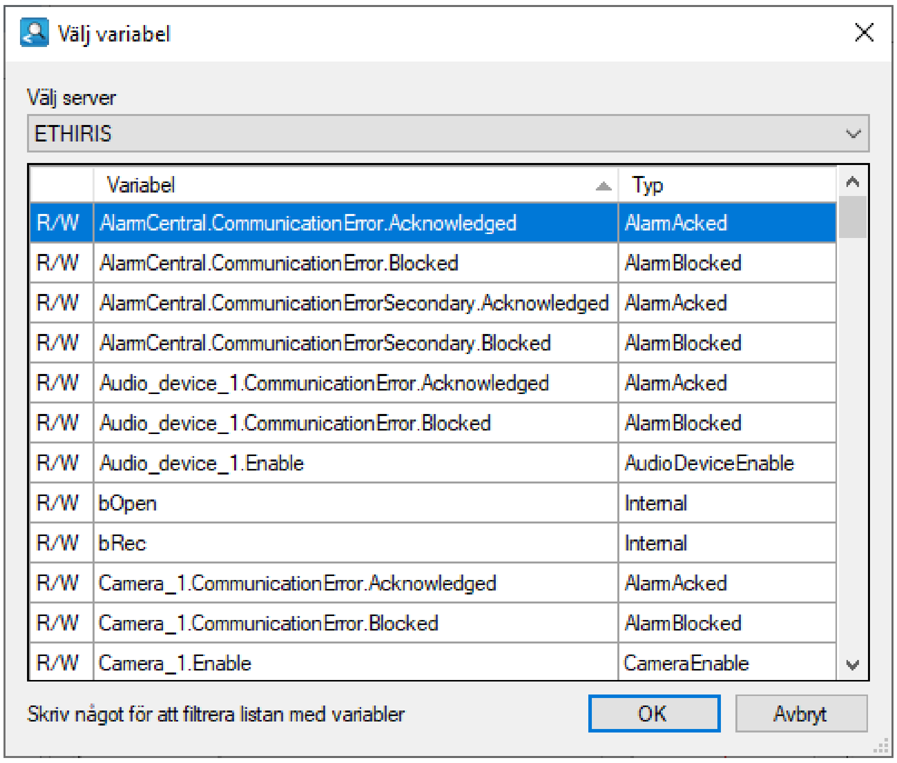
Maximize/Restore selected view – This is the same as double-clicking the camera view with the mouse. For example, in a 4-split view, if you maximize one of the four camera views, this camera only will be displayed in the whole window. Restoring the view will display all four cameras again.

Move marker to next window – If you have more than one live window, this function can be used for moving focus to the first camera view in the next window.

Move marker to previous window – If you have more than one live window, this function can be used for moving focus to the first camera view in the previous window.

IO – This is a very powerful function. You can connect the joystick button to any Boolean signal in the Ethiris Servers data store. When the button is pressed, the connected signal is activated (set to true). The options here are almost endless. The signal can be a direct function such as *RecordEvent* on a camera or a *Preset* on a PTZ camera. The signal can also be an internal variable in Ethiris Server's data store which in turn can be used for activating a piece of script code. Other options are to activate a popup live window in an Ethiris Client.

When you select the *IO* function for a joystick button, a dialog for selecting an I/O-signal is displayed.


Välj variabel

Välj server: ETHIRIS

	Variabel	Typ
R/W	AlarmCentral.CommunicationError.Acknowledged	AlarmAcked
R/W	AlarmCentral.CommunicationError.Blocked	AlarmBlocked
R/W	AlarmCentral.CommunicationErrorSecondary.Acknowledged	AlarmAcked
R/W	AlarmCentral.CommunicationErrorSecondary.Blocked	AlarmBlocked
R/W	Audio_device_1.CommunicationError.Acknowledged	AlarmAcked
R/W	Audio_device_1.CommunicationError.Blocked	AlarmBlocked
R/W	Audio_device_1.Enable	AudioDeviceEnable
R/W	bOpen	Internal
R/W	bRec	Internal
R/W	Camera_1.CommunicationError.Acknowledged	AlarmAcked
R/W	Camera_1.CommunicationError.Blocked	AlarmBlocked
R/W	Camera_1.Enable	CameraEnable

Skriv något för att filtrera listan med variabler

Figure 2.528 Dialog for browsing for I/O signals.

Select the desired signal and click *OK*. Note that only *writeable* variables make sense in this context. Writeable variables have a *W* or *R/W* in the first column in the list.

Signal indicates if the corresponding button is pressed.

2.4.90 Popup Windows node

Under the Ethiris Client node, there is a *Popup Windows* node. This node is used for configuring pre-defined popup live windows for Ethiris Client.

You give the popup window a name and determine which monitor it will popup on, the size and position of the window. In Ethiris Client, you can open any of the defined popup windows and display live video from cameras selected in the *Cameras* panel or by right-clicking a view button and select the desired popup window on the menu. You can also configure a view such that the view will popup in one of the popup windows when activated by an I/O or a hot spot.

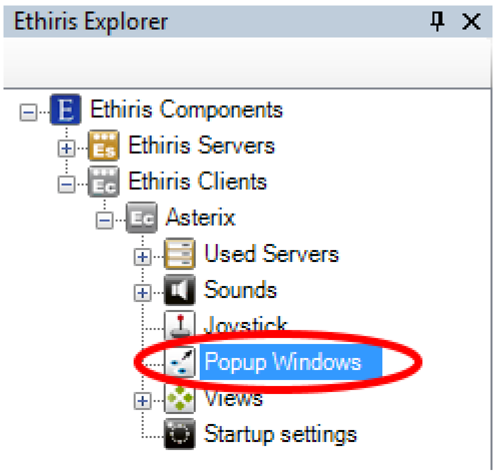


Figure 2.529 The Popup Windows node in Ethiris Explorer treeview.

Popup windows popup menu

Right-click on this node opens a menu.

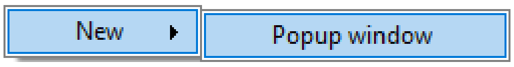
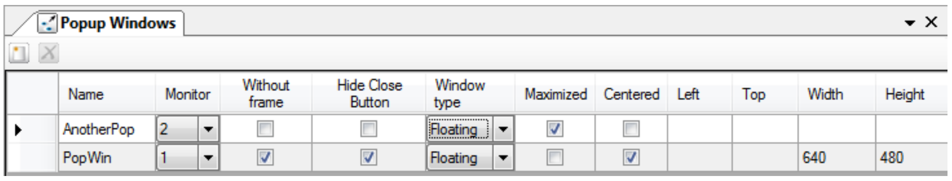


Figure 2.530 The popup menu for the Popup Windows node in Ethiris Explorer treeview.

New->Popup window adds a new popup window to the client configuration. If the panel *popup Windows* is open, it will be displayed there also.

Popup Windows panel

Double-clicking the *Popup Windows* node in the treeview opens the corresponding panel.



Name	Monitor	Without frame	Hide Close Button	Window type	Maximized	Centered	Left	Top	Width	Height
AnotherPop	2	☐	☐	Floating	✓	☐				
PopWin	1	✓	✓	Floating	☐	✓			640	480

Figure 2.531 The Popup Windows panel.

This panel consists of a list of all currently defined popup windows.

At the top of the panel, there is a toolbar.

Popup Windows panel toolbar

Figure 2.532 The toolbar in the Popup Windows panel.

Add new popup window	Use this button to add a new popup window.
Delete selected popup window(s)	Use this button to delete the selected popup window(s) from the configuration. You can select more than one popup window by using the *Ctrl*-button and/or the *Shift*-button.

Popup Windows panel window list

The window list consists of several columns.

Name is the name for the window. This name is used as a reference for the popup window in various contexts.

Monitor determines on which monitor the window will be displayed when activated. It can be between 1 – 8.

Without frame determines if the window will have a frame or not. This property can be combined with other properties such as any window that can be without a frame, not just maximized windows. When combined with maximized, the effect is that the whole screen is filled with live images, just like an old-fashioned CCTV monitor.

Hide Close Button determines if the window can be manually closed by clicking the close button at the top right corner of the window. If this function is selected, the window can only be closed by an automatic inactivation or by closing the whole Ethiris Client.

Windows type. Determines how the window will be shown. The options are *Floating*, *docked left,* and *docked right*. When window type is *docked left* or *docked right* neither *Maximized, Centered*, *Top*, *Left*, *Height* nor *Width* can be defined

Maximized*.* When checked, the popup window will be maximized on the selected monitor when activated. When maximized is selected, neither *Centered*, *Top*, *Left*, *Height* nor *Width* can be defined.

Centered*.* When checked, the popup window will be centered on the selected monitor. When *Centered* is selected, neither *Top* nor *Left* can be defined.

Left is the desired left position of the window when activated.

Top is the desired top position of the window when activated.

Width is the desired width of the window when activated.

Height is the desired height of the window when activated.

2.4.91 Views node

Under the Ethiris Client node, there is a *Views* node. This is a collection node for all *Sections*, *Views,* and *Buttons* in the client configuration.

Sections provide a way of dividing the system into smaller parts. Each section will appear in the *Sections Explorer* tool window in Ethiris Client. See the *Ethiris Client User´s Guide* for more information. Each section can contain many *views*, *buttons,* and, for that matter, even other *sections* (subsections).

Views are pre-defined views with one or several *camera views*. A camera view is usually used for displaying live video from a pre-defined camera. There are other options for a camera view as well, which we will look into later on.

Buttons are used for activating signals/variables in the Ethiris Servers data store. Examples are activation of a preset position or a guard tour.

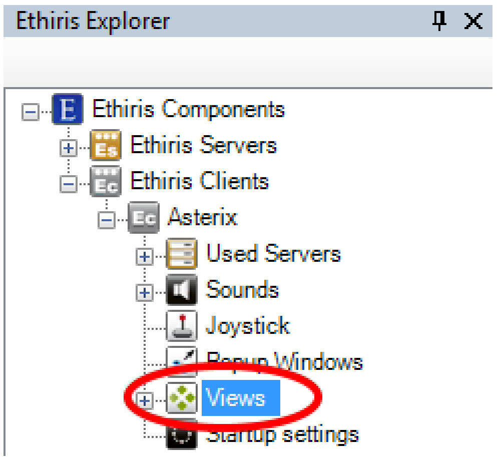


Figure 2.533 The Views node in Ethiris Explorer treeview.

Views popup menu

Right-clicking this node brings up a context menu.

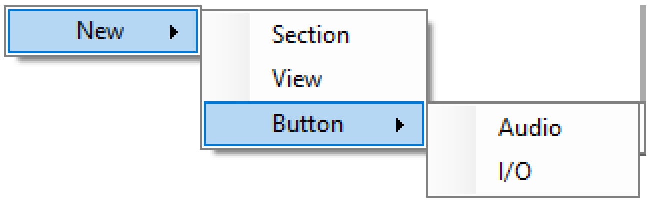


Figure 2.534 The popup menu for the Views node.

New->Section adds a new section to the client configuration. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the section directly in the treeview.

New->View adds a new view to the client configuration. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the view directly in the treeview. When you add a view at this level, the view will be created in the *Root* section. This section is created automatically when necessary.

The purpose of a view is to pre-configure a layout of camera views. The view can be displayed in Ethiris Client in live views and the player for recorded video.

New->Button->Audio adds a new audio button to the client configuration. It is immediately added to the treeview, and the new node is set in *rename* mode.

You can enter the desired name of the button directly in the treeview. When you add a view at this level, the view will be created in the *Root* section. This section is created automatically when necessary.

The purpose of an audio button is to connect it to one or more audio devices. When you click the audio button in Ethiris Client, the microphone will activate, and what you say will be sent to all connected audio devices in parallel.

New->Button->I/O adds a new I/O button to the client configuration. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the button directly in the treeview. When you add a view at this level, the view will be created in the *Root* section. This section is created automatically when necessary.

The purpose of an I/O button is to connect it to a writeable Boolean variable in Ethiris Server's data store. When you click the button in Ethiris Client, the corresponding variable is activated. You can use this functionality for many different purposes, e.g., activate a preset position for a PTZ camera, start recording, or sending an email.

Views panel

There is no panel for the Views node.

2.4.92 Section node

Under the Views node, there may be one or several *Section* nodes.

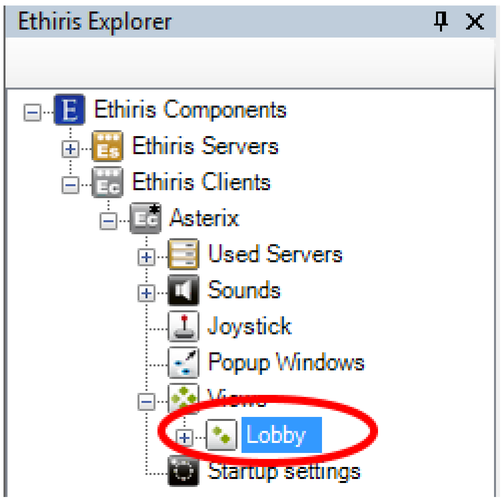


Figure 2.535 A Section node in Ethiris Explorer treeview.

Section popup menu

Right-clicking this node brings up a context menu.

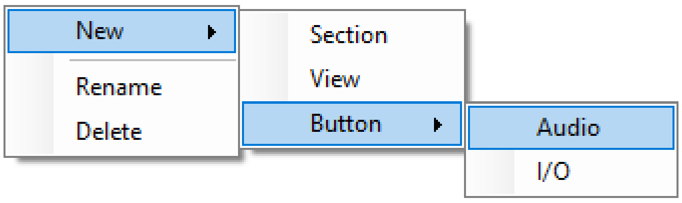


Figure 2.536 The popup menu for a Section node.

New->Section adds a new section to the client configuration. The new section will be a subsection to the section you have right-clicked. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the section directly in the treeview.

New->View adds a new view to the client configuration in the section you have right-clicked. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the view directly in the treeview.

New->Button->Audio adds a new audio button to the client configuration in the section you have right-clicked. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the button directly in the treeview.

New->Button->I/O adds a new I/O button to the client configuration in the section you have right-clicked. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the button directly in the treeview.

Rename sets the node in *rename* mode. You can enter the new name directly in the treeview.

Delete removes the section from the client configuration. If the section has any content, such as views and buttons, a confirmation dialog will be displayed before deletion.

Section panel

There is no panel for the Section node.

Section Copy/Move

By *dragging* a Section node in the treeview, you can change the order of sections.

You *drag* by clicking the left mouse button on the desired section in the treeview, move the mouse pointer while holding the left mouse button down, and then release the mouse button when the mouse pointer is in the desired position.

By pressing the *Ctrl* key at the same time as you release the mouse button, the section, and its content, such as possible subsections, views, and buttons, will be copied.

To move/copy a section into another section and thus making it a subsection, press the *Shift* key at the same time as you release the mouse button when the mouse pointer is above the desired section in the treeview.

When you move a section down in the treeview, the section will be positioned *after* the section you drop on.

When you move a section up in the treeview, the section will be positioned *before* the section you drop on.

2.4.93 View node

Under the Views node and possibly under any Section node, there may be one or several *View* nodes.

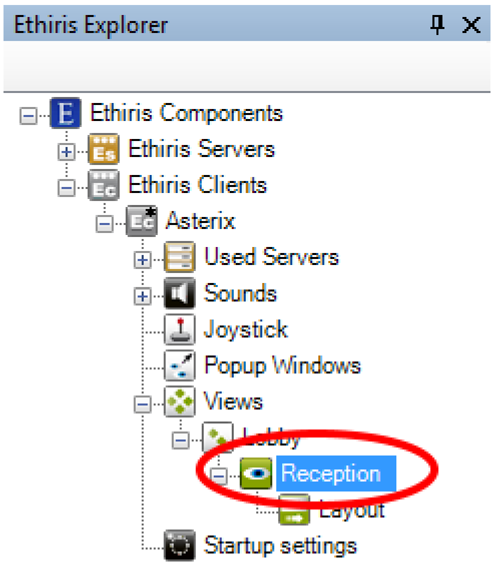


Figure 2.537 A View node in Ethiris Explorer treeview.

View popup menu

Right-clicking this node brings up a context menu.

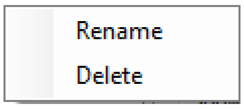


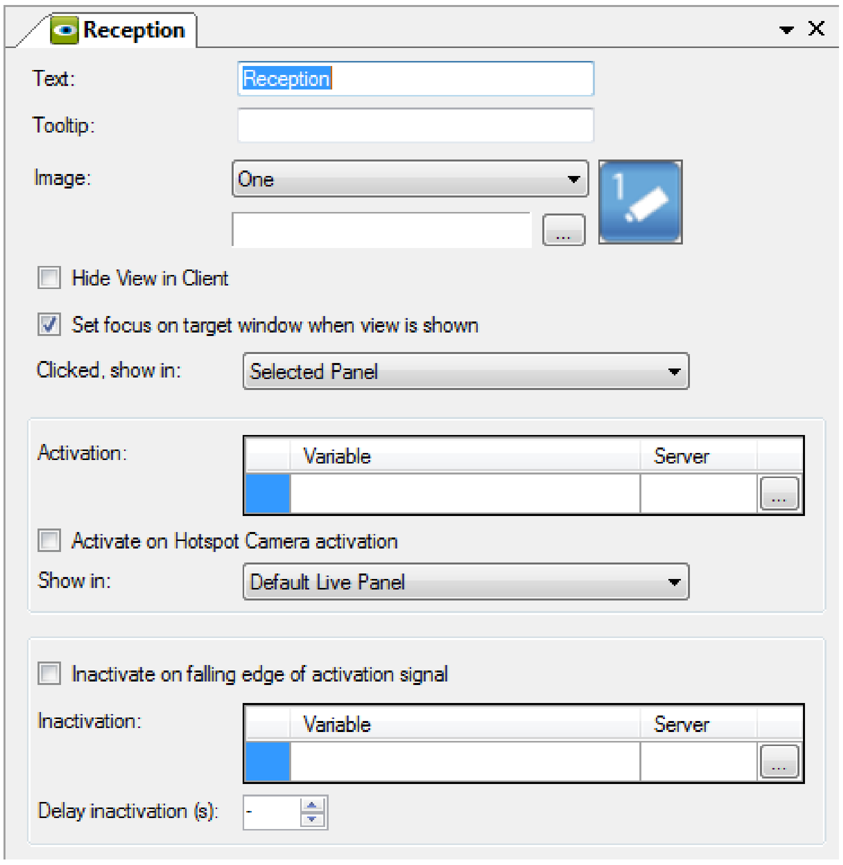
Figure 2.538 The popup menu for a View node.

Rename sets the node in *rename* mode. You can enter the new name for the view directly in the treeview.

Delete removes the view from the client configuration. A confirmation dialog will be displayed before deletion.

View panel

Double-clicking a *View* node in the treeview opens the corresponding panel.


Reception

Text: Reception

Tooltip:

Image: One

☐ Hide View in Client

☑ Set focus on target window when view is shown

Clicked, show in: Selected Panel

Activation:

Variable	Server

☐ Activate on Hotspot Camera activation

Show in: Default Live Panel

☐ Inactivate on falling edge of activation signal

Inactivation:

Variable	Server

Delay inactivation (s): -

Figure 2.539 The View panel.

This panel contains properties for the view such as text and image for the corresponding view button that will be displayed in Ethiris Client in the *Views Explorer* and the *Section Explorer* tool windows.

There are also properties for activation and inactivation of the view besides clicking the view button in Ethiris Client.

Text determines which text will be displayed in the corresponding view button in Ethiris Client.

Tooltip is a text that will be displayed in a tooltip when holding the mouse pointer above the view button in the *Views Explorer* tool window in Ethiris Client. This can be used to explain the purpose of a view.

Image determines which image will be displayed in the view button in the Ethiris Client. There are two main options; either you use one of the standard images that are provided with Ethiris, or you use your own image.

The following standard images are available:

One

Two

Three

Four

3x2 *Six*

4x2 *Eight*

3x3 *Nine*

4x3 *Twelve*

5x3 *Fifteen*

4x4 *Sixteen*

5x4 *Twenty*

5x5 *Twentyfive*

Free

I/O *IO*

Audio

When you use your own image, the following formats are supported: *bmp*, *gif*, *jpg*, *png,* and *ico*. For the best result, use an image with the resolution *32 x 32 pixels*.

Hide View in Client. When checked, there will be no corresponding view button in Ethiris Client. The purpose is to, in some way, display the view automatically. This can be a hot spot function when clicking a camera view or a camera symbol on a map. It can be when a certain signal in Ethiris Server is activated, e.g., motion detection or a digital input that is activated when somebody pushes a doorbell.

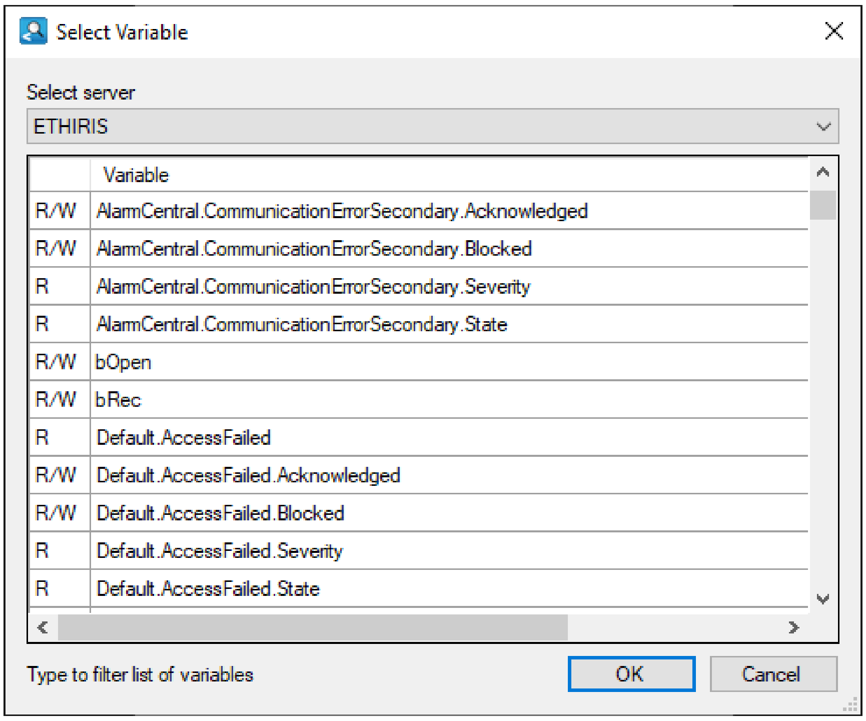
Set focus on target window when view is shown. When checked, focus in Ethiris Client will move to the target window where the view will be displayed. This is the default setting. In some cases, you don't want the focus to be moved from the window currently having focus. Then you check this box.

Show frame marker (green rectangle). When checked, the green frame marker will be visible in Ethiris Client to highlight which camera view is in focus. By default, this is active.

Clicked, show in determines where the view will be displayed when the operator clicks the corresponding button in Ethiris Client. As a default, the *Selected Panel* is selected. That means that the live window that happens to have focus when you click the view button will load the view. You also can hard code the button to *Default Live Panel* or one of the popup windows you have defined. In this case, the button will be marked with a small black square in Ethiris Client.

Activation This frame is for activating the view without any operator clicking the corresponding view button. The first choice is to select a variable for activation. That is, when the variable goes active (from 0 -> 1 or from *false* to *true*), the view will be displayed.

When you browse for an activation variable, a dialog for selecting it is displayed.



Select Variable

Select server: ETHIRIS

	Variable
R/W	AlarmCentral.CommunicationErrorSecondary.Acknowledged
R/W	AlarmCentral.CommunicationErrorSecondary.Blocked
R	AlarmCentral.CommunicationErrorSecondary.Severity
R	AlarmCentral.CommunicationErrorSecondary.State
R/W	bOpen
R/W	bRec
R	Default.AccessFailed
R/W	Default.AccessFailed.Acknowledged
R/W	Default.AccessFailed.Blocked
R	Default.AccessFailed.Severity
R	Default.AccessFailed.State

Type to filter list of variables

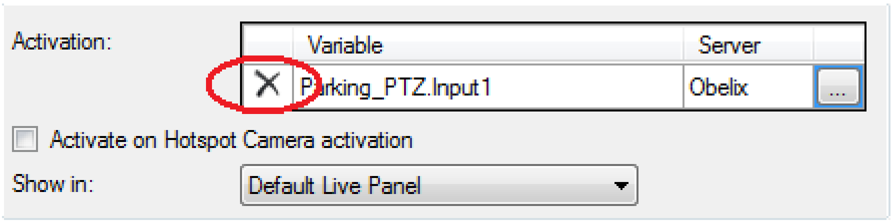
Figure 2.540 Dialog for browsing for variables in Ethiris Server's data store.

In the list, only *readable Boolean* variables are displayed. Readable variables have an *R* in the first column of the list.

To make it easier to find the variable you want, it is possible to filter the list. Just start typing a part of the name of the Variable you want to find, when no input has been made for 700ms, the filter will be applied to the list. To clear the filter, press the Delete button.

Select the desired variable and click *OK*.

If you change your mind, you can remove the variable by clicking the delete button to the left of the variable.



Activation:

	Variable	Server
✕	Parking_PTZ.Input1	Obelix

☐ Activate on Hotspot Camera activation

Show in: Default Live Panel

Figure 2.541 The delete button for removing a variable reference.

Activate on Hotspot Camera activation. When checked, the view will be displayed when a hotspot camera view in this view is activated. A hotspot camera view contains a list of cameras that can be activated by clicking on a camera view displaying live video from one of the cameras or by clicking a camera symbol representing one of the cameras in the hotspot list. For more information on hotspot camera views, see section Larger camera views on page 2:359.

Show in determines in what window the view will be displayed when activated. As a default, the *Default Live Panel* is selected, but you can choose any of the defined *Popup windows* as well.

Inactivate on falling edge of activation signal*.* When checked, the view will be closed (in case of a popup window) when the variable that is specified for *Activation* goes inactive (from 1 -> 0 or from *true* to *false*). If the view were displayed in the *Default Live panel,* the first view in the first section would be displayed instead of this view when the view is inactivated.

Inactivation variable is used for having a specific variable for inactivation of the view. This is an alternative to using the same variable for both activation and inactivation. The browsing for an inactivation variable is similar to browsing for an activation variable, as described above.

Delay Inactivation (s) is used to define a delay, in seconds, until the view will inactivate after the inactivation criterion is fulfilled.

2.4.94 Layout node

Under each View node, there is a corresponding *Layout* node.

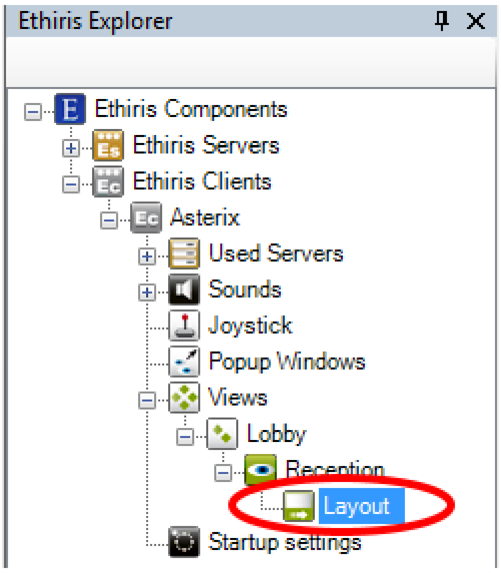


Figure 2.542 A Layout node in Ethiris Explorer treeview.

Layout popup menu

There is no popup menu for this node.

Layout panel

Double-clicking a *Layout* node in the treeview opens the corresponding panel.

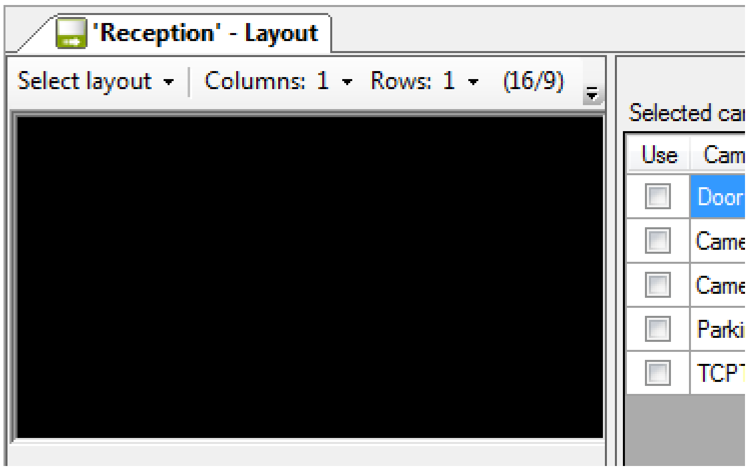


Figure 2.543 The Layout panel.

This panel is divided into two parts; left and right. On the left side is a representation of a monitor/screen where the view will be displayed, and on the right side are properties for the currently selected camera view on the left side.

Select layout is a list of pre-defined layouts from which you can choose.

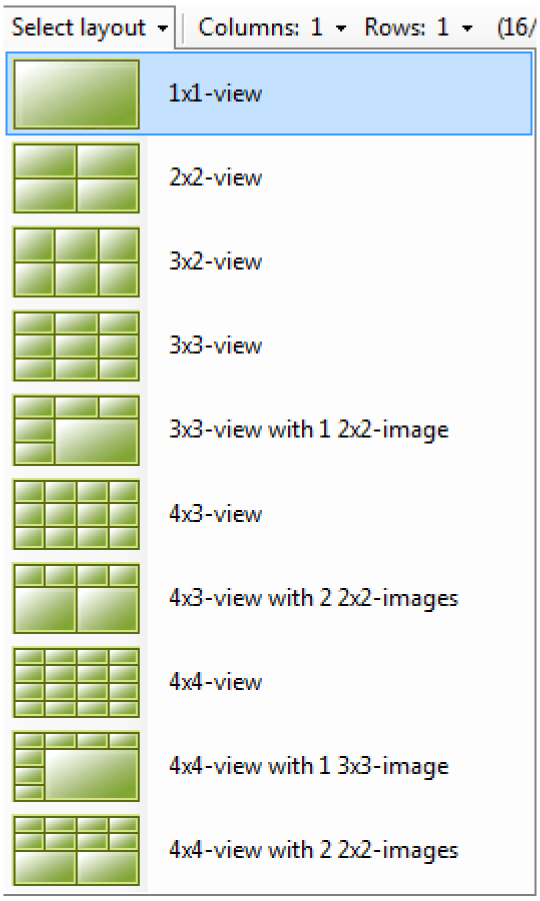

Select layout
Columns: 1
Rows: 1
(16/
1x1-view
2x2-view
3x2-view
3x3-view
3x3-view with 1 2x2-image
4x3-view
4x3-view with 2 2x2-images
4x4-view
4x4-view with 1 3x3-image
4x4-view with 2 2x2-images

Figure 2.544 Pre-defined layouts.

This is a quick way to create a layout. If no one of the pre-defined layouts suits your needs, you can still manually determine the number of columns and rows between 1 – 8. You can even start with a pre-defined layout, which you then can alter as you wish.

Cols determines the number of columns in the view. Each cell in the view is called a *camera view*. You can choose between 1 – 8 columns for each view. In the example below, *3* columns are selected (with still only 1 row).

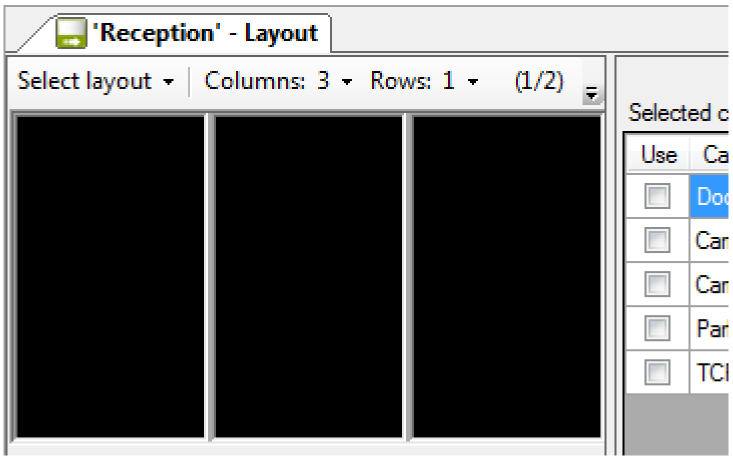


Figure 2.545 The Layout panel with 3 columns.

Rows determine the number of rows in the view. You can choose between 1 – 8 rows for each view. In the example below, *2* rows are selected (with still 3 columns).

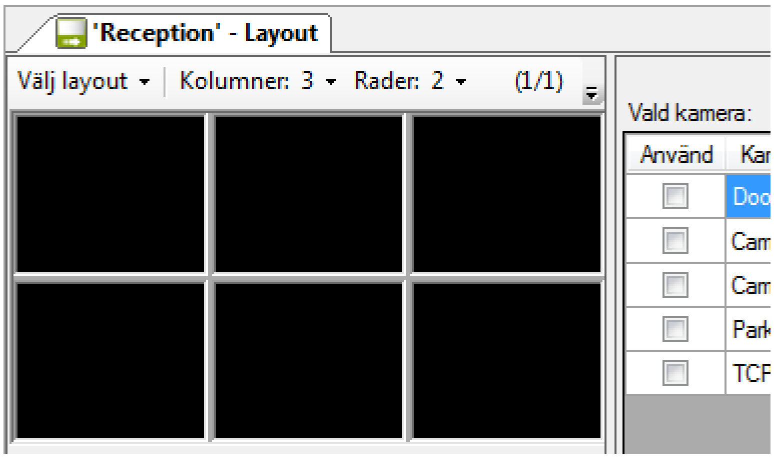


Figure 2.546 The Layout panel with 3 columns and 2 rows.

Current aspect ratio is information about the current aspect ratio of the camera views. In the example below, the aspect ratio is ringed in. *1/1* means that each camera view has (about) the same width as height.

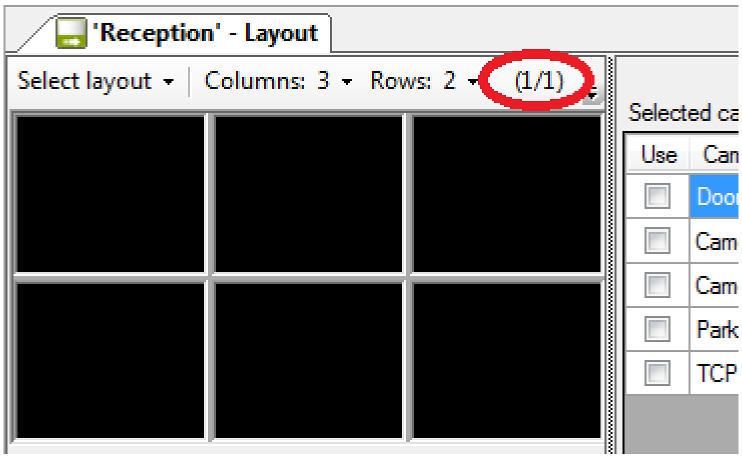


Figure 2.547 The Current aspect ratio is 1/1 for the camera views.

In the next example with 3 x 3 camera views, the aspect ratio has changed to *16/9*, meaning that each camera view is almost twice as wide as high. Width/Height is 16/9.

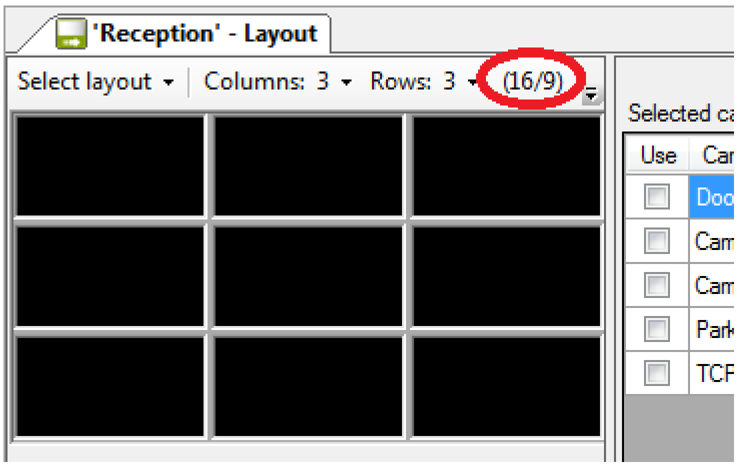


Figure 2.548 The current aspect ratio is 16/9 for the camera views.

Screen type is a help for getting a feel of how the view will look like. You can choose between three different screen formats; *16/9*, *16/10,* and *4/3*. When changing the screen type, the format of the black area changes to reflect the current setting.

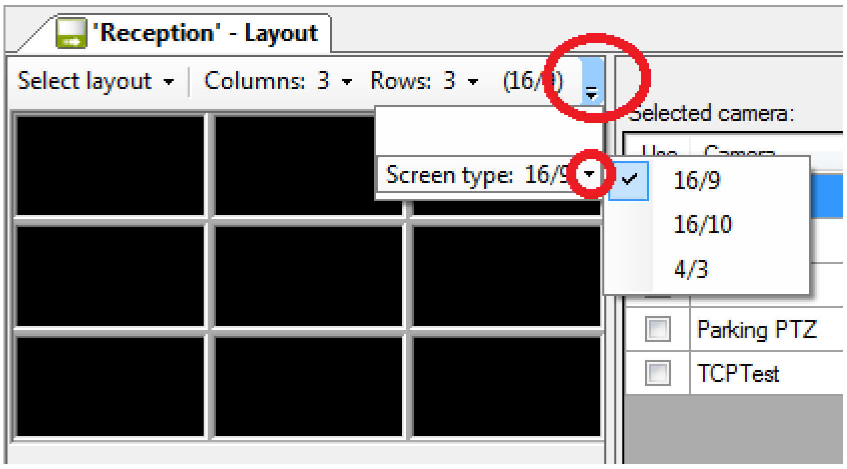


Figure 2.549 Select Screen type by clicking the ringed in arrow.

Larger camera views

You can combine several original camera views into one larger camera view. In the example below, the four camera views in the bottom right of the view have been combined into one larger camera view.

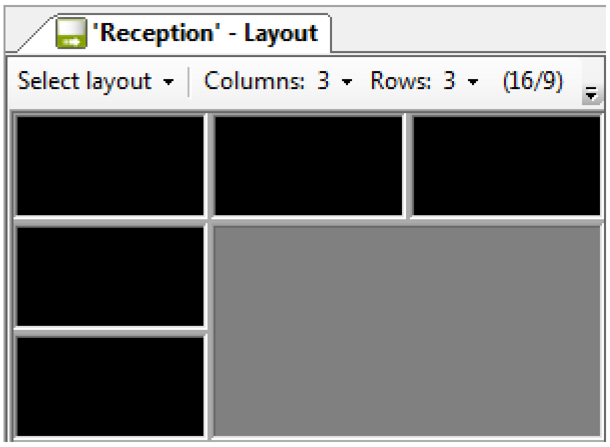


Figure 2.550 Four camera views combined into one larger camera view.

To accomplish this, you first select the camera view in the bottom right corner. See *Figure 2.551* below.

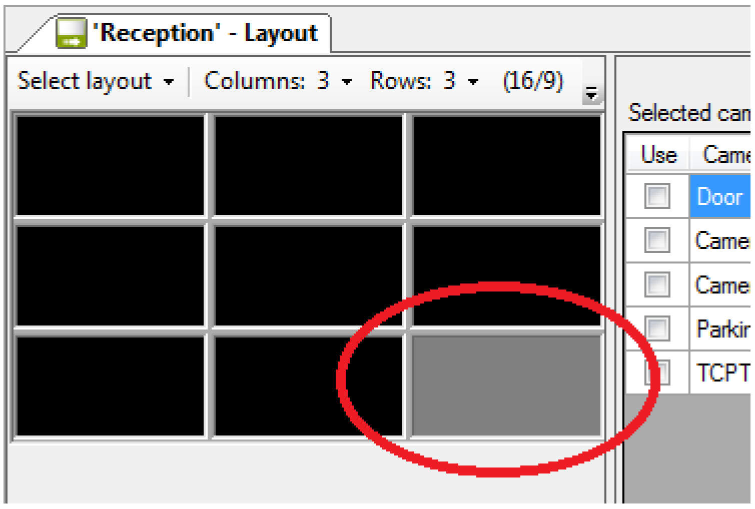


Figure 2.551 First step, select a camera view.

The next step is to click-and-hold the left mouse button in the selected camera view. Then move the mouse pointer while still holding the left mouse button down. Move the mouse pointer to the middle camera view and release the mouse button.

As you move the mouse over the camera views, the new larger camera view is indicated by grey color.

In *Figure 2.552* below, the mouse pointer has been moved up, resulting in a camera view spanning two original camera views. To get the desired camera view spanning 2x 2 original camera views, move the mouse pointer to the left.

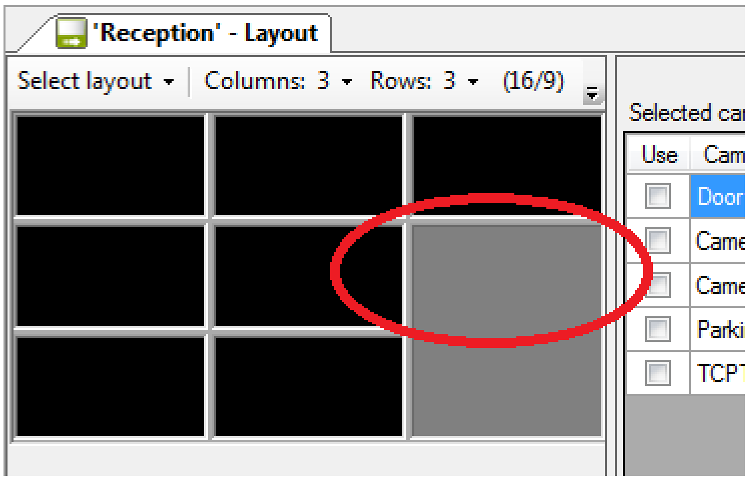


Figure 2.552 Second step, move the mouse pointer to another camera view.

Release the mouse button when you are satisfied.

You can also go the other way around, decreasing the size of a larger camera view by dragging the mouse from the edge of the camera view towards another end of the camera view.

There are some limitations. You can only combine camera views in rectangular shapes. You cannot create, e.g., an L shape.

Neither you can overwrite a camera view containing content, i.e., only black camera views can be included in a larger camera view. It’s alright, though, to enlarge a camera view containing content.

Camera view popup menu

Right-clicking a camera view brings up a popup menu.

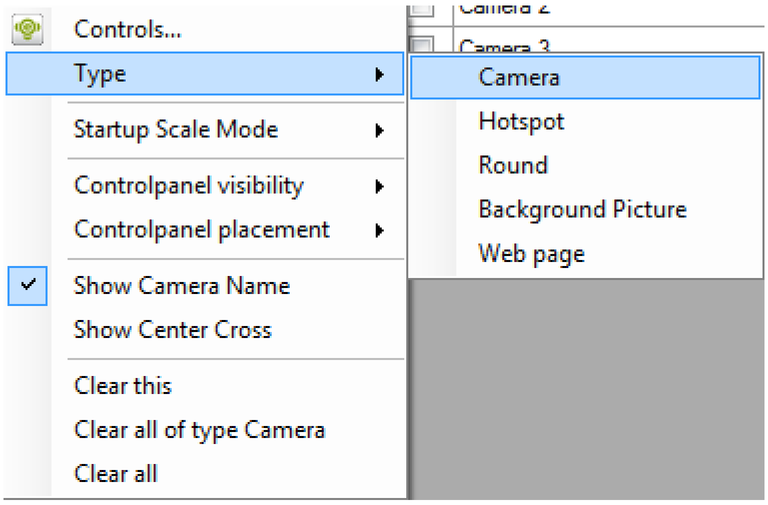


Figure 2.553 Camera view popup menu with submenu Type visible.

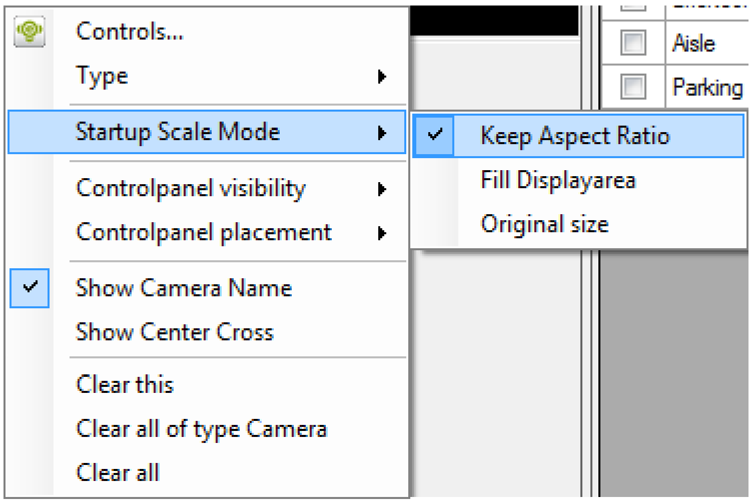


Figure 2.554 Camera view popup menu with submenu Startup Scale Mode visible.

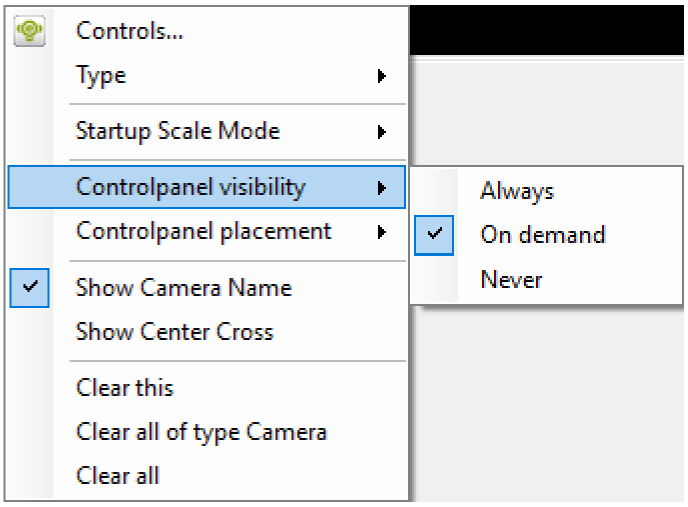


Figure 2.555 Camera view popup menu with submenu Controlpanel visibility visible.

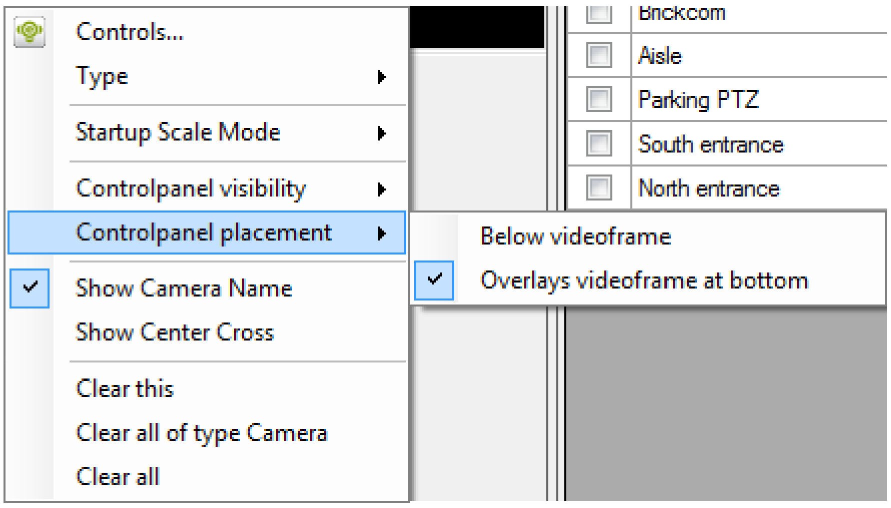


Figure 2.556 Camera view popup menu with submenu Controlpanel placement visible.

Controls... brings up a *control editor* where you can put control objects on top of the content in a camera view. Read more about this in section *Controls* on *page 2:369*.

Type selects the type of the camera view. The four different types *Camera*, *HotSpot*, *Round, Background picture,* and *Web page* will be explained in the following sections.

Startup Scale Mode determines how the content will be displayed at startup in Ethiris Client. This can be changed at runtime in Ethiris Client if you want. The options are:

Keep Aspect Ratio means that the frame fills out as much as possible of the camera view without distorting the proportions of the frame. This is the default mode of every camera view.

Fill Displayarea means that the frame is ”dragged” out so that it covers the entire area of the camera view. The aspect ratio is not preserved.

Original size means that the frame is displayed in the size it comes from the camera. This means that parts of the frame are cut if there is not enough room in the camera view.

Controlpanel visibility specifies when the control panel will be displayed when Ethiris Client is started. The operator can change this by right-clicking the camera view in live and select the desired mode.

Always means that the control panel will always be visible in the camera view.

On demand means that the control panel will normally be hidden, but when the user moves the mouse to the bottom of the camera view, the control panel will slide up and be visible. When the user moves the mouse out of the control panel, the panel will slide down again and hide.

Hidden means that the control panel will always be hidden.

Controlpanel placement specifies where the control panel will be located when it is displayed.

Below videoframe means that the image has slightly less space since the control panel is located at the bottom of the camera view. This area cannot be used for displaying video.

Overlays videoframe at bottom means that the whole area in the camera view can be used for displaying video. The control panel is on top of the video frame and ”covers” the lower part om the image. The control panel is semi-transparent.

Show Camera Name determines if the name of the camera will be displayed in live when the client is started. The operator can change this by right-clicking the camera view in live and select the desired mode.

Show Center Cross determines if a cross will be displayed in live to indicate the center of the video image. The operator can change this by right-clicking the camera view in live and select the desired mode.

Clear this deletes the content in the selected camera view.

Clear all of type Camera deletes the content of all camera views of type *Camera* in the current view.

Clear all deletes the content in all camera views in the current view.

Camera view type

There are four different types of camera views. The default type is *Camera*. When you right-click a camera view, a popup menu is displayed where you, among other things, can select *Type*.

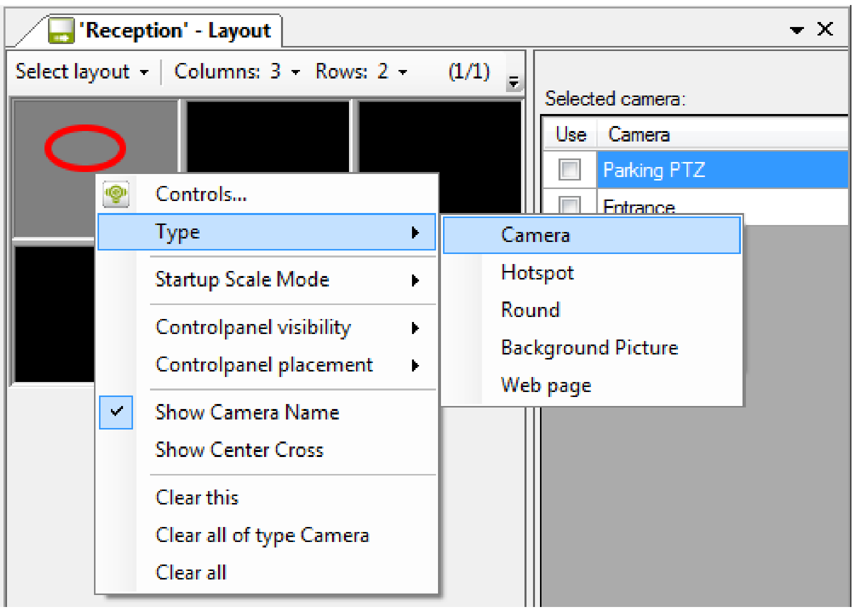


Figure 2.557 Right-click a camera view to display the popup menu.

Depending on which type the selected camera view has, different properties appear at the right side of the layout panel.

You select a camera view by clicking it. The selected camera view is highlighted. In the example above, the first (upper left) camera view is selected, which is indicated by a grey color instead of black.

From the start in a new view, no camera view has any content. To indicate this, they are black.

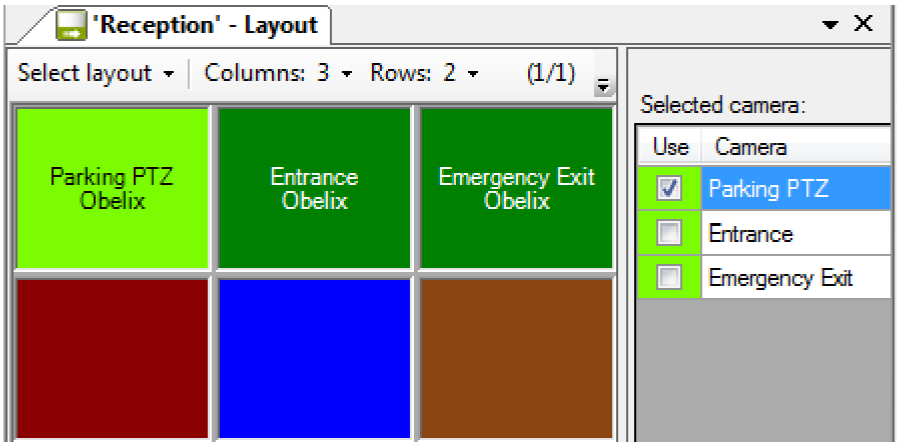


Figure 2.558 A view with camera views of different types.

Above in *Figure 2.558* is an example of a view with camera views of all four types. In the top row are all three camera views of type *Camera*. All three are green, including the name of the currently selected camera, the first one is selected, hence the light green color.

In the second row, the first camera view is of type *HotSpot*. This camera view is red.

The second camera view in the second row is of type *Round* and is blue.

The third camera view in the second row is of type *Background picture* and is brown.

You can change the type of any camera view by right-clicking it and select the desired type in the popup menu.

Let's have a look at the various types and their properties.

View type Camera

Camera is the most common view type. This is the default view type that every camera view has from the beginning. It is used for displaying live video from a specific and pre-defined camera.

When no camera is selected for such a camera view, the color of the camera view is black. In the example below, only the first camera view has a defined camera. The other 5 camera views have no cameras defined and are thus black.

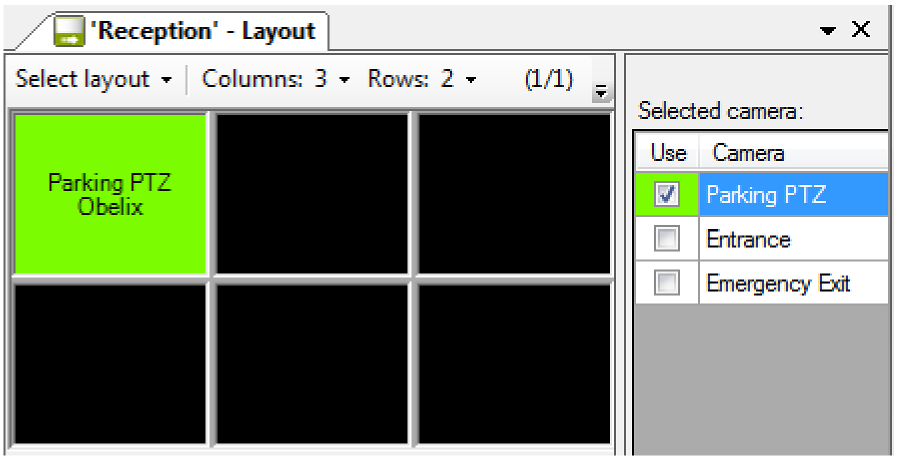
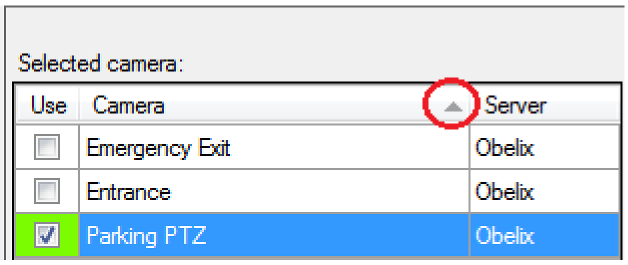


Figure 2.559 View type of camera.

To the right of the camera views, is a list of available cameras. This list consists of the cameras you have selected for the *Used Servers*. See section *Used Server panel* on *page 2:329* for more information about that.

Simply check the camera in the list which you want to display live video from in the selected camera view.

If you have many cameras, it can be convenient to sort them in alphabetical order. Click the *Camera* column header to do this. Click again to reverse the sort order. An arrow indicates the current sort order.



Selected camera:

Use	Camera	Server
☐	Emergency Exit	Obelix
☐	Entrance	Obelix
☑	Parking PTZ	Obelix

Figure 2.560 Sort the cameras by clicking the column header.

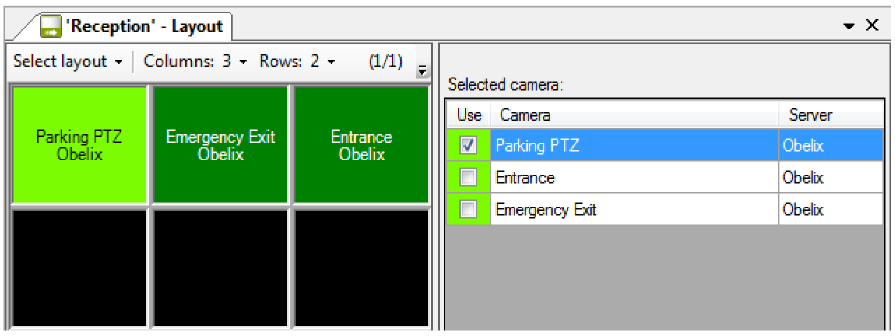
Add several cameras at once.

Another useful hint for adding many cameras to your layout is to select multiple cameras in the list and then drag them with the right mouse button. Release the mouse button on the first camera view you want to fill with cameras.

You can select cameras in several different ways. If you don't care about the order, you can simply click a row in the list with the left mouse button, hold the button down while moving the mouse over more rows, and finally release the mouse button when you have selected the desired cameras. In this case, the camera you clicked first will end up in the first camera view, and the following cameras will be placed in the order you selected the cameras.

You can also select cameras one by one by pressing the *Ctrl* key when selecting camera no 2 and forward. Then the cameras will be added to the camera views in the order you selected them.

Finally, when dragging the cameras into the camera views, you have to use the right mouse button. Release the mouse button on the first camera view you want to fill. The cameras will be added from left to right and from top to bottom.



Selected camera:

Use	Camera	Server
☑	Parking PTZ	Obelix
☐	Entrance	Obelix
☐	Emergency Exit	Obelix

Figure 2.561 Several cameras have been added at once.

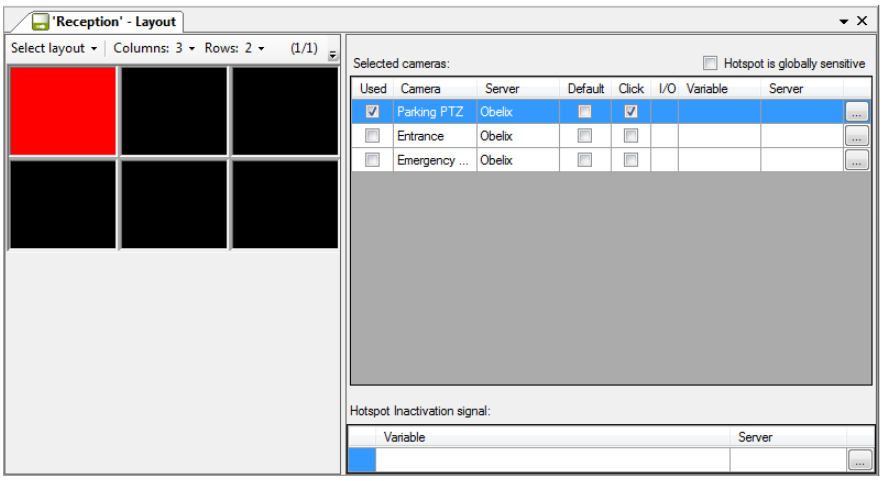
In the example above, the cameras have been selected in another order than they appear in the list.

You can change the order of the cameras.

If you want, you can change the order of the cameras in the camera views. Press the *Shift* key at the same time you drag a camera view with the left mouse button and release it on another camera view. The two camera views will then swap content.

View type Hotspot

The *Hotspot* type is used for displaying live video from a camera that is not pre-defined. The camera can change depending on the circumstances.



Selected cameras: ☐ Hotspot is globally sensitive

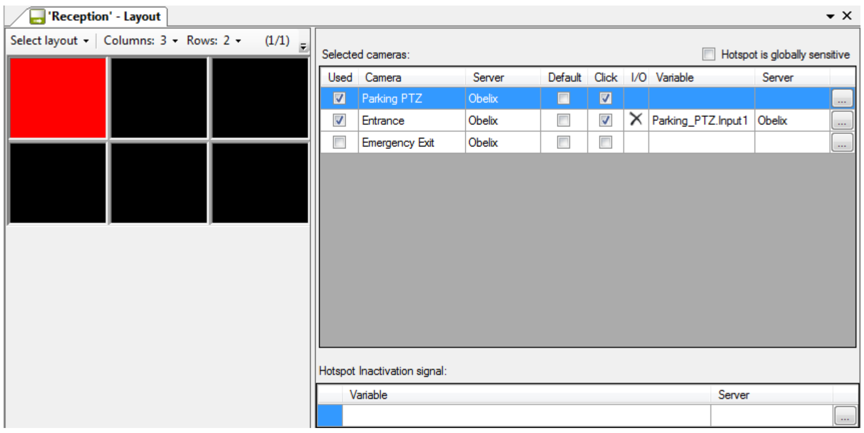
Used	Camera	Server	Default	Click	I/O	Variable	Server
☑	Parking PTZ	Obelix	☐	☑			
☐	Entrance	Obelix	☐	☐			
☐	Emergency ...	Obelix	☐	☐			

Figure 2.562 View type Hotspot.

To the right of the camera views is a list of available cameras, just as for the *Camera* view type. You decide which cameras can be displayed in the hotspot camera view.

There are two main functions. One is to display live cameras by manually clicking on a camera view or a camera symbol in Ethiris Client. The other is to display live cameras that are activated by a variable in Ethiris Server.

There are several columns in the camera list. Following is an explanation of each one of them.



Selected cameras: ☐ Hotspot is globally sensitive

Used	Camera	Server	Default	Click	I/O	Variable	Server
☑	Parking PTZ	Obelix	☐	☑			
☑	Entrance	Obelix	☐	☑	X	Parking_PTZ.Input1	Obelix
☐	Emergency Exit	Obelix	☐	☐			

Figure 2.563 Hotspot camera list.

Hotspot is globally sensitive. Check this if you want this hotspot view to react on clicks in camera views or camera symbols in another view.

Used is read-only and indicates that the camera is part of the hotspot. A camera is *used* if either the *Click* column is checked and/or the *Variable* column is specified. In the example above, 4 out of 8 cameras are used.

Camera indicates the name of the camera.

Server indicates the name of the Ethiris Server that the camera belongs to.

Default indicates the camera to display initially. You don't have to choose any default camera in which case the camera view will be black before any hotspot camera is activated. You can only choose any of the used cameras as default.

Click shall be checked for those cameras you want to be able to activate by clicking another camera view displaying the camera or clicking a camera symbol representing the camera. More about *camera symbols* later on.

I/O contains a delete button when an Ethiris Server variable is specified for the camera. Click on the delete button to delete the variable from the camera.

Variable contains the name of the selected variable for the activation of the camera. Click the *Browse* button to the far right for selecting the variable from a list. The hotspot camera is activated when the variable becomes true.

Server indicates the name of the Ethiris Server that the variable belongs to.

At the bottom of the panel, under the camera list, there is a field for selecting an inactivation signal. When the inactivation signal is activated, the camera view will go back to one out of two modes. If no *Default* camera is selected, no live image will be displayed, i.e., the view goes black. If a default camera is selected for the hotspot view, live images from this camera will be displayed when the inactivation signal becomes *true*.

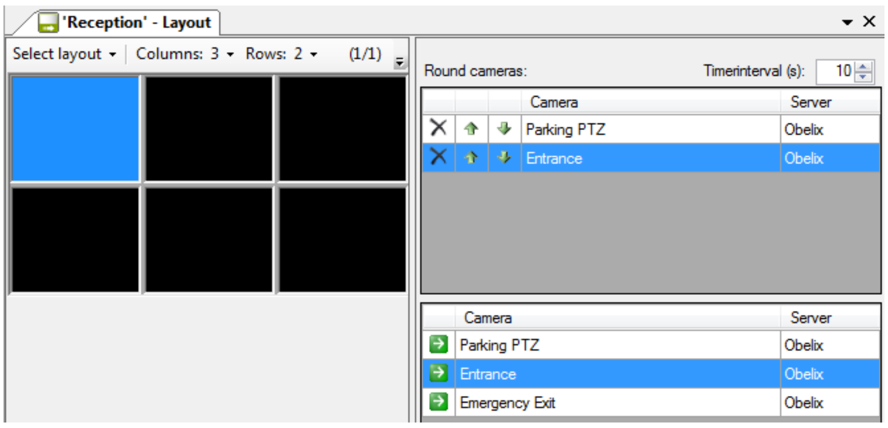
Hotspot Inactivation signal

Variable contains the name of the selected variable for the inactivation of the hotspot view. Click the *Browse* button to the far right for selecting the variable from a list. The camera view is inactivated when the variable becomes true.

Server indicates the name of the Ethiris Server that the variable belongs to.

View type Round

The *Round* type is used for displaying live video from several cameras, one at a time. You define a list of cameras from which live video automatically will change from camera to camera with a certain time interval.



			Camera	Server
✕	↑	↓	Parking PTZ	Obelix
✕	↑	↓	Entrance	Obelix

	Camera	Server
→	Parking PTZ	Obelix
→	Entrance	Obelix
→	Emergency Exit	Obelix

Figure 2.564 View type Round.

To the right of the camera views are two lists. The lower list is a list of available cameras, just as for the *Camera* view type. Click on a row in the lower list to add that camera to the end of the current list of Round cameras. The top list is a list of cameras included in the *Round* list. From the beginning, this list is empty.

There are several columns in the camera Round list. Following is an explanation of each one of them.

Timerinterval (s) is the number of seconds live video for each camera will be displayed in the camera view. The default value is *10* seconds.

✕ **Delete** is a button for deleting a camera from the list.

↑ **Up** is a button for moving the camera one step up on the list.

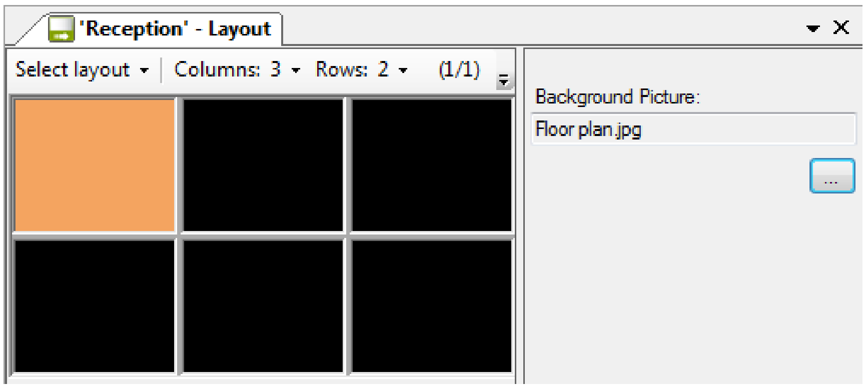
↓ **Down** is a button for moving the camera one step down in the list.

Camera indicates the name of the camera.

Server indicates the name of the Ethiris Server that the camera belongs to.

View type Background picture

The *Background picture* type is used for displaying a static image. Usually, you put some controls on the image like buttons or camera symbols that the operator can use for interacting with the system. But, it can also be a company logo or some other static information.



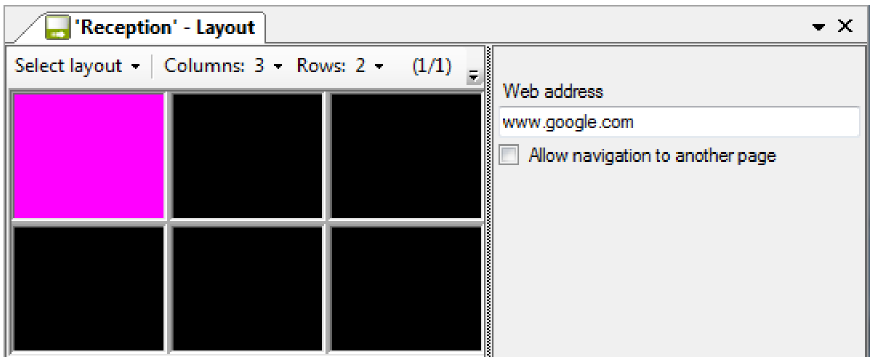
Background Picture:
Floor plan.jpg

Figure 2.565 View type Round.

Background picture is the name of an image file. It can be of type *bmp*, *jpg*, *gif*, or *png*. The image file is automatically imported to the client's configurations resource.

View type Web page

The *Web page* type is used for displaying a web page in the camera view. You enter the address of the start page to display when the camera view is first showed. If you check *Allow navigation to another page,* the user can navigate to another page by clicking links on the web page. It is not possible to put controls on a web page.



Web address
www.google.com
Allow navigation to another page

Figure 2.566 View type Web page.

Web page is the address of the page that will be loaded when the view is showed in the client.

Allow navigation to another page means that the user can navigate by clicking links on the web page.

Controls

You can put various types of *controls* on top of the content in a camera view. This is usually done in *Background picture* camera views, but may also be done in the other types of camera views. In the examples below, we will use a background picture.

Right-click the desired camera view and select *Controls…* in the popup menu. A still frame from the camera selected or the background image selected for the current camera view is displayed here. You can insert your control objects in this frame.

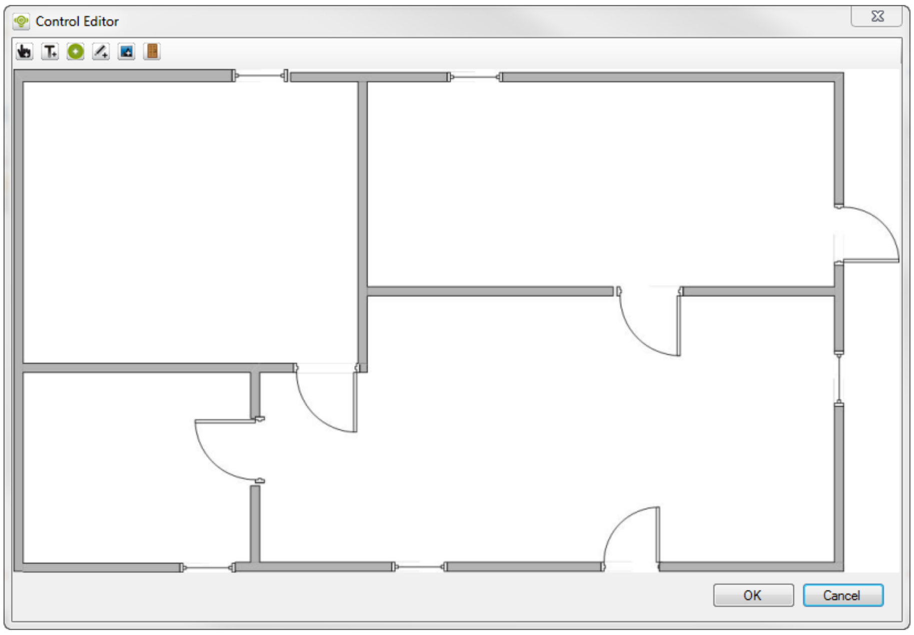


Figure 2.567 The control editor with a situation plan image loaded.

At the top of the editor, there are six tool buttons.

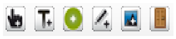


Figure 2.568 The toolbar in the Control Editor.

Add a new button	Use this button to add a *button* control. These are used to activate output signals or internal variables in the Ethiris server.
Add a new label	Use this button to add a *Label* control. These are used to present static texts or the current value from variables in the Ethiris server.
Add a new LED	Use this button to add a *LED* control. These are used to indicate the status of a Boolean variable in the Ethiris server.
Add a new camera	Use this button to add a *Camera* control. These are used to symbolize a camera. Clicking the camera symbol produces a live video from the current camera in the desired camera view and/or in a Hotspot view.
Add a new image	Use this button to add an *Image* control. These are used to display a static image, usually from a file.
Add a new door	Use this button to add a *Door* control. These are used to indicate the status of a door connected via an Access Controller in Ethiris Server.

Button

When you have added a button, you can move it around and resize it by dragging one of the 8 size handles.

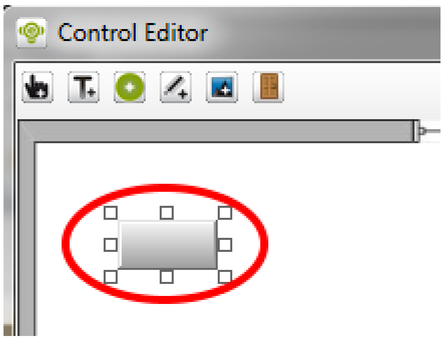


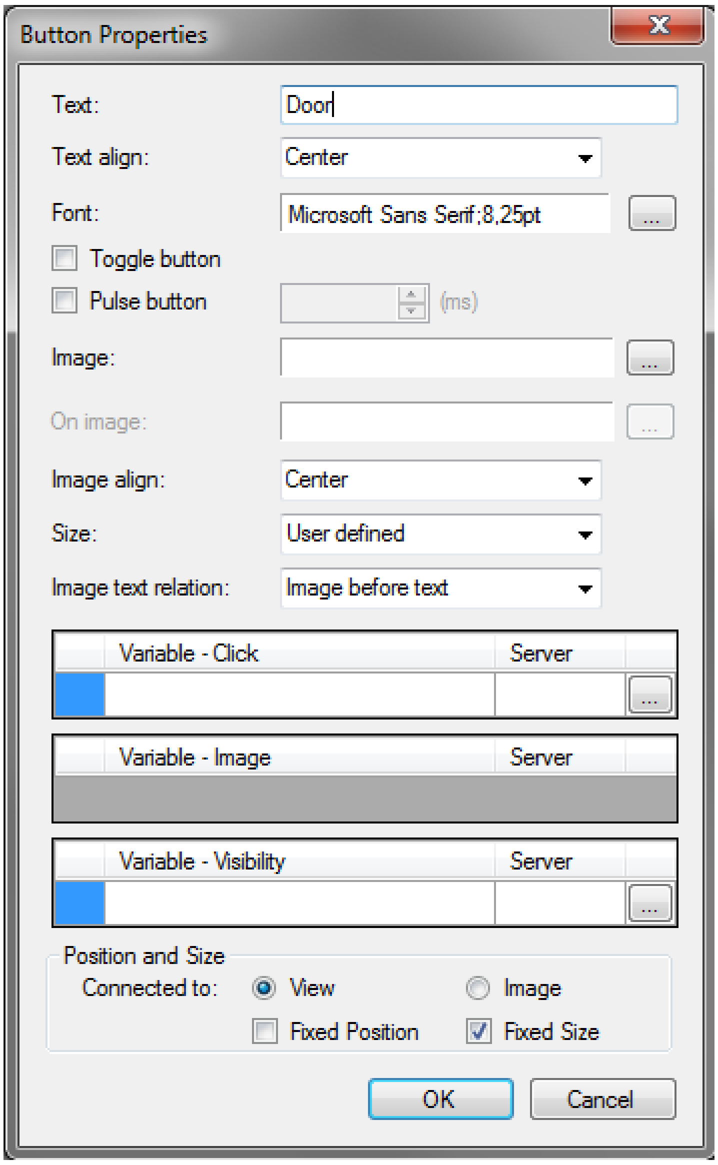
Figure 2.569 Button added to the control editor.

Right-click the button to popup a menu.

Figure 2.570 Popup menu for a control button.

Properties open a property dialog for the button. See the explanation below.

Delete removes the button from the camera view.



Text: Door
Text align: Center
Font: Microsoft Sans Serif;8,25pt
[ ] Toggle button
[ ] Pulse button (ms)
Image:
On image:
Image align: Center
Size: User defined
Image text relation: Image before text

Variable - Click	Server

Variable - Image	Server

Variable - Visibility	Server

Position and Size
Connected to: [✓ View] Image
[ ] Fixed Position [✓ Fixed Size]

Figure 2.571 Properties dialog for a control button.

Text is the caption of the button.

Text align determines the horizontal alignment of the text. You can choose between *Left*, *Center* & *Right*.

Font is the font of the text in the button. A standard font selection dialog is opened when you click the browse button to the right.

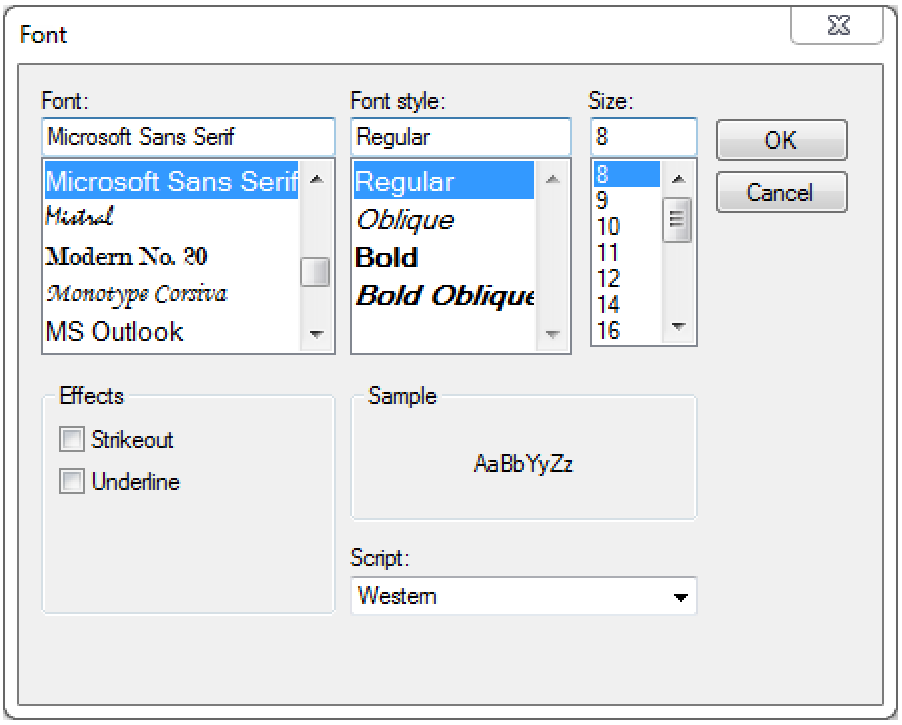


Figure 2.572 Font selection dialog.

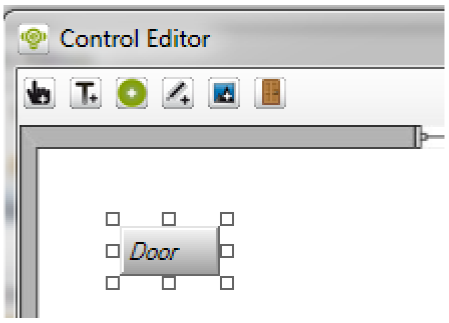


Figure 2.573 Button with text "Door" aligned to Left and in Italics.

Toggle button determines the behavior of the button. When checked, the button becomes a toggle button, i.e., it stays pressed when clicked, and when clicked again, it goes up. When not checked, the button goes up as soon as you release the mouse button when clicking the button.

Pulse button determines the behavior of the button. When checked, it is possible to define the pulse time in ms. This means that when you click the button in Ethiris Client, the signal will remain active the set time after you release the button. This setting can only be used if the button is not a *toggle button*.

Image lets you choose an image for the button. You can choose images of type *bmp*, *gif*, *jpg*, *png,* and *ico*.

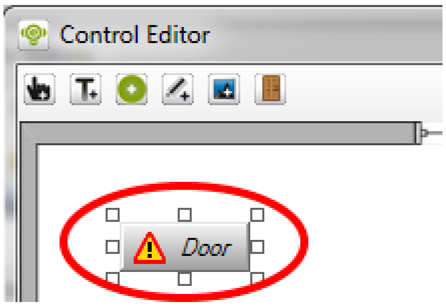


Figure 2.574 Button with both Text and Image.

The placement of the image depends on other settings for the control button such as *Image align* and *Image Text relation*.

On image lets, you choose a secondary image for the button. You can choose images of type *bmp*, *gif*, *jpg*, *png* & *ico*. On image can only be defined if Image is defined. This image can be displayed when the button is pressed or controlled via a separate variable, see *Variable – Image* below.

Image align determines the alignment of the image. If there is no text, the alignment is relative to the whole button. If there is text, space is divided between the image and the text, and then the image is aligned within the space reserved for the image. The alternatives are *Left*, *Center,* and *Right*.

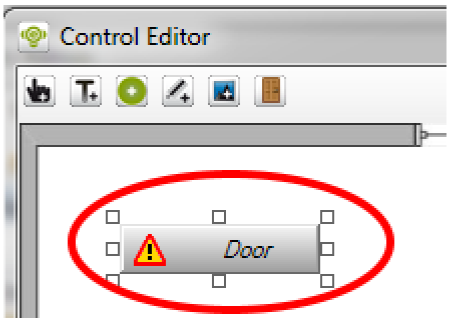


Figure 2.575 Button with Text aligned to Left and Image aligned to the Center.

In the example above, the image is centered on the left half of the button.

Size determines the size of the button. There are four alternatives with predefined sizes and one alternative for user-defined size. The alternatives are *Mini*, *Small*, *Medium*, *Large,* and *User defined*. User defined is default and means that you can size the button by dragging the size handles to the desired size.

Image text relation determines the relation between text and image. The alternatives are *Image before text*, *Text before image, Image above text,* and *Only image*. The default is *Image before text*.

Variable - Click is the variable the button is connected to. There is not much use for a button without a corresponding variable. The variable is set to *true* when the button is pressed, and the variable is set to *false* when the button is released. When you browse for a variable, only *Boolean* variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a writeable variable. Click the *Delete* button to remove the variable.

Variable - Image is the variable that controls which image (*Image* or *On image)* should be displayed in the button. This variable can only be defined if both *Image* and *On image* are defined. If this variable is not defined, the displayed image is controlled by the variable *Variable – Click.* When you browse for a variable, only *Boolean* variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a readable variable. Click the *Delete* button to remove the variable.

Variable - Visibility is the variable that controls whether the button is visible in Ethiris Client. If the variable isn't defined, the button is always visible. When you browse for a variable, only Boolean variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a readable variable. Click the *Delete* button to remove the variable.

Position & Size is where you determine how the button will be positioned and sized depending on the size of the camera view/underlying image where the button will be displayed.

Connected to View means that the button's position and size are relative to the camera view rather than the image that is displayed in the camera view. When the size of the camera view changes, the position and size of the button is adjusted accordingly unless *Fixed position* and/or *Fixed size* are checked.

Connected to Image means that the button's position and size are relative to the underlying image that is displayed in the camera view. When the size of the image changes, due to digital zooming, the position and size of the button is adjusted unless *Fixed position* and/or *Fixed size* are checked.

Fixed position means that the button's position is not affected by the current size of the camera view/underlying image. The position always remains the same in absolute pixels.

Fixed size means that the button's size will not be affected by the current size of the camera view/underlying image. The size will remain the same in absolute pixels.

Label

When you have added a label control, you can move it around and resize it by dragging one of the 2 size handles.

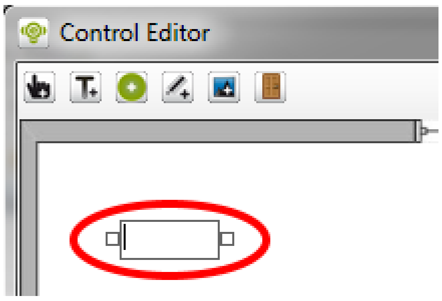


Figure 2.576 Label added to the control editor.

Right-click the label object to popup a menu.

Figure 2.577 Popup menu for a control label.

Properties open a property dialog for the label. See the explanation below.

Delete removes the label object from the camera view.

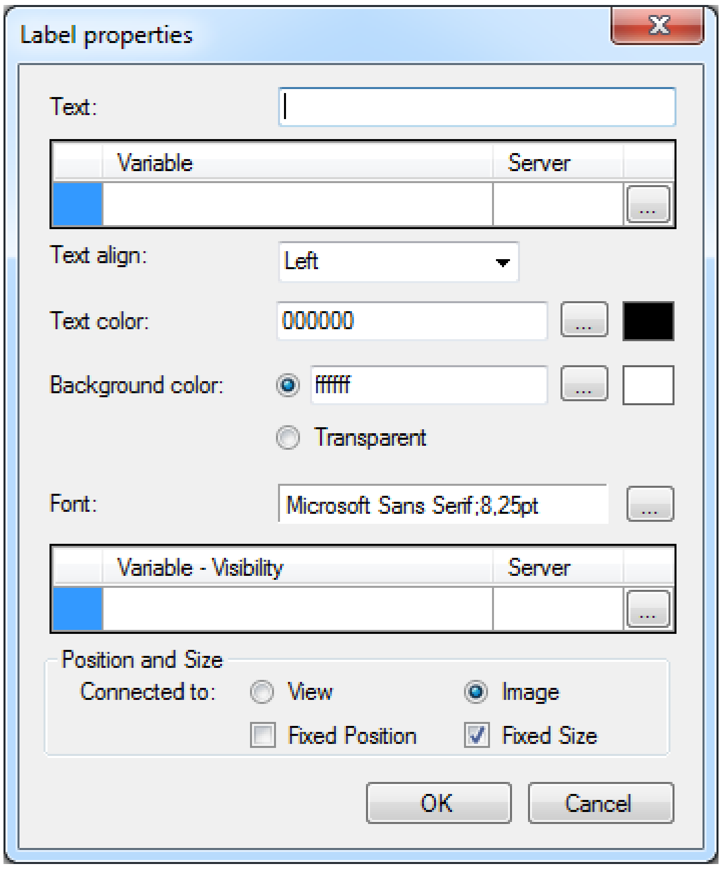


Figure 2.578 Properties dialog for a label.

Text can be used for entering a static caption that will be displayed in the label object.

Variable is the variable the label is connected to. Any variable can be connected to a label. The value of the variable is converted to text and is displayed as a caption in the label. If a variable is selected, the value of the variable is displayed instead of any text in the *Text* field. Click the *Delete* button to remove the variable.

Text align determines the horizontal alignment of the text. You can choose between *Left*, *Center* & *Right*.

Text color determines the color of the text.

Background color determines the background color in the label object. If you check *Transparent,* the label will be drawn without a background.

Font is the font of the text in the label. A standard font selection dialog is opened when you click the browse button to the right. This is the same dialog as for a *Button*.

Variable - Visibility is the variable that controls whether the label is visible in Ethiris Client. If the variable isn't defined, the label is always visible. When you browse for a variable, only *Boolean* variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a readable variable. Click the *Delete* button to remove the variable.

Position & Size is where you determine how the label will be positioned and sized depending on the size of the camera view/underlying image where the label will be displayed.

Connected to View means that the label's position and size are relative to the camera view rather than the image that is displayed in the camera view. When the size of the camera view changes, the position and size of the label is adjusted accordingly unless *Fixed position* and/or *Fixed size* are checked.

Connected to Image means that the label's position and size are relative to the underlying image that is displayed in the camera view. When the size of the image changes, due to digital zooming, the position and size of the label is adjusted unless *Fixed position* and/or *Fixed size* are checked.

Fixed position means that the label's position is not affected by the current size of the camera view/underlying image. The position always remains the same in absolute pixels.

Fixed size means that the label's size will not be affected by the current size of the camera view/underlying image. The size will remain the same in absolute pixels.

LED

When you have added an LED control, you can move it around and resize it by dragging one of the 2 size handles.

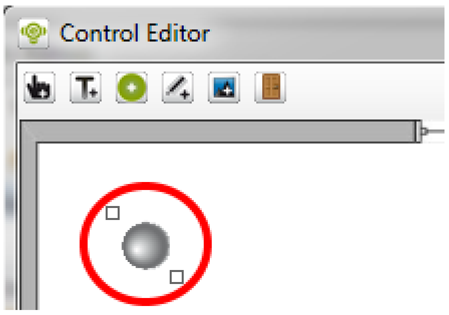


Figure 2.579 LED added to the control editor.

Right-click the LED object to popup a menu.

Figure 2.580 Popup menu for an LED.

Properties open a property dialog for the LED. See the explanation below.

Delete removes the LED object from the camera view.

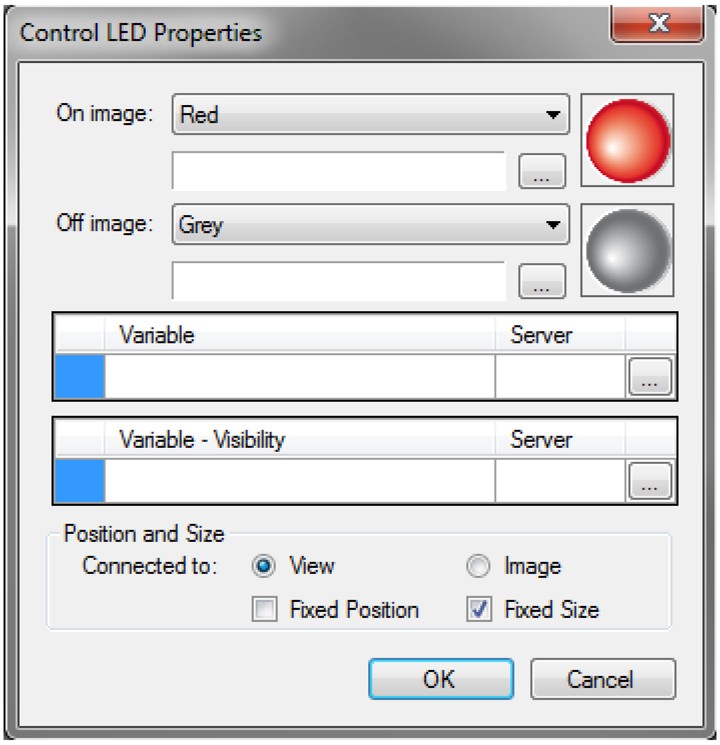


Figure 2.581 Properties dialog for a LED.

On image is the image that is displayed when the value of the connected variable is *true*. You can choose from 5 standard images in the colors *Red*, *Green*, *Yellow*, *Blue,* and *Grey*. You can also choose an image file in one of the formats *bmp*, *gif*, *jpg*, *png* & *ico*.

Off image is the image that is displayed when the value of the connected variable is *false*. You can choose from 5 standard images in the colors *Red*, *Green*, *Yellow*, *Blue,* and *Grey*. You can also choose an image file in one of the formats *bmp*, *gif*, *jpg*, *png* & *ico*.

Variable is the variable the LED is connected to. Only Boolean variables can be connected to an LED. When the value of the variable is *true*, the *On image* is displayed, and when the value of the variable is *false*, the *Off image* is displayed. Click the *Delete* button to remove the variable.

Variable - Visibility is the variable that controls whether the LED is visible in Ethiris Client. If the variable isn't defined, the LED is always visible. When you browse for a variable, only Boolean variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a readable variable. Click the *Delete* button to remove the variable.

Position & Size is where you determine how the LED will be positioned and sized depending on the size of the camera view/underlying image where the LED will be displayed.

Connected to View means that the LED's position and size are relative to the camera view rather than the image that is displayed in the camera view. When the size of the camera view changes, the position and size of the LED is adjusted accordingly unless *Fixed position* and/or *Fixed size* are checked.

Connected to Image means that the LED's position and size are relative to the underlying image that is displayed in the camera view. When the size of the image changes, due to digital zooming, the position and size of the LED is adjusted unless *Fixed position* and/or *Fixed size* are checked.

Fixed position means that the LED's position is not affected by the current size of the camera view/underlying image. The position always remains the same in absolute pixels.

Fixed size means that the LED's size will not be affected by the current size of the camera view/underlying image. The size will remain the same in absolute pixels.

Camera

When you have added a Camera control, you can move it around and resize it by dragging one of the 2 size handles.

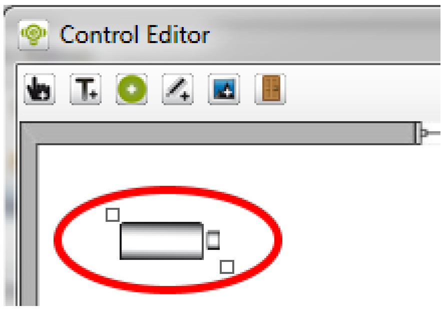


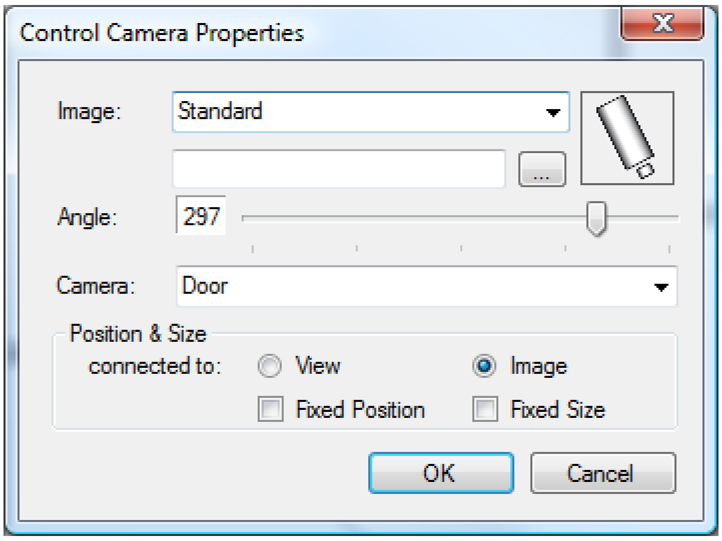
Figure 2.582 Camera added to the control editor.

Right-click the Camera object to popup a menu.

Figure 2.583 Popup menu for a Camera.

Properties open a property dialog for the Camera. See the explanation below.

Delete removes the Camera object from the camera view.



Control Camera Properties

Image: Standard

Angle: 297

Camera: Door

Position & Size connected to: View [✓ Image]

Fixed Position Fixed Size

Figure 2.584 Properties dialog for a Camera.

Image is the image that is displayed representing the camera. You can choose *Standard,* or you can choose an image file in one of the formats *bmp*, *gif*, *jpg*, *png* & *ico*.

Angle is the angle the camera will be rotated.

Camera is the actual camera that is represented. In Ethiris Client, when the camera symbol is clicked, live images from the represented camera will be displayed in any *Hotspot* view where the camera is included. The effect is the same as when clicking a camera view with live video from a certain camera.

Position & Size is where you determine how the camera will be positioned and sized depending on the size of the camera view/underlying image where the camera will be displayed.

Connected to View means that the camera's position and size are relative to the camera view rather than the image that is displayed in the camera view. When the size of the camera view changes, the position and size of the camera is adjusted accordingly unless *Fixed position* and/or *Fixed size* are checked.

Connected to Image means that the camera's position and size are relative to the underlying image that is displayed in the camera view. When the size of the image changes, due to digital zooming, the position and size of the camera is adjusted unless *Fixed position* and/or *Fixed size* are checked.

Fixed position means that the camera's position is not affected by the current size of the camera view/underlying image. The position always remains the same in absolute pixels.

Fixed size means that the camera's size will not be affected by the current size of the camera view/underlying image. The size will remain the same in absolute pixels.

Image

When you have added a label control, you can move it around and resize it by dragging one of the 8 size handles.

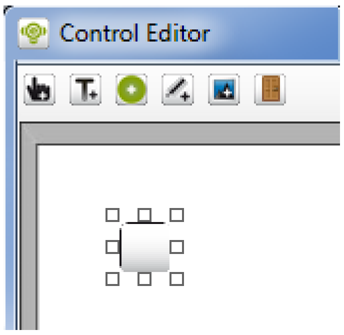


Figure 2.585 Image added to the control editor.

Right-click the image object to popup a menu.

Figure 2.586 Popup menu for an image.

Properties open a property dialog for the image. See the explanation below.

Delete removes the image object from the camera view.

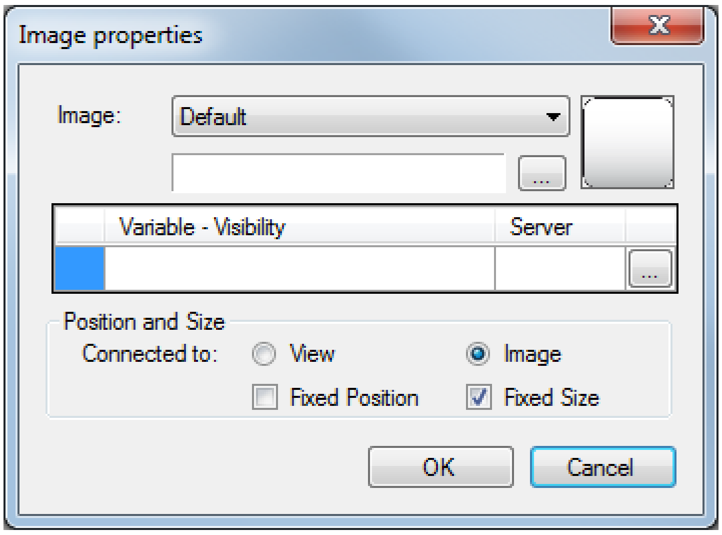


Figure 2.587 Property dialog for an image.

Image is the image to show. You can select *Standard* or an image file in one of the formats *bmp*, *gif*, *jpg*, *png,* or *ico*.

Variable - Visibility is the variable that controls whether the image is visible in Ethiris Client. If the variable isn't defined, the label is always visible. When you browse for a variable, only *Boolean* variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a readable variable. Click the *Delete* button to remove the variable.

Position & Size is where you determine how the image will be positioned and sized depending on the size of the camera view/underlying image where the image will be displayed.

Connected to View means that the position and size of the image control are relative to the camera view rather than the image that is displayed in the camera view. When the size of the camera view changes, the position and size of the image control is adjusted accordingly unless *Fixed position* and/or *Fixed size* are checked.

Connected to Image means that the position and size of the image control are relative to the underlying image that is displayed in the camera view. When the size of the image changes, due to digital zooming, the position and size of the image control is adjusted unless *Fixed position* and/or *Fixed size* are checked.

Fixed position means that the position of the image control is not affected by the current size of the camera view/underlying image. The position always remains the same in absolute pixels.

Fixed size means that the size of the image control will not be affected by the current size of the camera view/underlying image. The size will remain the same in absolute pixels.

Door

When you have added a Door control, you can move it around and resize it by dragging one of the 2 size handles.

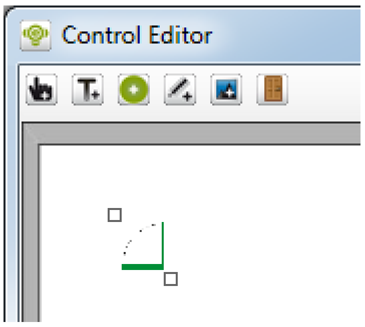


Figure 2.588 Door added to the control editor.

Right-click the Door control to popup a menu.

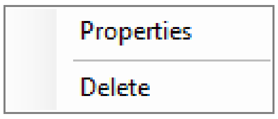


Figure 2.589 Popup menu for a Door.

Properties open a property dialog for the LED. See the explanation below.

Delete removes the LED object from the camera view.

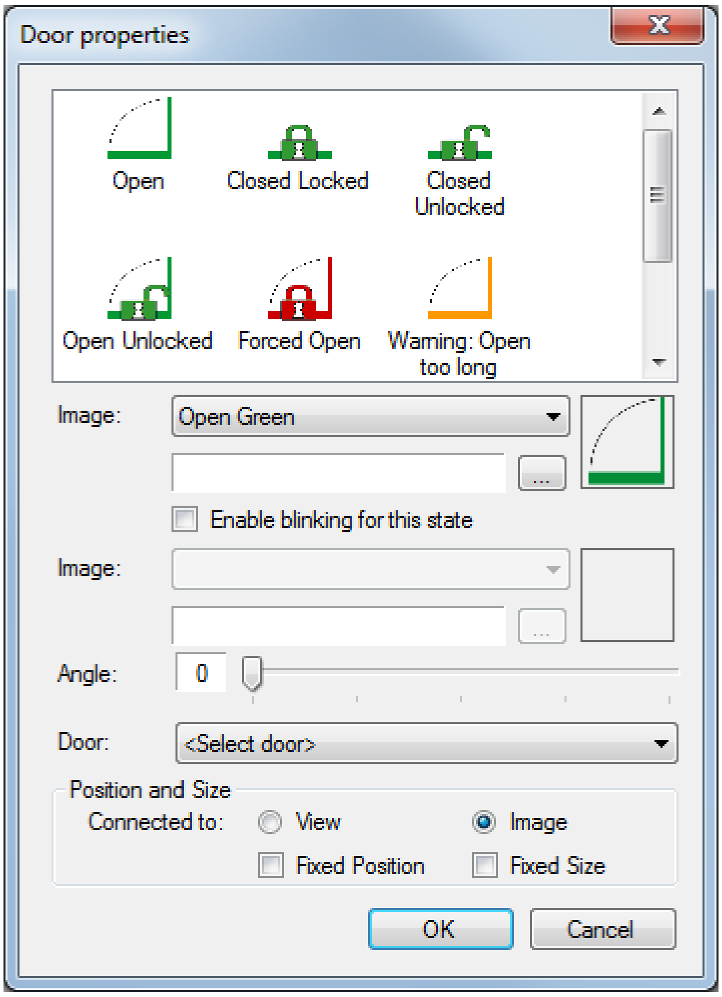


Image: Open Green

[ ] Enable blinking for this state

Image:

Angle: 0

Door: <Select door>

Position and Size

Connected to: ( ) View [✓ Image]

[ ] Fixed Position [ ] Fixed Size

Figure 2.590 Properties dialog for a door.

Image is the image to show. You can select *Standard* or an image file in one of the formats *bmp*, *gif*, *jpg*, *png,* or *ico*.

Click on one of the states that the door can have in the list at the top of the dialog. Then select an image from the list of standard images or select any image file (*bmp*, *gif*, *jpg*, *png,* or *ico*) to show for that door state. Note that, for the images to display properly in Ethiris Client, images for all door states of a door must have the same relation between width and height. The standard images are 223x262 pixels.

Enable blinking for this state check this if the door control should blink when shown in this state in Ethiris Client. If checked, select an image that the blink shall alternate with in the field **Image** below.

Angle is the angle the door will be rotated.

Door is the door that should be represented. Select a door from the list.

Position & Size is where you determine how the door will be positioned and sized depending on the size of the camera view/underlying image where the door will be displayed.

Connected to View means that the position and size of the door are relative to the camera view rather than the image that is displayed in the camera view. When the size of the camera view changes, the position and size of the door is adjusted accordingly unless *Fixed position* and/or *Fixed size* are checked.

Connected to Image means that the position and size of the door are relative to the underlying image that is displayed in the camera view. When the size of the image changes, due to digital zooming, the position and size of the door is adjusted unless *Fixed position* and/or *Fixed size* are checked.

Fixed position means that the position of the door is not affected by the current size of the camera view/underlying image. The position always remains the same in absolute pixels.

Fixed size means that the size of the door will not be affected by the current size of the camera view/underlying image. The size will remain the same in absolute pixels.

2.4.95 Button node

Under the *Views* node and possibly under any *Section* node, there may be one or several *Button* nodes. The buttons can be an *Audio button* or an *I/O button*.

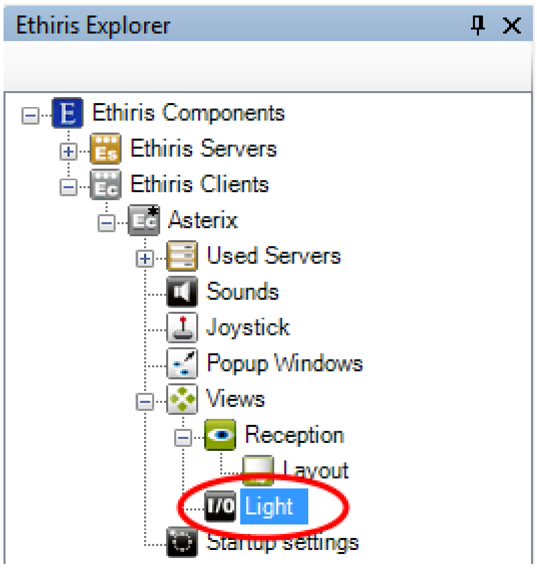


Figure 2.591 A View node in Ethiris Explorer treeview.

Button popup menu

Right-clicking this node brings up a context menu.

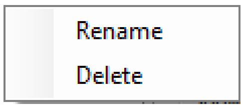


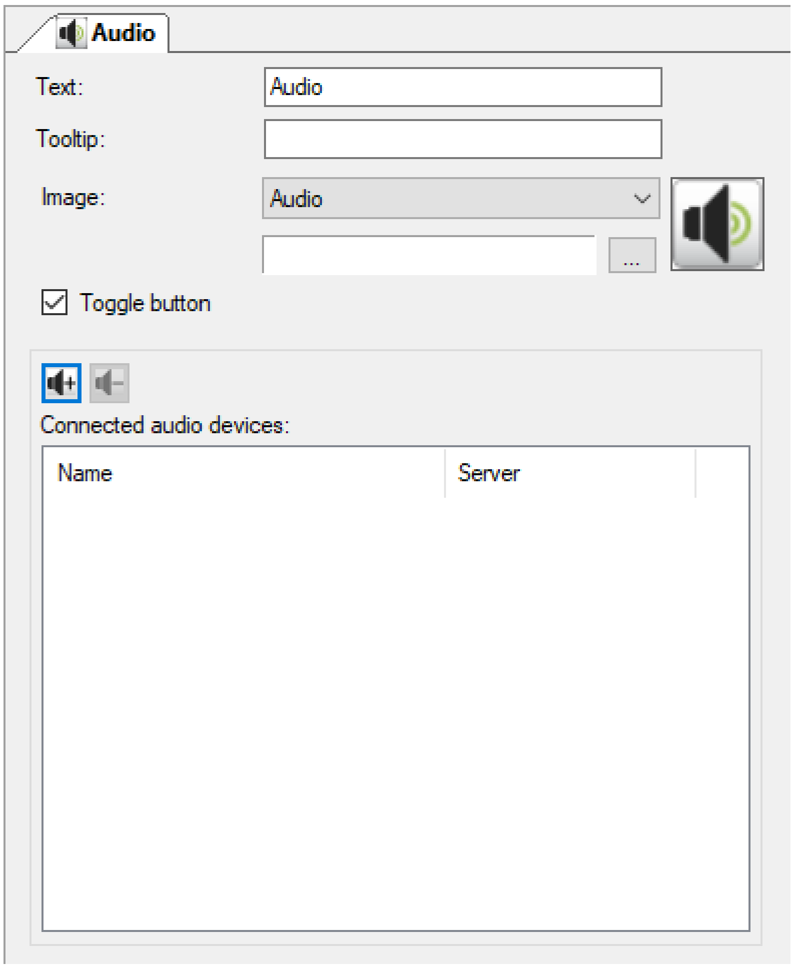
Figure 2.592 The popup menu for a Button node.

Rename sets the node in *rename* mode. You can enter the new name for the button directly in the treeview.

Delete removes the button from the client configuration. A confirmation dialog will be displayed before deletion.

Audio button panel

Double-clicking an *Audio button* node in the treeview opens the corresponding panel.



Audio

Text: Audio

Tooltip:

Image: Audio

[✓] Toggle button

Connected audio devices:

Name	Server

Figure 2.593 The Audio button panel.

This panel contains properties for the button such as text and image for the corresponding button that will be displayed in Ethiris Client in the *Views Explorer* and the *Section Explorer* tool windows.

Text determines which text will be displayed in the corresponding button in Ethiris Client.

Tooltip is a text that will be displayed in a tooltip when holding the mouse pointer above the button in the *Views Explorer* tool window in Ethiris Client. This can be used to explain the purpose of a button.

Image determines which image will be displayed on the button in Ethiris Client. There are two main options; either you use one of the standard images that are provided with Ethiris, or you use your own image.

The following standard images are available:

Audio

One

Two

Three

2x2 *Four*

3x2 *Six*

4x2 *Eight*

3x3 *Nine*

4x3 *Twelve*

5x3 *Fifteen*

4x4 *Sixteen*

5x4 *Twenty*

5x5 *Twentyfive*

Free

I/O *IO*

When you use your own image, the following formats are supported: *bmp*, *gif*, *jpg*, *png,* and *ico*. For the best result, use an image with the resolution *32 x 32 pixels*.

Toggle button determines the behavior of the button. When checked, the button becomes a toggle button, i.e., it stays pressed when clicked, and when clicked again, it goes up. When not checked, the button goes up as soon as you release the mouse button when clicking the button.

Connected audio devices is a list of connected audio devices. Click the button to begin adding audio devices to the list. A panel, *Connect audio device to audio button* will open.

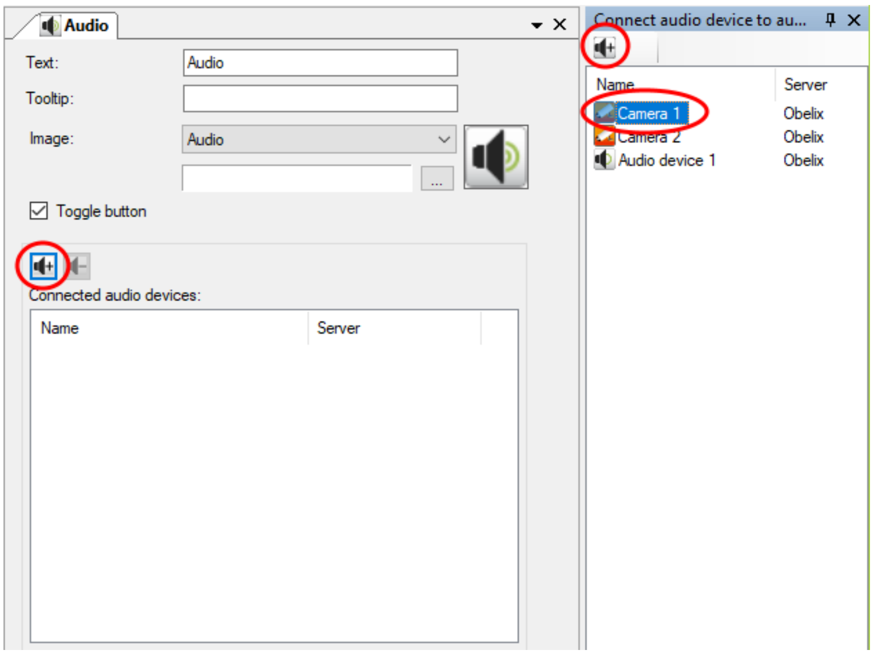


Figure 2.594 Adding audio devices to Audio button.

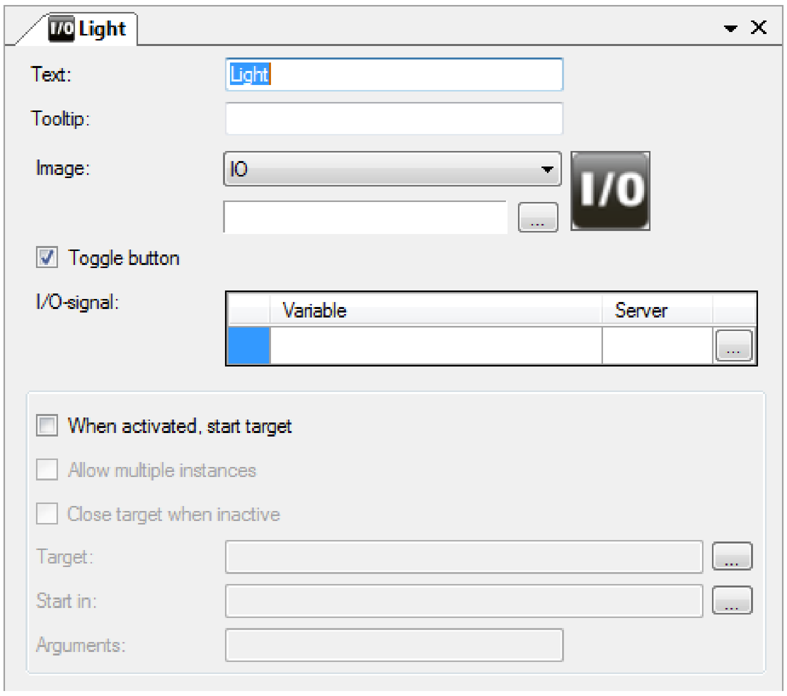
Select the audio device you want to add to the button in the list to the right and click the button in the *Connect audio device to audio button* panel.

You can add several audio devices to the audio button if you want to cover a large area when you talk. The talk will be sent in parallel to all audio devices in the list.

If you want to remove a connected audio device from the audio button, select the audio device in the list *Connected audio devices* and click the button.

I/O Button panel

Double-clicking an I/O*Button* node in the treeview opens the corresponding panel.



Light

Text:	Light
Tooltip:	
Image:	IO
☑ Toggle button	
I/O-signal:	Variable / Server
☐ When activated, start target	
☐ Allow multiple instances	
☐ Close target when inactive	
Target:	
Start in:	
Arguments:	

Figure 2.595 The Button panel.

This panel contains properties for the button such as text and image for the corresponding button that will be displayed in Ethiris Client in the *Views Explorer* and the *Section Explorer* tool windows.

Text determines which text will be displayed in the corresponding button in Ethiris Client.

Tooltip is a text that will be displayed in a tooltip when holding the mouse pointer above the button in the *Views Explorer* tool window in Ethiris Client. This can be used to explain the purpose of a button.

Image determines which image will be displayed on the button in Ethiris Client. There are two main options; either you use one of the standard images that are provided with Ethiris, or you use your own image.

The following standard images are available:

IO

One

Two

Three

Four

3x2 *Six*

4x2 *Eight*

3x3 *Nine*

4x3 *Twelve*

5x3 *Fifteen*

4x4 *Sixteen*

5x4 *Twenty*

5x5 *Twentyfive*

Free

Audio

When you use your own image, the following formats are supported: *bmp*, *gif*, *jpg*, *png,* and *ico*. For the best result, use an image with the resolution *32 x 32 pixels*.

Toggle button determines the behavior of the button. When checked, the button becomes a toggle button, i.e., it stays pressed when clicked, and when clicked again, it goes up. When not checked, the button goes up as soon as you release the mouse button when clicking the button.

I/O-signal is the variable the button is connected to. The variable is set to *true* when the button is pressed, and the variable is set to *false* when the button is released. When you browse for a variable, only *Boolean* variables are presented in the browser dialog. Even though both writeable and readable variables are displayed, it only makes sense to use a writeable variable. Click the *Delete* button to remove the variable.

When activated, start target When checked, the defined application pointed out in the *Target* field is started both when the user clicks the button and when the button is pressed due to the I/O-signal connected to the button is activated. This means that the target application can be started locally on the computer where Ethiris Client runs from a script in the server if required.

Allow multiple instances When checked, a new instance of the *Target* application will be started each time the button is pressed. It is up to the operator to close the applications that are started.

Close target when inactive When checked, the *target* application will be closed when the button is released/deactivated. This option is only available if *Allow multiple instances* is not checked.

Target determines the application to start when the button I pressed. You can browse for the application by clicking the button with three dots or write the full path of the application in the text field. It is also possible to enter a web address (like www.kentima.com) or something else that is associated with a program in Windows. In that case, the associated program will start. It might not be possible to close the application from Ethiris Client when it is started like this, i.e., *Close target when inactive* might not work.

Start in determines the startup folder for the target application. Some applications might need a specific folder to start in, to work correctly. You can browse for the folder by clicking the button with three dots or write the path of the folder.

Arguments Enter start arguments of the target application in a format supported by the target application.

2.4.96 Startup settings node

Under the node Ethiris Client, there is a *Startup settings* node. This node is used for configuring how Ethiris Client will appear when it starts.

You can define size and position for the main window in Ethiris Client, which panels will be visible, if the menu will be visible, size of the tool buttons, and which client view will be displayed in the *Default Live View* and possible popup windows at start.

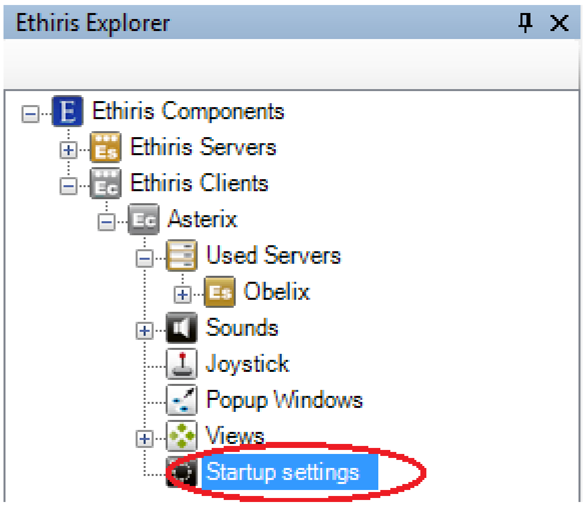


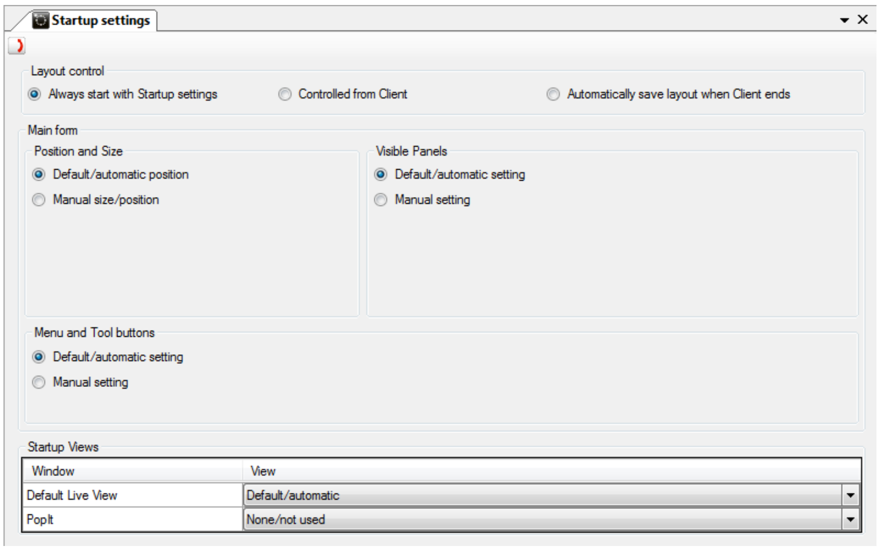
Figure 2.596 The node Startup settings in the Ethiris Explorer treeview.

Startup settings popup menu

There is no popup menu for this node.

Startup settings panel

Double-click on the node *Startup settings* in the treeview opens the corresponding panel.



Layout control
[✓ Always start with Startup settings] Controlled from Client Automatically save layout when Client ends

Main form

Position and Size
[✓ Default/automatic position]
Manual size/position

Visible Panels
[✓ Default/automatic setting]
Manual setting

Menu and Tool buttons
[✓ Default/automatic setting]
Manual setting

Startup Views

Window	View
Default Live View	Default/automatic
PopIt	None/not used

Figure 2.597 The panel Startup settings.

At the top of the panel, there is a toolbar.

Startup settings panel toolbar

Figure 2.598 The toolbar in the panel Startup settings.

Reset all settings to Default

Use this button to reset possible changes in the startup settings to their default values.

Startup settings panel

Layout control

You can choose how the client will behave at the start or at the reload of the configuration. There are two different states that the client can use when starting or reloading. The first is the startup settings that you set in this panel. The other one is a saved layout that contains information about the size, position, and content of all the windows that were open when the layout was saved. Which one of these two states that is used depends on which alternative you select for *Layout control*.

Always start with Startup settings means that each time the client is started or the configuration is reloaded, the settings that have been set further down in this panel are used. In this case, there is a menu in Ethiris Client under the *Tools* menu named *Restore Layout to Startup settings*. If you select this menu, the client will be restored to the state defined by the *Startup settings* in this panel.

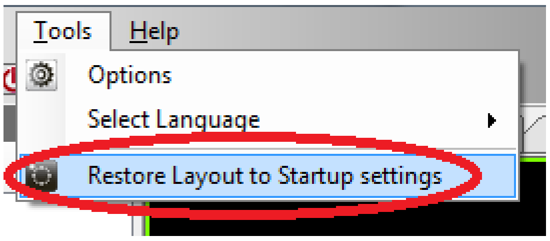


Figure 2.599 The Tools menu in Ethiris Client when Always start with Startup settings is selected.

Controlled from Client means that further menu options appear in the client *Tools* menu. Now you can save the current layout explicitly. You can also choose to automatically save the current layout every time the client is closed, or the configuration is reloaded.

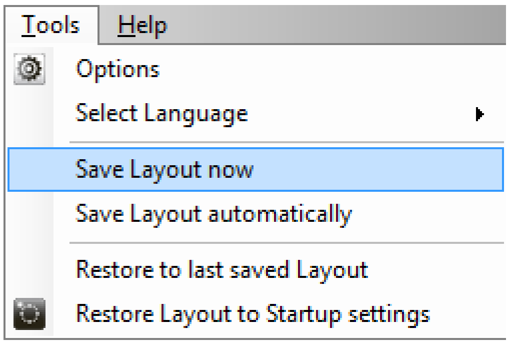

Tools
Help
Options
Select Language
Save Layout now
Save Layout automatically
Restore to last saved Layout
Restore Layout to Startup settings

Figure 2.600 The Tools menu in Ethiris Client when Controlled from Client is selected.

Automatically save layout when Client ends means once again that the client *Tools* menu changes.

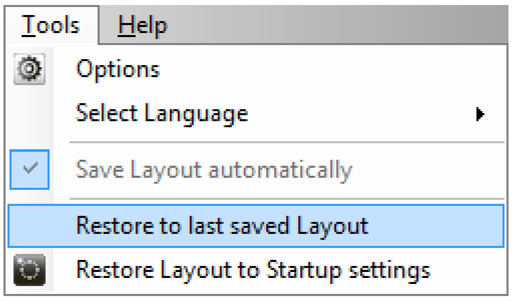

Tools
Help
Options
Select Language
Save Layout automatically
Restore to last saved Layout
Restore Layout to Startup settings

Figure 2.601 The Tools menu in Ethiris Client when Automatically save layout when Client ends is selected.

Now the menu **Save Layout automatically** in the client is automatically selected and cannot be changed. This means that the current layout in the client is automatically saved when the client is closed or when the configuration in the client is reloaded. This state is restored next time the client is started.

Main form

<u>Position and Size</u>

You can choose between *Default/automatic position* and *Manual size/position*.

Default/automatic position means that the client will be started with the size and position it had when it was closed. The client remembers its latest size and position for the main window. It doesn't remember size and position for possible popup windows.

Manual size/position means that you specify the size and position that will be applied when the client is started instead of the latest size and position of the client.

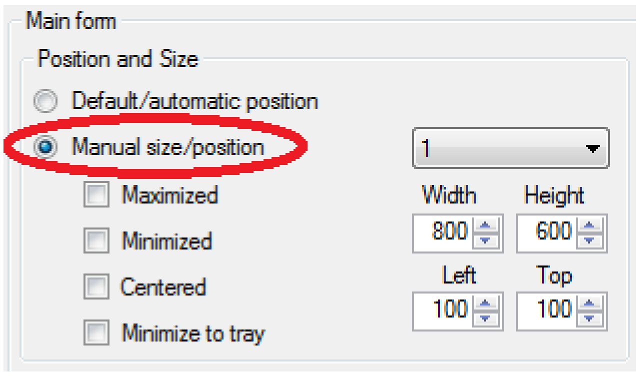

Main form
Position and Size
( ) Default/automatic position
(●) Manual size/position
Monitor: 1
[ ] Maximized
[ ] Minimized
[ ] Centered
[ ] Minimize to tray
Width: 800
Height: 600
Left: 100
Top: 100

Figure 2.602 Settings for manual size and position in the Startup settings panel.

Monitor defines on which monitor the client will start. You can choose between 1 - 8.

Maximized when this box is checked, the client will be maximized at startup on the specified monitor.

Minimized when this box is checked, the client will be minimized at startup on the specified monitor.

Centered when this box is checked, the client will be centered at startup on the specified monitor. When centered is selected, neither *Top* or *Left* can be specified.

Minimize to tray when this box is checked, the client, when it is minimized, will be minimized in the so-called *Tray* that normally is located at the bottom and far right of the screen. If this box is not checked, the client will be minimized as usual to the *Taskbar*. An advantage with minimizing to tray is that you get rid of the blinking icon in the taskbar when the client receives focus, e.g., after having reloaded the configuration.

Width is the desired width of the client when it is started.

Height is the desired height of the client when it is started.

Left is the desired left position of the client when it is started.

Top is the desired top position of the client when it is started.

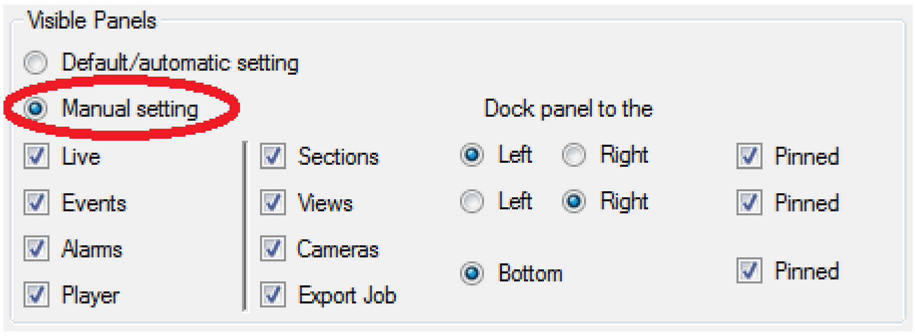
Visible panels

You can choose between *Default/automatic setting* and *Manual setting*.

Default/automatic setting means that all panels will be visible and located at their default positions when the client is started.

Manual setting provides the opportunity to specify which panels to be visible when the client is started.

In Ethiris Client, you can close and open panels at runtime. The settings in this section only determine which panels are open at the startup of the client.



Visible Panels

Default/automatic setting [✓ Manual setting]

Dock panel to the

✓ Live	✓ Sections	● Left	○ Right	✓ Pinned
✓ Events	✓ Views	○ Left	● Right	✓ Pinned
✓ Alarms	✓ Cameras	● Bottom		✓ Pinned
✓ Player	✓ Export Job			

Figure 2.603 Settings for which panels to be open in the panel Startup settings.

Live defines if the panel *Default Live View* will be visible at startup of the client.

Events define if the panel *Events* will be visible at startup of the client.

Alarms define if the panel *Alarms* will be visible at startup of the client.

Player defines if the panel *Player* will be visible at startup of the client.

Sections define if the panel *Sections* will be visible at startup of the client. This is a so-called *tool window* which normally is docked to the left edge of the main window in the client.

Dock panel to the Left/Right determines which side the *Sections* panel will be docked to.

Pinned defines if the panel *Sections* will be pinned at the start of the client. As a default, it is pinned.

Views define if the panel *Views* will be visible at startup of the client. This is a so-called *tool window* which normally is docked to the right edge of the main window in the client.

Dock panel to the Left/Right determines which side the *Views* panel will be docked to.

Pinned defines if the panel *Views* will be pinned at the start of the client. As a default, it is pinned.

Cameras define if the panel *Cameras* will be visible at startup of the client. This is a so-called *tool window* which normally is docked to the bottom edge of the main window in the client. The panel Cameras is normally part of the same tab group as the panel Export Job.

Export Job defines if the panel *Export Job* will be visible at startup of the client. This is a so-called *tool window* which normally is docked to the bottom edge of the main window in the client. The panel Export Job is normally part of the same tab group as the panel Cameras.

Dock panel to the Bottom determines that the *Cameras* and *Export Job* panels will be docked to the *Bottom*.

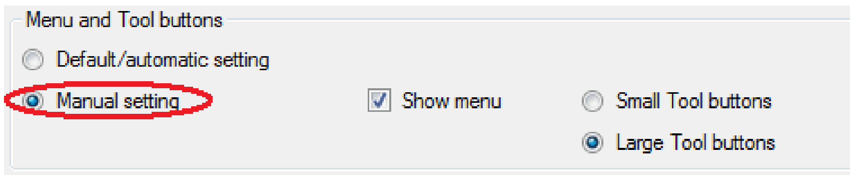
Pinned defines if the panels *Cameras* and *Export Job* will be pinned at the start of the client. As a default, they are not.

Menu and Tool buttons

You can choose between *Default/automatic setting* and *Manual setting*.

Default/automatic setting means that the client is started with the menu visible and with small tool buttons.

Manual setting allows choosing if the menu will be visible or if the tool buttons will be small or large.


Menu and Tool buttons

Default/automatic setting [✓ Manual setting]

[✓ Show menu]

Small Tool buttons [✓ Large Tool buttons]

Figure 2.604 Settings for menu and tool buttons in the panel Startup settings.

Show menu defines if the menu will be visible in the client.

Small/Large Tool buttons define the size of the tool buttons. If the menu is not visible, the tool buttons will be large. There is no other choice.

Startup Views

The purpose of this section is to be able to decide which client view(s) will be displayed when the client is started.

In the list, the window *Default Live View* is always present. If you have defined popup windows, they will appear as separate rows in the list.

For each window, there are three main options; *None/not used*, *Default/automatic,* and *<Client view>*.

None/not used means for *Default Live View* that the live window will be left empty (black) at startup of the client. The operator has to manually select what to display in the window. For possible popup windows, this alternative means that the window will not be opened at startup.

Default/automatic means for *Default Live View* that the first suitable client view will be displayed at startup of the client. For possible popup windows, this setting has no effect but is reserved for future use.

<Client view> means that you select one of the pre-defined client views in the list. For *Default Live View,* this means that the client view will be displayed automatically at startup of the client. For possible popup windows, this means that the popup window will be automatically opened and loaded with the selected client view when the client is started.

2.4.97 Ethiris Mobile configuration node

Under the *Ethiris Clients node,* there might be one or several *Ethiris Mobile* nodes, each one representing an Ethiris Mobile configuration in the system. Not, that to be able to use mobile configurations, you need at least version 5.82 of the Ethiris Mobile application for Android or iOS. Also, note that not all possible configuration options that can be configured in Admin might be possible to use in the mobile app.

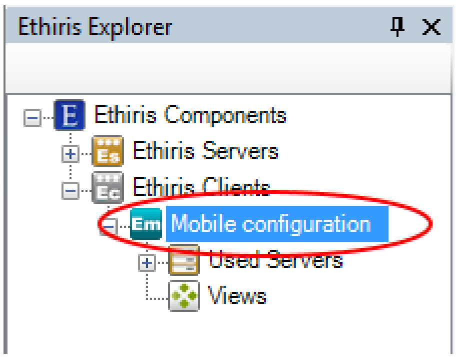


Figure 2.605 The Ethiris Mobile node in Ethiris Explorer treeview.

Ethiris Mobile popup menu

Right-clicking this node brings up a context menu.

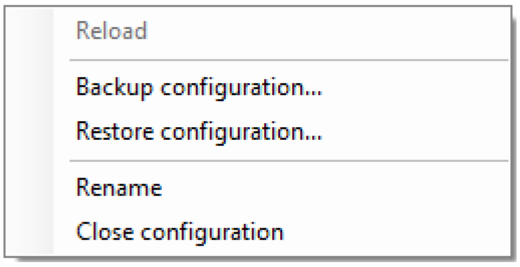
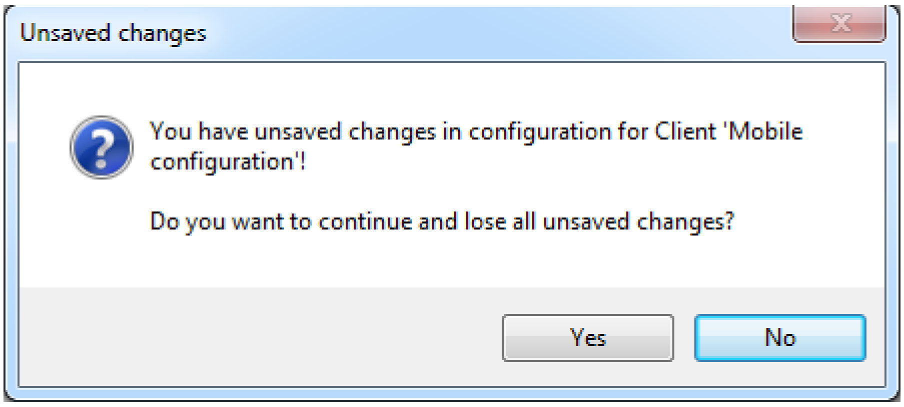


Figure 2.606 The popup menu for the Ethiris Client node.

Reload is only available if you have made changes to the configuration. *Reload* reads the configuration from the server and loads it into Ethiris Admin. Any open panels belonging to the mobile client will be closed. Before the configuration is reloaded, you will be prompted about unsaved changes and have a chance to change your mind.



Unsaved changes

You have unsaved changes in configuration for Client 'Mobile configuration'!

Do you want to continue and lose all unsaved changes?

Figure 2.607 Unsaved changes dialog.

Click *Yes* for reloading anyway or click *No* to <u>not</u> reload and get the chance to save your changes first.

Backup configuration… is more or less the same function as described in section *2.4.2 Ethiris components node* on page *2:44*, with the small difference that in this context backup is performed only for the current Ethiris Mobile configuration.

Restore configuration… is also the same function as described earlier. In this context, it is about restoring the Ethiris Client configuration.

Rename sets the node in the treeview in change mode. You can enter a new name directly in the treeview. You also can press F2 after selecting the node to put it in change mode.

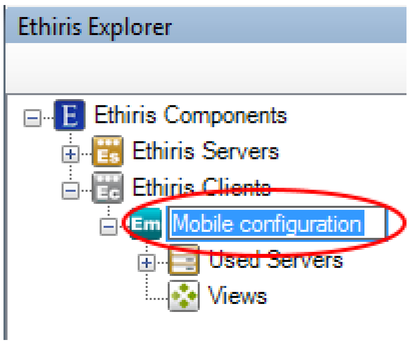
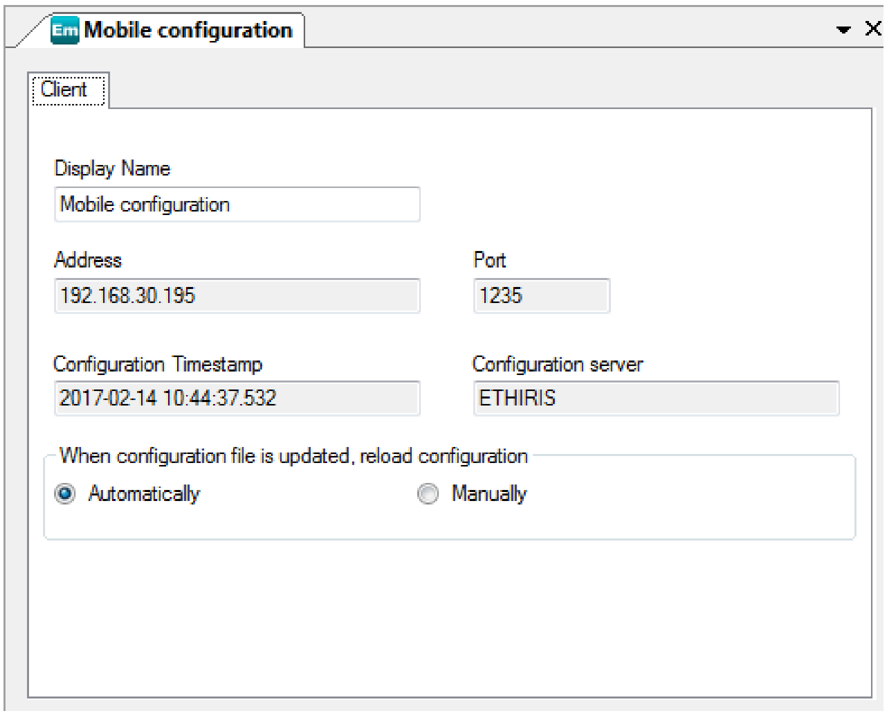


Figure 2.608 Rename client.

Close configuration removes the mobile configuration from Ethiris Admin. Note that the configuration itself is not changed, it's only the current project in Ethiris Admin that's affected.

Ethiris Mobile panel

Double-clicking the *Ethiris Mobile* node in the treeview opens the corresponding panel.



Display Name: Mobile configuration
Address: 192.168.30.195
Port: 1235
Configuration Timestamp: 2017-02-14 10:44:37.532
Configuration server: ETHIRIS
When configuration file is updated, reload configuration
[✓ Automatically] Manually

Figure 2.609 The Ethiris Mobile panel.

The *Ethiris Mobile* panel consists of one tab; *Client*.

Client

Here you find information about the Ethiris Mobile configuration itself.

Display Name is the name you gave the Ethiris Mobile component when adding it to the project. It can be changed. As you change the name, it is immediately updated in the treeview. This name is only used for displaying an appropriate name for the client in the treeview.

Adress is the IP-address to the Ethiris Server that is maintaining this mobile configuration. This field is read-only.

Configuration Timestamp indicates when the mobile configuration was last saved. This field is read-only.

Configuration server is the name of the Ethiris Server that is maintaining this mobile configuration. This field is read-only.

When configuration file is updated, reload configuration determines if clients who are started and have this configuration loaded automatically will reload a new version of the configuration when the configuration is updated.

2.4.98 Used Servers node

Under each Ethiris Mobile node, there is a *Used Servers* node. This is a collection node for all Ethiris Servers that this mobile configuration will connect to.

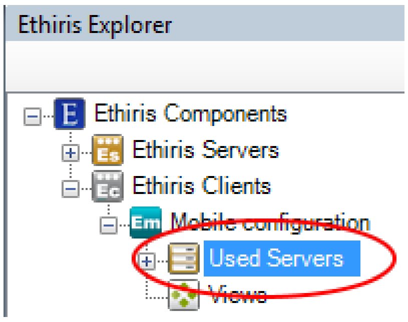


Figure 2.610 The Used Servers node in Ethiris Explorer treeview.

Used Servers popup menu

Right-clicking this node brings up a context menu.

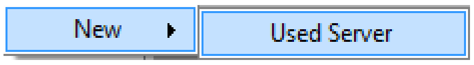


Figure 2.611 The popup menu for the Used Servers node.

New->Used Server brings up a dialog for connecting to an Ethiris Server. The dialog looks different depending on if there are any available Ethiris Servers in the current Admin project. In this case, the Ethiris Server named *Obelix* is available and thus presented in the list of already loaded server configurations.

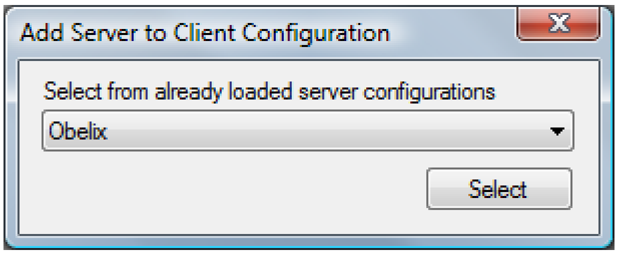
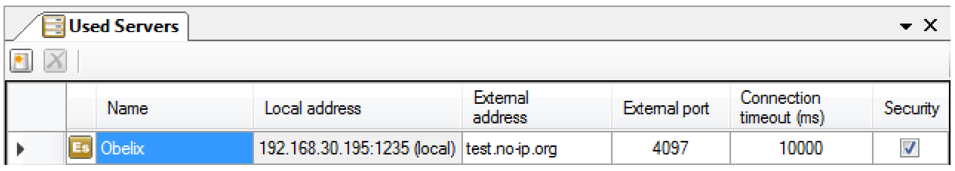

Add Server to Client Configuration
Select from already loaded server configurations
Obelix
Select

Figure 2.612 Add server to mobile configuration dialog.

Select an Ethiris Server in the list and click *Select* for connecting the server to this mobile configuration.

Used Servers panel

Double-clicking the *Used Servers* node in the treeview opens the corresponding panel.



Name	Local address	External address	External port	Connection timeout (ms)	Security
Obelix	192.168.30.195:1235 (local)	test.no-ip.org	4097	10000	✓

Figure 2.613 The Used Servers panel.

This panel consists of a list of all currently selected Ethiris Servers in the mobile configuration.

At the top of the panel, there is a toolbar.

Used Servers panel toolbar

Figure 2.614 The toolbar in the Used Servers panel.

Add a new server

Use this button to add a new Ethiris Server. This is the same as *New->Used Server* in the popup menu for the Used Servers node described above.

Delete selected Server(s)

Use this button to delete the selected Ethiris server(s) from the configuration. You can select more than one server by using the *Ctrl*-button and/or the *Shift*-button.

Used Servers panel server list

The server list consists of several columns.

Name is the desired name of the server. This name has to be unique in the configuration. If you enter an illegal name there will be an error icon to the left of the server in the list indicating the problem. This name is used for identifying the server in various contexts.

Local address is the IP address (and port) the client will use to connect to the server. This field is read-only. It is merely for information on which physical computer the Ethiris Server runs on. The address *127.0.0.1* is an alias for the local computer, i.e., the computer you work on for the moment.

Port is the TCP/IP port that the Ethiris Server listens to for incoming calls from Ethiris Clients. The port has to be determined when the server is connected under *Ethiris Servers* in Ethiris Admin. This is as a default *1235,* and there usually is no reason for changing it.

External address is the IP address the Ethiris Mobile will use to try to connect to the server if the connection to the *local address* fails. This is to be able to use the same configuration both locally and externally. Here you would enter the publicly accessible IP address of the site where Ethiris Server runs. It could also be a dynamic address. If you don’t want to use this feature, leave the field empty.

Ethiris Mobile normally tries to connect using the *Local address*, but if that fails and connection using *External address* and *External port* succeeds, Ethiris Mobile will, after that, use the E*xternal address* at first. If that connection fails, connection using *Local address* will be tried. Ethiris Mobile will try the local and external addresses until a connection is made. When a connection is successful, Ethiris Mobile will remember if it was the local or external address and try that one first the next time the client is started.

External Port is the port mapped in the router that redirects incoming connections to the correct computer on the local network. Refer to the manual of your router for more information regarding this. If not in use, set value to 0 (will be displayed as a blank field).

Note that, currently, Ethiris Mobile doesn’t support clusters.

Connection timeout (ms) is the number of milliseconds a client waits for the Ethiris Server to respond on calls to the server. *10 000* ms is the default, i.e., *10* seconds.

Security determines which Ethiris Server that acts as a *Security server* for this client configuration. The Security server verifies user privileges that are client-specific. These are the last 6 *Operations* in the *Server Security* panel in the server configuration. The 6 operations are *Start client*, *Allow export of video from client*, *Show player in client*, *Show event list in client*, *Show alarm list in client* & *Exit client*. You can read more about Server Security in section *Security* node on *page 2:255*. Only one server can be the security server. In the case of several used servers in the client configuration, you can check the desired Ethiris server that will act as a Security server.

2.4.99 Used Server node

Under the Used Servers node, all used Ethiris Servers are listed. Each node represents an Ethiris Server.

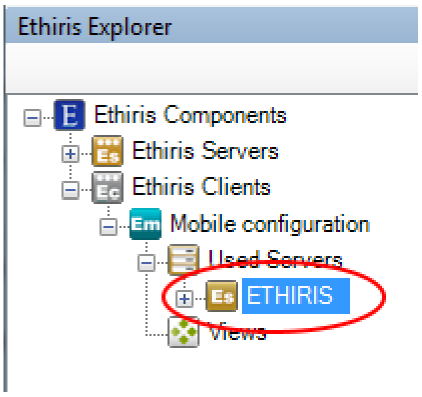


Figure 2.615 A Used Server node in Ethiris Explorer treeview.

Used Server popup menu

Right-clicking this node brings up a context menu.

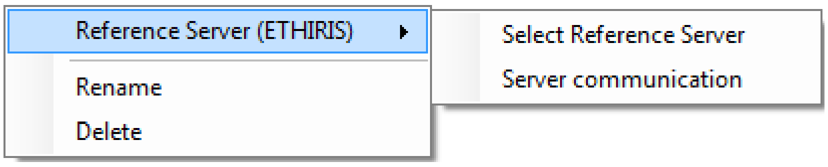


Figure 2.616 The popup menu for a Used Server node.

Reference Server (<Name>)->Select Reference Server brings up a dialog for selecting an Ethiris Server. If there is a *name* in parenthesis, it means that a reference already exists. If no reference exists, it means that Ethiris Admin has no idea of which cameras and I/Os that are available in the Ethiris Server.

When you first add an Ethiris Server to the used servers list, the reference is set automatically. The only reason a reference should not exist is that the mobile configuration has been opened in Ethiris Admin before the corresponding Ethiris Server configuration has been opened in Ethiris Admin. In this case, you can connect to the Ethiris Server, then select it as a reference for the corresponding used server in the client configuration. You can also use this menu item for changing the referenced server to another Ethiris Server that is loaded into Ethiris Admin.

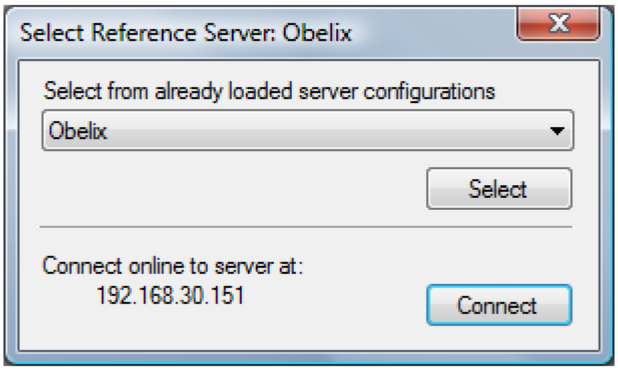
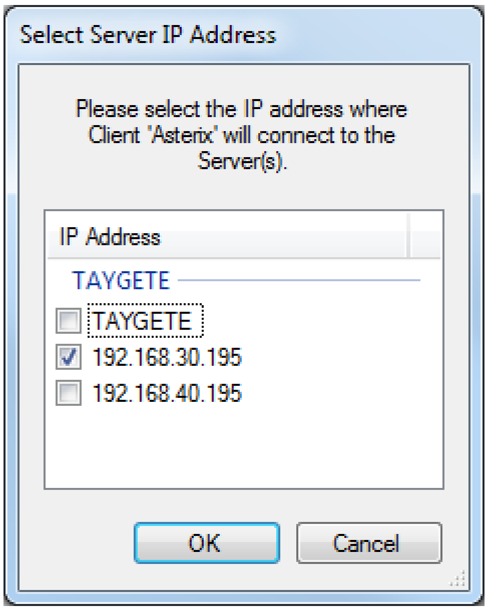

Select Reference Server: Obelix
Select from already loaded server configurations
Obelix
Select
Connect online to server at:
192.168.30.151
Connect

Figure 2.617 Select reference server dialog.

Select the desired server in the list and click *Select* alternatively click *Connect* to select the local Ethiris Server as reference.

Reference Server (<Name>)->Server communication brings up a dialog for selecting which IP address to use when connecting to the Ethiris Server(s). This is important when a server has multiple IP addresses.


Select Server IP Address

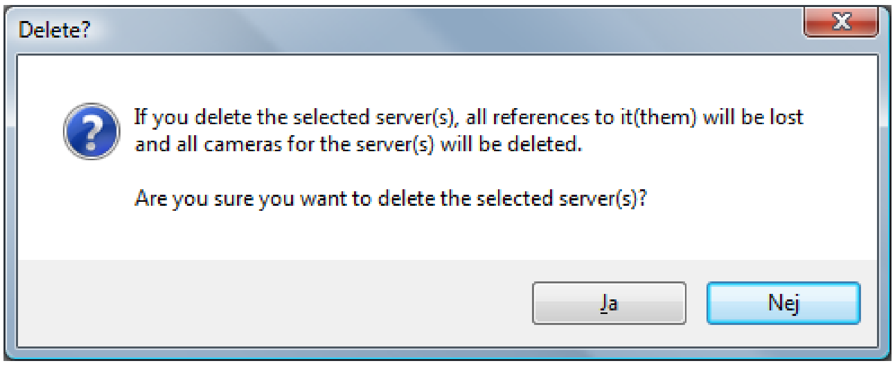
Please select the IP address where Client 'Asterix' will connect to the Server(s).

IP Address
TAYGETE
- [ ] TAYGETE
- [x] 192.168.30.195
- [ ] 192.168.40.195

Figure 2.618 The dialog for Server communication."TAYGETE" is the name of the computer which the Ethiris Server is running on.

Rename sets the node in *rename* mode. You can enter a new name for the server directly in the treeview.

Delete removes the Ethiris Server from the used servers list for the mobile configuration. If you delete the server, all references to cameras and I/Os belonging to that server will disappear from the client configuration. A view, for example, will be kept with the layout intact, but all camera references in the camera views will be deleted. Before deletion, a confirm dialog is displayed.


Delete?

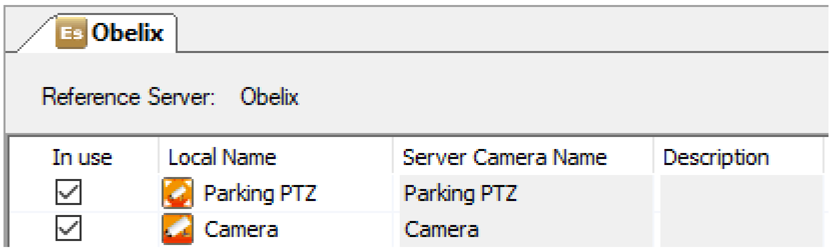
If you delete the selected server(s), all references to it(them) will be lost and all cameras for the server(s) will be deleted.

Are you sure you want to delete the selected server(s)?

Figure 2.619 Dialog for confirming the deletion of the used server.

Used Server panel

Double-clicking the *Used Server* node in the treeview opens the corresponding panel.



Reference Server: Obelix

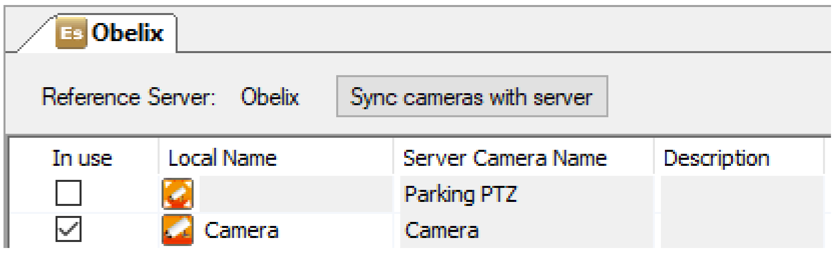
In use	Local Name	Server Camera Name	Description
✓	Parking PTZ	Parking PTZ	
✓	Camera	Camera	

Figure 2.620 The Used Server panel.

This panel consists of a list of all available cameras in the referenced Ethiris Server.

At the top of the panel, the name of the currently referenced Ethiris Server is displayed.

A button, ”Sync cameras with server”, will appear if one or more cameras are not checked for usage. If pressed, it will automatically enable all cameras for usage and enter the server’s camera names as the local names.

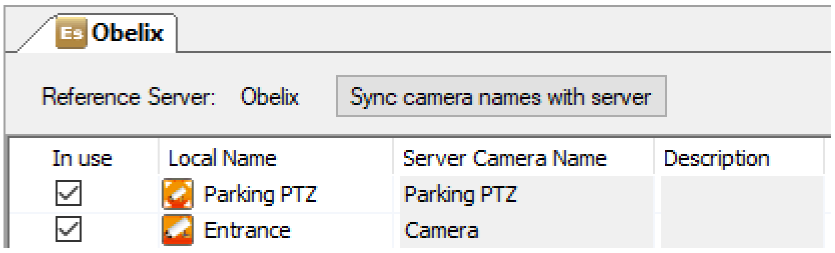


Reference Server: Obelix

In use	Local Name	Server Camera Name	Description
☐		Parking PTZ	
✓	Camera	Camera	

Figure 2.621 The Used Server panel with the button ”Sync cameras with server” visible, due to a camera not being in use.

If any local camera name does not match the server camera name, another button will appear, ”Sync camera names with server”. When pressed, it will change all local camera names to the corresponding server camera name.



Reference Server: Obelix

In use	Local Name	Server Camera Name	Description
✓	Parking PTZ	Parking PTZ	
✓	Entrance	Camera	

Figure 2.622 The Used Server panel with the button ”Sync camera names with server” visible, due to a different local name than the server camera name.

Used Server panel camera list

The camera list consists of several columns.

In Use*.* Check this box to use the corresponding camera in the mobile configuration. This means that the camera is available for camera views and in the *Cameras panel* in Ethiris Mobile. When first adding a used server, all cameras are selected. If you add more cameras to the server configuration after the Ethiris Server is referenced in the mobile configuration, you have to manually select the new cameras by checking the *Use* column.

Local name is the name of the camera you want in Ethiris Mobile. This need not be the same as the name used for the camera in Ethiris Server. The local name has to be unique. The column also contains an icon that indicates which type of camera this is.

Fixed camera	This icon indicates a fixed camera. The icon is also used for cameras that only exist in the client configuration (but don't reference any camera in the server configuration), as it is then not possible to know what type of camera it would be.
PTZ camera	This icon indicates a camera that supports PTZ in some way. This also means that cameras only supporting optical zoom will get this icon.
360 camera	This icon indicates that the camera is a 360 camera that Ethiris supports. Note that it isn't possible to dewarp the image in Ethiris Mobile at the moment.

Server camera name is the name used for the camera in Ethiris Server. This field is read-only.

Description is an optional description entered for the camera in the Ethiris Server configuration. This field is also read-only.

2.4.100 Camera node

Under each *Used Server* node in the treeview, all available cameras in the Ethiris Server are presented as treeview nodes.

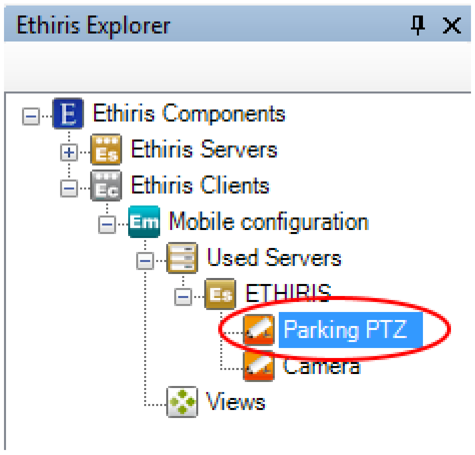


Figure 2.623 A Camera node in Ethiris Explorer treeview.

Camera Change order

The order in which the cameras are presented in the treeview is used in various contexts where a list of cameras is presented, e.g., when you select a camera into a camera view or in the camera list in Ethiris Mobile.

You can rearrange the order of cameras in the treeview by dragging a camera to another position. Click the desired camera with the left mouse button, hold the button down and, at the same time, move the mouse pointer to another camera in the list, release the mouse button when the mouse pointer is at the desired position in the treeview.

When a camera is moved down in the treeview, the camera will be positioned *after* the camera it was dropped on. If the camera is moved up in the treeview, the camera will be positioned *before* the camera it was dropped on.

Camera popup menu

Right-clicking this node brings up a context menu.

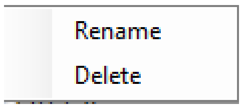


Figure 2.624 The popup menu for a Camera node.

Rename sets the node in rename mode. You can change the camera name directly in the treeview. This is the same as changing the *Local name* in the Used Server camera list.

Delete deselects the camera for this client configuration. This is the same as unchecking the *Use* checkbox in the Used Server Camera list.

Camera panel

Double-clicking a *camera* node in the treeview opens a panel that is the same as *Used Server* panel.

2.4.101 Views node

Under the Ethiris Mobile node, there is a *Views* node. This is a collection node for all *Views* and *Buttons* in the mobile configuration.

Views are pre-defined views with one or several *camera views*. A camera view is usually used for displaying live video from a pre-defined camera.

Buttons are used for activating signals/variables in the Ethiris Servers data store. Examples are activation of a preset position or a guard tour.

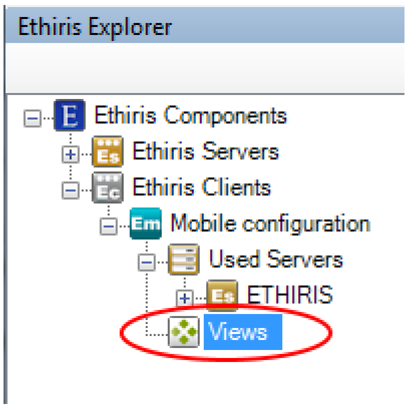


Figure 2.625 The Views node in Ethiris Explorer treeview.

Views popup menu

Right-clicking this node brings up a context menu.

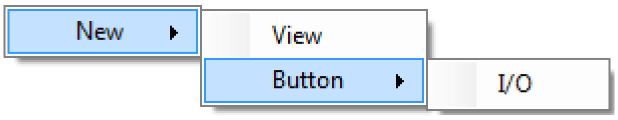


Figure 2.626 The popup menu for the Views node.

New->View adds a new view to the mobile configuration. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the view directly in the treeview.

The purpose of a view is to pre-configure a layout of camera views. The view can be displayed in Ethiris Mobile in live views

New->Button->I/O adds a new button to the mobile configuration. It is immediately added to the treeview, and the new node is set in *rename* mode. You can enter the desired name of the button directly in the treeview.

The purpose of a button is to connect it to a writeable *Boolean* variable in Ethiris Server’s data store. When you click the button in Ethiris Mobile, the corresponding variable is activated. You can use this functionality for many different purposes, e.g., activate a preset position for a PTZ camera, start recording, or sending an email.

Views panel

There is no panel for the Views node.

2.4.102 View node

Under the *Views* node, there may be one or several *View* nodes.

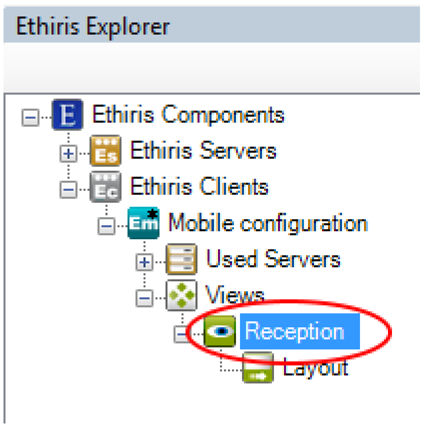


Figure 2.627 A View node in Ethiris Explorer treeview.

View popup menu

Right-clicking this node brings up a context menu.

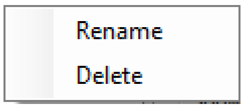


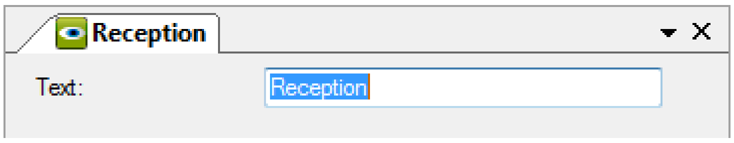
Figure 2.628 The popup menu for a View node.

Rename sets the node in *rename* mode. You can enter the new name for the view directly in the treeview.

Delete removes the view from the client configuration. A confirmation dialog will be displayed before deletion.

View panel

Double-clicking a *View* node in the treeview opens the corresponding panel.


Reception

Text: Reception

Figure 2.629 The View panel.

This panel contains properties for the view, such as text for the corresponding view button that will be displayed in Ethiris Mobile in the *Views panel* and the *I/O-buttons panel*.

Text determines which text will be displayed in the corresponding view button in Ethiris Client.

The image of the button is not configurable, but instead is the appropriate image automatically selected based on the layout of the view.

2.4.103 Layout node

Under each View node, there is a corresponding *Layout* node.

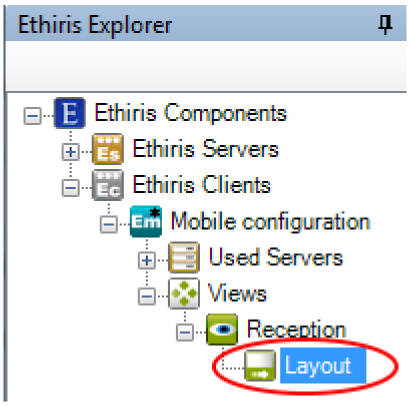


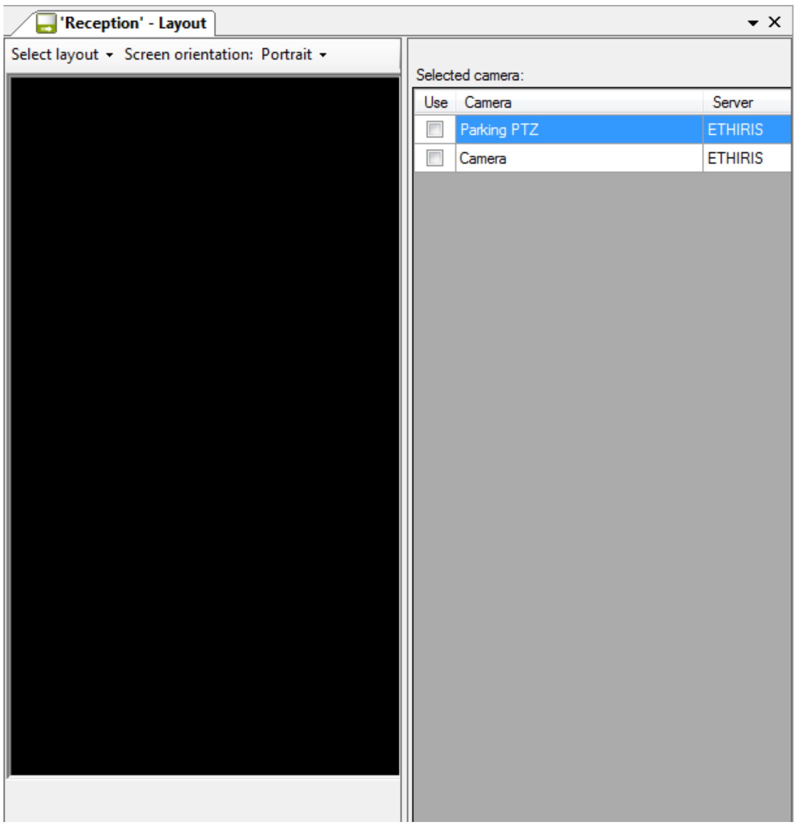
Figure 2.630 A Layout node in Ethiris Explorer treeview.

Layout popup menu

There is no popup menu for this node.

Layout panel

Double-clicking a *Layout* node in the treeview opens the corresponding panel.



Use	Camera	Server
☐	Parking PTZ	ETHIRIS
☐	Camera	ETHIRIS

Figure 2.631 The Layout panel.

This panel is divided into two parts; left and right. On the left side is a representation of a screen where the view will be displayed, and on the right side are properties for the currently selected camera view in the left side.

Select layout is a list of pre-defined layouts from which you can choose.

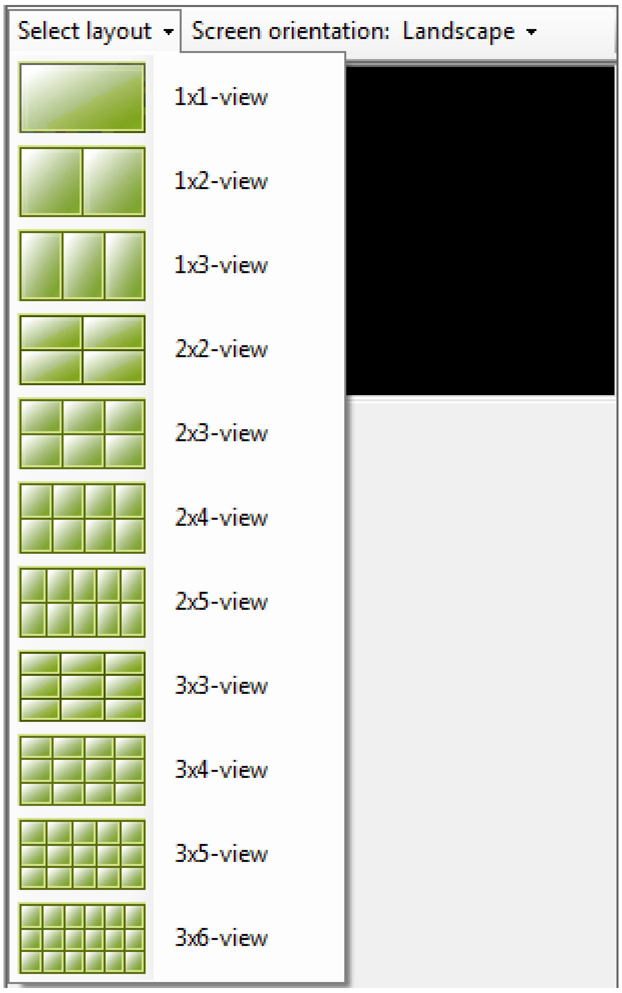


Figure 2.632 Pre-defined layouts in landscape mode.

This is a quick way to create a layout.

Screen orientation sets the orientation of the screen when you design the view. If you rotate the phone when the view is shown, the layout will automatically adjust to the new orientation. Switching between *Landscape* and *Portrait* modes lets you see how the view will behave when displayed on the phone.

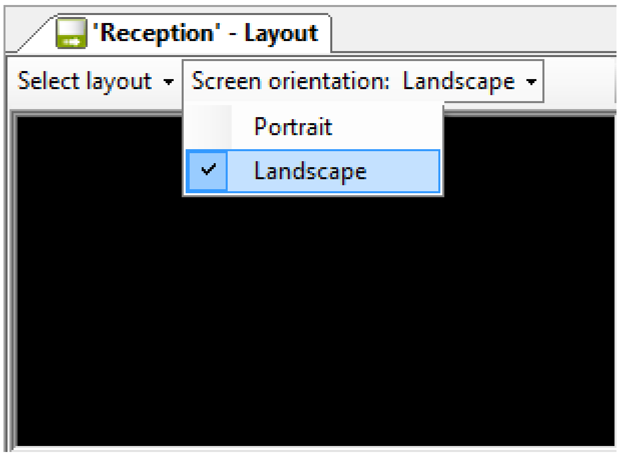


Figure 2.633 Screen orientation for the camera views.

Camera view popup menu

Right-clicking a camera view brings up a popup menu.

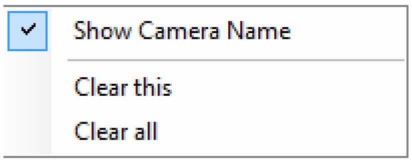


Figure 2.634 Camera view popup menu.

Show Camera Name determines if the name of the camera will be displayed in live when the Ethiris Mobile is started. The operator can change this in the settings panel in Ethiris Mobile.

Clear this deletes the content in the selected camera view.

Clear all deletes the content in all camera views in the current view.

Camera view type

There is only one type of camera view in the mobile configuration; *Camera*.

View type Camera

This is the default view type that every camera view has. It is used for displaying live video from a specific and pre-defined camera.

When no camera is selected for such a camera view, the color of the camera view is black. In the example below, only the first camera view has a defined camera. The other 5 camera views have no cameras defined and are thus black.

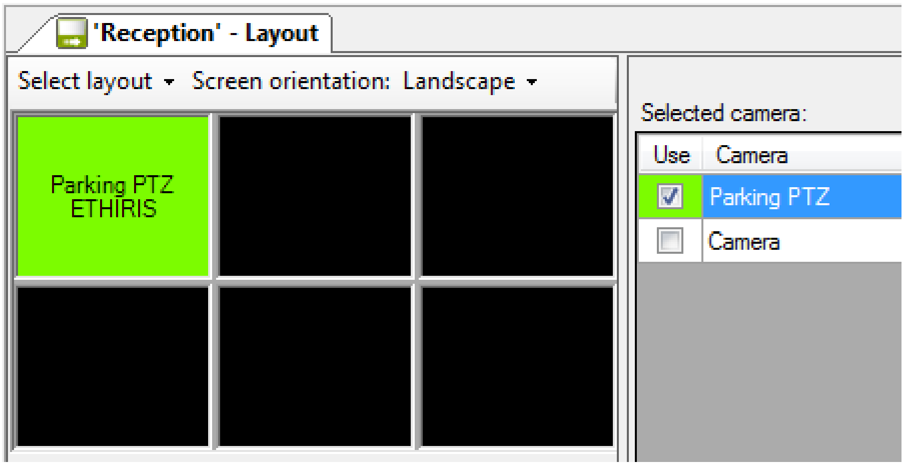
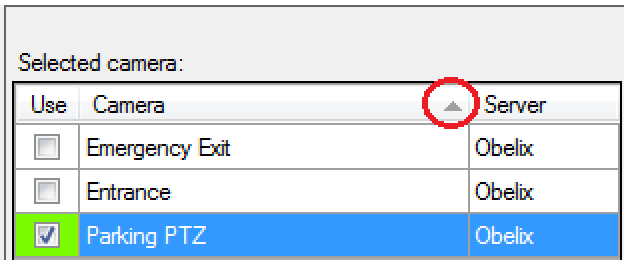


Figure 2.635 View type camera.

To the right of the camera views, is a list of available cameras. This list consists of the cameras you have selected for the *Used Servers*. See section *Used Server panel* on page *2:408* for more information about that.

Simply check the camera in the list which you want to display live video from in the selected camera view.

If you have many cameras, it can be convenient to sort them in alphabetical order. Click the *Camera* column header to do this. Click again to reverse the sort order. An arrow indicates the current sort order.


Selected camera:

Use	Camera	Server
☐	Emergency Exit	Obelix
☐	Entrance	Obelix
☑	Parking PTZ	Obelix

Figure 2.636 Sort the cameras by clicking the column header.

Add several cameras at once.

Another useful hint for adding many cameras to your layout is to select multiple cameras in the list and then drag them with the right mouse button. Release the mouse button on the first camera view you want to fill with cameras.

You can select cameras in several different ways. If you don’t care about the order, you can simply click a row in the list with the left mouse button, hold the button down while moving the mouse over more rows, and finally release the mouse button when you have selected the desired cameras. In this case, the camera you clicked first will end up in the first camera view, and the following cameras will be placed in the order you selected the cameras.

You can also select cameras one by one by pressing the *Ctrl* key when selecting camera no 2 and forward. Then the cameras will be added to the camera views in the order you selected them.

Finally, when dragging the cameras into the camera views, you have to use the right mouse button. Release the mouse button on the first camera view you want to fill. The cameras will be added from left to right and from top to bottom.

You can change the order of the cameras.

If you want, you can change the order of the cameras in the camera views. Press the *Shift* key at the same time you drag a camera view with the left mouse button and release it on another camera view. The two camera views will then swap content.

Controls

For the time being, controls are not supported in Ethiris Mobile.

3 Script

3.1 Principles of the Scripting language

In this section, anyone who has not used a language such as JavaScript, Java, C, or C++ can get an introduction to basic syntactical constructions that are common to these program languages. You can also get an introduction to the differences between ECMAScript and Java, C, and C++.

This section cannot replace a JavaScript or JScript manual. However, the examples in this section are linked to camera surveillance applications where possible. This means that it may make for interesting reading even if you are familiar with the programming language.

To get started, you need to know how to edit and run scripts in Ethiris. This is described in the first section. This is followed by a few sections that cover the most basic elements of the scripting language and a few important troubleshooting techniques.

3.1.1 Editing and Running Scripts in Ethiris

To test the script in this section, enter it in the script editor. The language's reserved words are displayed in blue text. String literals are displayed in red, and comments are displayed in green.

If a line contains a syntax error, it is highlighted with a red underline. Please note that the line in which the syntax error is detected need not necessarily be the one causing the error.

When you edit scripts, you can use the *Variable Browser* to see which variables the data store contains. A readable variable can, for example, be a schedule's variable for the current state. A writeable variable can be a camera's variable for starting recording. You can drag and drop variables from the browser lists to your scripts to insert the name of the variable in the script. You can also double-click a variable in the browser to achieve this.

The script is executed continuously at the interval set in the text box for *Execution interval.* If the execution interval is selected as 100 ms, the script will be executed 10 times per second. Remember that, if a script performs many demanding calculations, there is no guarantee that it will be executed as often as intended. The system variable *System.Script.ExecutionTime* states the current execution time.

3.1.2 Basic Syntax

Variables are declared with ”var”. As values with different data types can be stored in a specific variable, no data type is specified in the variable declaration.

Statements in the scripting language are ended with a semicolon, ”;”. Blocks in the scripting language are defined with special parentheses, ”{” and ”}”. A block is a statement that consists of several other statements.

An assignment is done with an equals sign, ”=”. Comparison is done with a double equals sign, ”==”. Do not mix these up!

Example:

```
var b = true;
var c = 13;
if(b == true)
{
   b = false;
   c = c + 1;
}
```

If you want to make a comparison that is true if two values are different, write ”!=”. If you want to make a comparison that is true if two values are equal and have the same type, write ”===”. The reverse operation is ”!==”.

Boolean expressions are AND ”&&”, OR ”||” and NOT ”!”. Example:

```
sumMotion = motion1 || motion2 || motion3;
```

The individual mathematical operations are addition: ”+”, subtraction: ”-”, multiplication: ”*”, division: ”/” and modulo: ”%”. Example:

```
angle = (angle + 1) % 360;
```

Just as in C/C++/Java, you can increase and reduce a variable’s value by one by using the operators ”++” and ”--”. Example:

```
angle = (angle++) %360;
```

In many cases, the semicolon can be omitted. Instead, you can let a line break indicate where a statement ends. However, many JavaScript programmers recommend using a semicolon to avoid the script engine interpreting your script other than you intended.

Do not mix up ”=”, used for assignment, with ”==”, used for comparison. It gets silly in the following example:

```
var b = false;
if(b = true)
{
   ...
}
```

The code in the if-statement will always be executed as b is assigned the value *true* in the expression in the if-statement.

Comments

You can enter comments that explain and clarify details in your script. There are single-line and multi-line comments. This works exactly as in C/C++/Java.

To create a comment that continues to the end of the line, enter ”//”.

```
// Initialise counter r
var r = 0;

var g = (1 + Math.sqrt(5))/2; // The golden ratio
```

To create a comment that may extend over several lines, start the comment with ”/*” and end it with ”*/”. This is sometimes also used to comment out parts of the code.

```
if(motion1 /*&& motion2 && motion3*/)
{
   ...
}
```

3.1.3 Flow Control

This section describes which statements in the scripting language can be used for flow control.

Flow control can be skips forwards, i.e., skipping over certain statements. This category includes the if-else statement. These are harmless to use.

It may also involve skips backward, i.e., executing certain statements again. This can be used to create loops in the program code, such as while and for statements. The problem is that you can create endless loops, which results in the program ”hanging”. This is undesirable behavior, least of all in a surveillance system. So the reader is urged to use loops with the utmost caution.

Example of an endless loop:

```
while(true);
```

if-else

An if-statement contains another statement that is only executed if the expression stated is true. If several statements are to be executed when the expression is true, they can be gathered in a block.

You can also enter a statement that is to be executed if the expression is false.

```
// If motion in camera Utb2
if(Utb2.motion2.Motion)
{
   // Start recording for camera Utb1
   Utb1.RecordEvent = true;
}
else
{
   // Stop recording
   Utb1.RecordEvent = false;
}
```

do-while

This is an iterative statement that executes another statement while an expression is true. However, the statement is executed at least once.

Example:

```
var result = 0;
do
{
   result++;
}
while(result < 12)
```

while

This is an iterative statement that executes another statement while an expression is true.

```
var result = 0;
while(result < 12)
   result++;
```

for

For loops exist in two versions in the scripting language. One of them works in the same way as the equivalent in C/C++/Java.

Example:

```
for(var i = 0; i < 12; i++)
{
   ...
}
```

for-in

For-in loops are used to allocate a variable name to the properties of an object in the correct order.

The following loop creates a string containing the name and value of variables in the data store.

```
var s = "";
for(var variableName in DataStore)
{
   s += "The variable " + variableName;
   s += " has the value " + DataStore[variableName];
   s += "\n";
}
```

Together with Sequence objects, this offers a simpler way to create for loops that use integers. The following script is equivalent to the example in the section on for loops.

```
for(var i in Sequence(0, 11))
{
   ...
}
```

Properties that have the attribute DontEnum are automatically skipped by for-in loops.

A for-in loop cannot result in an endless loop and can, therefore, be used to good effect instead of for loops.

continue

A continue statement is used to continue to the next cycle in the loop.

The following script creates a string containing the names of all variables belonging to the camera ”Utb1” in the data store.

```
var s = "";
for(var variableName in DataStore)
{
   if(variableName.indexOf("Utb1") != 0)
      continue;
   s += "The variable " + variableName;
   s += " has the value " + DataStore[variableName];
   s += "\n";
}
```

break

A break statement is used to interrupt a loop.

return

A return statement is used to interrupt the script’s execution of a function and possibly return a value.

switch

Switch statements work in the same way as in C/C++/Java. Depending on the value of an expression, they execute different statements. If a statement does not interrupt the flow with a *break* or *return*, it will continue to the next.

Example:

```
switch(currentIndex)
{
   case 0: Utb2.PTZ.Pos1.Preset = true; break;
   case 1: Utb2.PTZ.Pos2.Preset = true; break;
   case 2: Utb2.PTZ.Pos3.Preset = true; break;
}
```

If you do not use a *break* or *return*, you may get unexpected effects.

Example:

```
var s = "Current position: ";
switch(currentIndex)
{
   case 0: s += "Door";
   case 1: s += "Pool";
   case 2: s += "Roof";
   default: s += "Unknown";
}
```

If the variable currentIndex has the value 0, the string s will have the value ”Current position: DoorPoolRoofUnknown”, which is probably not the desired result. This is solved by interrupting the flow with a *break***.**

```
var s = "Current position: ";
switch(currentIndex)
{
   case 0: s += "Door"; break;
   case 1: s += "Pool"; break;
   case 2: s += "Roof"; break;
   default: s += "Unknown"; break;
}
```

3.1.4 Alert and Debug.Print

Do not use the alert function!!!

The alert-function was earlier used to print a message in the *Debug* window and has been replaced by Debug.Print. It is still possible to use alert, but it is recommended to use Debug.Print instead. The print function can be used to make it easier to see what is going on during runtime.

Instead, you can use a string variable that you watch in the *Watch* panel. In the following examples, the variable *sResult* will appear from time to time. The idea is that this variable should be added to the Watch panel so the result can be studied there.

3.1.5 Try-catch

The *try-catch* construction is an important aid when looking for errors in your scripts.

When something goes wrong in ECMAScript, an exception is often thrown. If you do not catch this exception, the script’s execution is interrupted, and an error message is written to the log file for Ethiris Server.

Let us say, for example, that you have problems with the script never starting to record for a camera when a specific condition is met. Your script looks roughly like this (replace the three dots with any condition):

```
Utb1.RecordEvent = ...
```

To see whether your calculation throws an exception, safeguard it with a *try-catch* statement.

```
try
{
   Utb1.RecordEvent = ...
}
catch(e)
{
   sResult = e.message;
}
```

An exception is usually an *Error* object. Such objects have a single important property, *message*. It contains a message stating what went wrong.

With the code above, *sResult* will contain the error message if something went wrong.

3.1.6 Throw

If, for some reason, you want to throw an exception yourself, you do it using the keyword *throw*. It is appropriate to throw an *Error* object:

```
var error = new Error("Something went wrong");
throw error;
```

3.1.7 Values

In the scripting language, you can use three different types of primitive values. These are numbers, strings, and Boolean values.

Type	Explanation	Storage in memory
Number	Number	64-bit floating-point
String	String	UNICODE string consisting of 16-bit characters
Boolean	Boolean value	true or false

12, 14.1, 1.5e-19, and 12.3E14 are all examples of numbers.

"Hello, world!", "Русский", and "中文" are all examples of strings. Thanks to the UNICODE standard, characters from many different languages can be used. You create strings by enclosing a text with quotation marks.

True and *false* are examples of Boolean values.

There are also three other types of values. These are *object*, *null,* and *undefined*. They are explained in the next section.

Type	Explanation	Storage in memory
Object	Object	A dictionary, mapping strings to values

Null	Reference to ”no object”	Null
Undefined	Undefined value	undefined

Number

To create numbers in the scripting language, you enter a number in decimal notation, scientific notation or hexadecimal notation:

12 and 12.4 are examples of decimal notation.

1e12, 1.3e5, 33E-2 and 412.4E7 are examples of scientific notation.

0xF01B, 0X120C and 0xffb0c are examples of hexadecimal notation.

A number can also have the value ∞ (infinity) or -∞ (minus infinity). For example, the result of dividing a positive number by zero is infinity. To create a number with the value infinity, write *Infinity*. To create a number with the value minus infinity, write -*Infinity*.

A number can also have the value *NaN*, Not a Number. This value indicates that a calculation has failed. For example, the result of dividing zero by zero is *NaN*. To create a number with the value *NaN*, write NaN.

The script engine can keep track of both +0 and -0. For most practical applications, these have the same value, but in some calculations, there may be a difference. For example, 1/(+0) =Infinity, while 1/(-0) = -Infinity.

Strings

To create strings in the scripting language, you write a text within quotation marks. Some characters are difficult to enter via the keyboard. By entering a backslash (\), followed by an *escape sequence*, you can enter special characters.

Character	Escape Sequence
Backspace	\b
Horizontal tab	\t
New line	\n
Vertical tab	\v
Form feed	\f
Carriage return	\r
Quotation marks	\”
Single quotation mark	\’
Backslash	\\

The following example creates two lines of text in the string (”Hello” and ”How are you?”) as you have entered a line break in the string. But it will be visible first when, e.g., writing the string to a file.

```
var s = "Hello,\nHow are you?";
```

You can also use single quotation marks. This makes it easier if you want to enter a string that is to contain double quotation marks and vice versa.

```
'The capital of Denmark is called "København" in
Danish'
```

Boolean values

You create Boolean values quite simply by writing *true* or *false* in your script.

Conversion of values

Values in the scripting language can be converted to other types automatically using the following tables.

Conversion to Boolean values:

Type	Boolean
Undefined	false
Null	false
Boolean	
Number	false if the number is +0, -0 or NaN, true otherwise
String	false if the string is empty (the length is 0). true otherwise
Object	true

Conversion to numbers:

Type	Number
Undefined	NaN
Null	+0
Boolean	1 if the value is true, +0 if the value is false
Number	
String	The string’s content parsed as a number or NaN if the parsing fails
Object	Call the function valueOf() for the object

Conversion to strings:

Type	String
Undefined	”undefined”
Null	”null”
Boolean	”true” if the value is true, ”false” if the value is false
Number	A string that contains the number
String	
Object	Call the function toString() for the object

When you want to convert an object to a number, the function valueOf() is therefore called behind the scenes, just as if you had written ***obj.valueOf()***. The equivalent applies when you want to convert an object to a string.

obj.toString();

To force a conversion of a *value* from one type to another, enter *Boolean(value), Number(value)* or *String(value)*, depending on the type to which you want to convert.

You can also convert to an object by writing *Object(value)*. If *value* is already an object, *value* is returned. Otherwise, the script engine tries to convert the *value* to an object.

Example:

When you add a string with a value, a new string is returned.

```
"12" + 3
```

becomes "123".

But

```
Number("12")+3
```

becomes 15.

As an object is always converted to true, while null and undefined are converted to false when they are to be converted to a Boolean value, you can use an if statement to find out whether a specific object exists.

3.1.8 Object

Objects exist so that you have somewhere to store your values. An object works like a reference book. You can look up a search word and find a value. You can also add new search words to objects and store values there.

Properties

We say that objects have properties. A property is quite simply an item in the reference book, i.e., a search word and the value stored there. The property, therefore, has a name and a value. The value of a property can be another object.

Example:

To retrieve the value of the property ”PI” in the object ”Math”, you write:

```
Math.PI
```

or

```
Math["PI"]
```

This property is a number, the value of π: the ratio between the circumference of a circle and its diameter.

To create a new property in the object ”Math” and store the value of the golden ratio in it, you can write:

```
Math.goldenRatio = 1.6180339887499
```

or

```
Math["goldenRatio"] = 1.61803398887499
```

But where is the object ”Math” stored? The explanation is that there is a global object that has a property called ”Math”, the value of which is an object that contains usable mathematical values and functions. When you write ”Math”, the script engine searches through the global object for a property called ”Math” and returns its value. The text ”Math” in your script is called an identifier.

When you declare a variable by writing, for example, *var myVariable*, a new property called *myVariable* is created in a separate object called the variable object. The variable object is also searched through when you write an identifier. Thanks to this, you can access the value of the variable from then on.

Let us look at the script:

```
var myVariable;
myVariable = 5;
```

What actually happens when the script engine runs this script? In the first line, a new property is created in the variable object with the name ”myVariable” and the value *undefined*. In the second line, the script engine searches through the variable object and the global object for a property with the name ”myVariable” and stores the value *5* in the property.

In addition to names and values, properties also have attributes. These attributes can be one or more of *DontEnum*, *DontDelete,* and *ReadOnly*. The simplest of these to explain is ReadOnly, which means that you cannot change the value of the property.

The property ”PI” in the object ”Math” has the attribute ReadOnly. So the following statement has no effect:

```
Math.PI = 4;
```

To remove a property from an object, you write *delete*. For example, we can remove the golden ratio from the object ”Math”:

```
delete Math.goldenRatio;
```

Properties with the attribute DontDelete cannot be removed. The property ”PI” in the object ”Math” has the attribute DontDelete. So the following statement has no effect:

```
delete Math.PI;
```

The statement *delete* returns *true* if the property could be removed. Otherwise, it returns *false*.

```
if(delete Math.PI)
{
   // We never get here. delete returns false
}
```

Properties with the attribute DontEnum are skipped by for-in loops. There is more on this in the section on flow control.

Undefined

If you try to retrieve a property that does not exist in the object, you get the value undefined. If you want to create a value of the type undefined in your script, you simply write *undefined*.

```
if(Math.goldenRatio == undefined)
{
   Math.goldenRatio = (1+Math.sqrt(5))/2
}
```

Creating objects

To create new objects, you use the keyword new. When you have created a new object, you can add properties to it.

```
var myObj = new Object();
myObj.name = "Kalle Kula";
myObj.address = "Kulgatan 9";
```

You can also create new objects by using an object literal.

```
var myObj = { name: "Kalle Kula", address: "Kulgatan
9" };
```

There are other types of object, for example File and Array. To create objects of a specific type, you write, for example:

```
var myFile = new File("Test.txt", 4);
var myArray = new Array();
```

File objects are used to read from and write to text files.

Array objects are like normal objects, but the names of the properties are numbers, and they always have a property called ”length”. The value of length is always one greater than the greatest name of any property in the Array object. The following example stores the value 12 in the property with the name ”0” and the value 9 in the property with the name ”5”. The length is 6.

```
myArray[0] = 12;
myArray[3] = 2;
myArray[5] = 9;
if(myArray.length == 6)
{
   // We get here as the length property is 6
}
```

Another way of creating new arrays is to use an array literal. The following example creates the same array as the previous example.

```
var myArray = [12, , ,2, , 9];
```

Null

If you want to indicate that a property can be an object, but currently have no object to use, you store the value null in the property. You create the value null by writing *null* in your script.

3.1.9 Function

Functions are objects that can do something. To make a function do its job, you call the function by using parentheses ”(” and ”)”.

The property ”sin” in the object ”Math” is a function that calculates the sine of a value. You call it by writing, for example:

```
var v = Math.sin();
```

This function expects an argument. Arguments are entered between parentheses, separated by commas where you want to enter several arguments. In this case, the argument will be the number of radians for which you want to calculate the sine.

```
var v = Math.sin(Math.PI/2);
```

A function that can take several arguments is the property ”max” in the object ”Math”. The following function call returns the greater of two numbers, in this case, 2 and 5.

```
var v = Math.max(2, 5);
```

We usually say that we call the function max in the object Math. This is because the object containing the function has a special role to play. The function’s script code can access the object in which it calls by writing *this*.

If you want to create a separate function, you use the keyword *function*.

Example:

```
var myObj = new Object();
myObj.addProperty = function(name, value)
{
   this[name] = value;
}

myObj.addProperty("name", "Kalle Kula");
myObj.addProperty("address", "Kulgatan 9");
var s = "";
s += "name = " + myObj.name;
s += ", ";
s += "address = " + myObj.address;
sResult = s;
```

When you run the script above and have the variable *sResult* in the Watch panel, the values of the properties ”name” and ”address” in myObj will be displayed. The text in sResult will be:

name = Kalle Kula, address = Kulgatan 9

This is because we first entered a function in the property ”addProperty”, after which we used the function to add another two properties, ”name” and ”address”.

Formal parameters

A formal parameter is a parameter that was named when the function was created. In the following example, a, b and c are formal parameters.

```
function sum(a, b, c)
{
   return a+b+c;
}
```

Argument

When the function is called, it has actual parameters or arguments.

In the example below, 1 and 2 are arguments. Please note that the number of arguments can be higher or lower than the number of formal parameters.

```
function sum(a, b, c) { return a+b+c; };
var mySum = sum(1, 2);
```

In the script above, mySum will contain the value *NaN*. Each argument that is omitted is assumed to have the value *undefined*, and 1+2+undefined becomes *NaN*.

3.1.10 Prototype

What happens when an object lacks the property you are looking for?

Each object has a prototype. This can be another object or null. If it is an object, the script engine continues to search for the property in the prototype. If the prototype has a prototype, in turn, the script engine can continue to search there, etc. We say that the object has a prototype chain.

Objects of type Object have the prototype Object.prototype. This has a function property called ”hasOwnProperty”. The function can list whether an object has a property with a specific name.

```
var myObj = new Object();
myObj.someProperty = 7;
if(myObj.hasOwnProperty("someProperty"))
{
   // We get here as the object has such a property
   sResult = "myObj has a property someProperty";
}
if(myObj.hasOwnProperty("hasOwnProperty"))
{
   // We do not get here as the property
   // hasOwnProperty
   // is actually in the prototype
}
else
{
   sResult = "myObj has no property hasOwnProperty";
}
```

How can we use the property ”hasOwnProperty” in myObj if myObj does not have a property called that?

When you write myObj.hasOwnProperty

the script engine starts to search in myObj but finds no such property there. It continues to the prototype and finds a property there with this name.

We usually enter functions in the prototype as they do not need to exist as properties in every object instance.

We usually also enter the default values of properties in the prototype so that, if the object itself has not defined any value, the default value in the prototype is used.

When you drag the property hasOwnProperty from the list view to the text editor, you get the text:

3.1.11 Constructor

A constructor is simply a function that is used to create a new object. Normally, you have to write ***new*** before the function call to make the constructor do its job correctly. According to convention, constructors always start their names with an upper case letter.

Example:

```
var myObject = new Object();
```

When a constructor is called in a new expression, it retrieves the value of its ***prototype*** property and sets it as the prototype for its new object.

So you can create your own constructors.

```
function Person(name, address)
{
   this.name = name;
   this.address = address;
}

Person.prototype.toString = function()
{
   return this.name + ", " + this.address;
}

var myPerson = new Person("Kalle Kula", "Kulgatan 9");
sResult = myPerson;
```

The example above defines a Person constructor that can be used to create new objects. These objects can be converted to strings because we enter the function ***toString*** in ***Person.prototype***. Finally, we create a new person and assign the person to the string sResult. The assignment to sResult converts the object to a string to calculate the text to be assigned to sResult. The result is:

```
Kalle Kula, Kulgatan 9
```

If you do not create a ***toString*** function in ***Person.prototype***, the one from ***Object.prototype*** will be used instead. The result would then have been:

[object Object].

3.1.12 The Global Object

When you run scripts in Ethiris, a number of objects are available for use. The most important of these is the global object. Another central object is the data store's script representation, the DataStore object.

A global object is always available. A global object has the properties specified by the specification for the ECMAScript program language.

A global object also has other properties:

- DataStore is the script representation of the data store. This object allows you to access variables in the data store from inside scripts.
- COMObject is a constructor that creates new COM objects. The constructor can take a GUID or a ProgID. When you have created the object, you can call methods, retrieve properties, and receive events from the COM object.
- Timer is a constructor that creates new objects for measuring and checking whether a specific time has passed.
- File is a constructor that creates new objects for reading from and writing to text files.
- Process is a constructor that can be used to start and terminate external programs from scripts.
- Clients is a helper object with a number of methods for getting references to client objects. Clients is only available from license level Advanced and upwards.
- Transaction is an object that is returned when you send a command to one or several clients.
- Cameras is an object that contains a list of all cameras that are defined in the server.
- RemoteClients is an object that contains a list of all clients that are defined in the Remote Clients panel in Ethiris Admin.
- AccessControllers is an object that contains a list of all defined Access controllers in the server configuration.
- Date is an object that can be used for calculations on date and time.

3.1.13 Scope Chain

The Scope chain looks slightly different in Ethiris to how it may look in other programs that use ECMAScript.

Let us start by looking at variable and function declarations. This is an example of such declarations:

```
var temp;
var myString = "hej";
function add(a, b) { return a+b; }
```

To explain the scope chain concept, there is a description below of how the script engine manages to reach properties in different objects.

The names you write in scripts are called identifiers. Take the following expression as an example:

```
offset + factor*value;
```

In the example above, ***offset***, ***factor,*** and ***value*** are identifiers.

When you use an identifier, the script engine tries to find a property with that name among the objects in the scope chain. The scope chain is simply a list of objects. The scope chain normally contains only the global object. So that you can easily access variables in the data store, the scope chain also contains the DataStore object.

Using this, you can write, for example:

```
variable1 + variable2;
```

instead of

```
DataStore.variable1 + DataStore.variable2;
```

The script engine starts to search for a property called **”variable1”** in the global object. When it finds no such property there, it continues to the next object in the scope chain, the DataStore object. Such a property is found there and is used in the expression. We say that the identifier could be resolved.

If an identifier cannot be resolved, an exception is thrown. The following script will return **”Reference error”** as there is no object with the property **”vriable1”** in the scope chain.

```
try
{
   return vriable1 + variable2; // NB. Typo!
}
catch(e)
{
   return e.message;
}
```

3.1.14 Variables in the Data Store

The data store’s variables are of the types Integer, Double, String, and Boolean and are converted to ECMAScript values when you retrieve the properties in the script object DataStore.

The DataStore types Integer and Double are converted to values of type Number without losing precision.

The DataStore types String and Boolean are converted without loss to the ECMAScript types String and Boolean.

DataStore type	ECMAScript type
Boolean	Boolean
Integer	Number
Double	Number
String	String

ECMAScript values of type Number are rounded to the nearest integer in connection with assignment to a DataStore variable of type Integer. You lose no precision in connection with the assignment to a Double.

KENTIMA Automation and Security Products

For allocation of different data types, the following conversion rules are used:

ECMAScript type	**DataStore type**	
Boolean	Boolean	
Number	Boolean	NaN and 0 are mapped on false. All other values are mapped on true
String	Boolean	Empty strings are mapped on false. All other strings are mapped on true
Boolean	Integer	true and false are mapped on 1 and 0 respectively
Number	Integer	The value is rounded to the nearest integer
String	Integer	The string’s contents are parsed as a floating-point, and the result is rounded to an integer. If it fails, the value is 0
Boolean	Double	true and false are mapped on 1.0 and 0.0 respectively
Number	Double	
String	Double	The string’s contents are parsed as a floating-point. If it fails, the value is 0
Boolean	String	true and false are mapped on ”true” and ”false” respectively
Number	String	The number is formatted to a string
String	String	

3.1.15 Bitwise Expressions

The bitwise operators AND, OR, XOR, and NOT are available in ECMAScript. There are also bitwise shift operators: Left Shift, Signed Right Shift, and Unsigned Right Shift.

An integer variable is often used to transfer up to 32 Boolean values between systems. Therefore, it may be good to know how to perform bitwise operations on such values.

There may, for example, be 32 different status values indicating whether different cameras are active or not, which you have packed in an integer instead of having 32 Boolean variables in the OPC server.

Some of the operators in this section can be used as allocation operators. These are:

```
<<=  >>=  >>>=  &=  ^=  |=
```

The expression

`a @= b` (where **@** represents any of the operators **<<**, **>>**, **>>>**, **&**, **^** or **|**)

is equivalent to

`a = a @ b.`

Bitwise Left Shift (<<)

Bitwise Left Shift moves the bit pattern in the binary representation of the integer one step to the left and adds zeroes. Example:

00000000000000000000000000000101 = 5
00000000000000000000000000001010 = 10

The number's value is then multiplied by two.

Creating a *mask* is an important area of application for this operator when it comes to working with individual bits in an integer. The script

```
var mask = 1 << k;
```

creates an integer value that has a one in position k and zeroes elsewhere. The value of the number is thus 2^k.

Bitwise Signed Right Shift (>>)

Bitwise Signed Right Shift moves the bit pattern in the binary representation of the integer one step to the right and adds the value of the bit that represents the sign of the integer. Example:

00000000000000000000000000001010 = 10
00000000000000000000000000000101 = 5

11111111111111111111111111110110 = -10
11111111111111111111111111111011 = -5

Bitwise Unsigned Right Shift (>>>)

Bitwise Unsigned Right Shift moves the bit pattern in the binary representation of the integer one step to the right and adds zeroes. Example:

11111111111111111111111111110110 = -10
01111111111111111111111111111011 = 2147483643

Bitwise AND (&)

Bitwise AND returns a number, the bits of which are set where the corresponding bits are set in both operands.

Bitwise AND can be used to filter out a specific bit from an integer value.

```
var mask = 1 << k;
var temp = value & mask;
var bit = temp? 0 : 1;
```

The operation ***1 << k*** produces a binary number that has a one in position k and zeros elsewhere. This number has the value 2^k.

The operation ***value & mask*** filters out the bit in position k from the value.

The value of temp is 0 if the bit in position i is set, 2^k otherwise. In the last step, we convert this to the integer 0 or 1, depending on whether the bit is set or not.

You can use the expression ***value & mask*** directly in *if* statements and logical expressions as the numerical value ***0*** is converted to ***false*** and all others to ***true***.

```
// If the bit in position k is set in value
if(value & (1 << k))
{
   ...
}


// If the bits in positions 3 and 5 are set in value
if( (value & (1 << 3)) && (value & (i << 5)) )
{
   ...
}
```

Bitwise OR (|)

Bitwise OR returns a number, the bits of which are set where the corresponding bits are set in any of the operands.

Bitwise OR can be used to set a specific bit in an integer value.

```
// Sets bit number k in the variable value
var mask = 1 << k;
value = value | mask;
```

Bitwise XOR (^)

Bitwise XOR returns a number, the bits of which are set where the corresponding bits are set in just one (but not both) of the operands.

Bitwise XOR can be used to change a specific bit in an integer value.

```
// Changes bit number k in the variable value to
// 0 if it is 1
// 1 if it is 0
var mask = 1 << k;
value = value ^ mask;
```

Bitwise NOT (~)

Bitwise NOT exchanges all zeroes for ones in the binary representation of a value. Example:

00000000000000000000000000001010
11111111111111111111111111110101

Bitwise NOT can be used together with bitwise AND to reset a specific bit in an integer value.

```
// Sets bit number k in the variable value
var mask = 1 << k;
value = value & ~mask;
```

3.2 Differences to the ECMAScript Standard

To make it easy to use the scripting language in Ethiris, some additions have been made. There are also some restrictions, above all to prevent the risk of scripts ending up in endless loops.

The ECMAScript standard has the designation ECMA-262 and is available at http://www.ecma-international.org/.

3.2.1 Additions to the Scripting language

return

The statement return can be used anywhere in a script, not just inside functions.

Safe for loop

With the standard for loop in ECMAScript, you can write loops that do not stop. While loops and do-while loops also have this problem.

Therefore, it can be difficult to guarantee that your scripts stop when you have written complicated loops. So a variant of ECMAScript's for-in loop has been added to Ethiris.

Go through your scripts and see whether you can replace your for, while and do-while loops with the type of for-in loop described in this section.

The addition to the language consists of a new type of object, the Sequence object, which is created with the Sequence constructor in the global object.

Example 1:

```
for (i in Sequence(5, 12))
{
        arr[i] = 0;
}
```

This loop sets the elements with index 5 to 12 in the array arr to 0. The variable *i* will thus have the value 5 the first time the loop is run, 6 the next time, etc. If we see Sequence(5, 12) as an object that has the properties 5, 6, 7, 8, 9, 10, 11 and 12, this loop works almost exactly like the built-in for-in loop in ECMAScript. The difference is that *i* becomes an integer instead of a string. Sequence(1, 1000000) will also not take up any more memory than Sequence(1,2). If you write Sequence(12,5), the control variable will get the values in reverse order instead.

Example 2:

```
var result = 0;
for (i in Sequence(1, 10))
{
        result = result + i;
}
```

As *i* is always an integer, the *result* will have the value (1+2+3+4+5+6+7+8+9+10):

55

not the value

”12345678910”

which you would have got if *i* had been a value of type String. The plus operator (+) performs the string concatenation in ECMAScript if either link is of type string.

Events

An Event object and an EventTarget object have also been added to the script engine. They can be used to send an event, for example, an event from a COM object.

These objects generally follow the ”Document Object Model (DOM) level 3 events” specification from the standardization organization W3C, the World Wide Web Consortium. The specification is available at http://www.w3.org/DOM/DOMTR.

synchronized

A new keyword has been introduced into the scripting language, ”synchronized”. This keyword can be used to create synchronized statements in which you synchronize access to a specific object so that only one execution thread can use the object at a time.

The objects that are accessible from several different threads in Ethiris are DataStore and the variables in DataStore. These are used by both the script thread and the thread that manages OPC communication.

Example 1:

In the following example, we can ensure that the communication variables var1-var4 have consistent values at the time at which they are written to the OPC server as the script below cannot be run while the OPC thread is writing.

```
synchronized(DataStore)
{
   DataStore.var1 = DataStore.var2 || DataStore.var3;
   DataStore.var4 = true;
}
```

If you omitted the synchronized block, it might, for example, happen that the values of var1-var3 are sent to the OPC server after the value of var4, although the statements in the script appear to show that var1-var3 are changed before var4.

Example 2:

Instead of synchronizing the entire data store, you can also synchronize a specific variable. As shown below, we can ensure that the variable var4 has the same value each time it is evaluated.

```
synchronized(var4)
{
   var1 = var4 + var4*var4;
}
```

If you omitted the synchronized block, it might, for example, happen that the OPC server notifies Ethiris of a data change for var4 while the expression is being calculated, so that, if the value is changed from 3 to 4 before we have performed the addition, you would get the value (3 + 4*4) instead of the expected (3 + 3*3).

Another way of avoiding this problem is to copy var4 to a temporary variable before the calculation is performed:

```
var tmp = var4;
var1 = tmp + tmp*tmp;
```

3.2.2 Restrictions in the Scripting Language

Labels

The lines in a script cannot have labels. For example, you cannot write

```
label: myNumber = 7;
```

If you omit the label, however, the above is valid code.

Regular expressions

The scripting language does not implement support for regular expressions. You cannot enter regular expressions such as

```
var myRegExp = /a[a-z]{2,4}/;
```

Properties in the global object

The following properties do not exist in the global object.

Functions:

- eval()
- parseInt()
- parseFloat()
- decodeURI()
- decodeURIComponent()
- encodeURI()
- encodeURIComponent()
- String.prototype.localeCompare()
- String.prototype.match()
- String.prototype.search()
- String.prototype.split()
- String.prototype.toLowerCase()
- String.prototype.toUpperCase()
- String.prototype.toLocaleLowerCase()
- String.prototype.toLocaleUpperCase()
- Array.prototype.toLocaleString()
- Array.prototype.concat()
- Array.prototype.reverse()
- Array.prototype.shift()
- Array.prototype.slice()
- Array.prototype.sort()
- Array.prototype.splice()
- Array.prototype.unshift()
- Function.prototype.toString()

Constructors:

- RegExp()
- EvalError()
- RangeError()
- ReferenceError()
- SyntaxError()
- TypeError()
- URIError()

As the script engine does not contain regular expressions, String.prototype.replace cannot take a regular expression as its first argument.

3.3 The Global Object

The global object is always available when you write scripts and expressions. You can retrieve properties from the global object by writing the property's name as the global object is always in the scope chain.

3.3.1 Properties in the Global Object

alert(message)

Displays a message in the debug window. This function is basically the same as Debug.Print().

app

A reference to the global object itself.

AccessControllers

The AccessControllers object.

Array

The array constructor.

Boolean

The Boolean constructor.

Cameras

The Cameras object.

Clients

The Clients object.

COMObject

The COMObject constructor.

DataStore

The DataStore object.

Date

The Date constructor.

Debug

The Debug-object.

Error

The error constructor.

Function

The function constructor.

File

The file constructor.

Infinity

Positive infinity.

isFinite(value)

Decides whether the **value** is a finite number.

isNaN(value)

Decides whether the **value** is any of the ”Not-a-Number” values in accordance with the IEEE standard.

Math

The math object. This object contains properties that can be used in mathematical calculations.

NaN

A number that represents the ”Not-a-Number” values in accordance with the IEEE standard.

Number

The number constructor.

Object

The object constructor.

Process

The process constructor.

ProcessRunning

The ProcessRunning-object.

RemoteClients

The RemoteClients object.

Sequence

The sequence constructor.

String

The string constructor.

Timer

The timer constructor.

Transaction

The Transaction object.

undefined

The value is undefined.

3.4 DataStore Object

The data store is represented by the script object ***DataStore***, which is always available in scripts in Ethiris.

3.4.1 Properties in DataStore Objects

The data store has a property for each variable in your data store. The property has the same name and data type as the variable in the data store.

3.5 Variables in DataStore

Variables in the DataStore are directly available in the script code.

3.5.1 Events in DataStore variables

The user-defined variables in the DataStore have the following events:

onDataChange(event)

Each variable in the data store that is defined by the user can generate an event each time the value of that variable changes.

The event can be tied to a function that will get called each time the variable gets a new value. If we have defined a variable named V1 in the datastore we could use the following script to detect changes of its' value.

```
V1.onDataChange = function(event)
{
    Debug.Print("new value: " + event.Data);
}
```

This is an alternative to detecting changes in a value manually in the script by means of another variable.

Note however that the script inside the function will be not be executed sequentially with the surrounding code but instead at a point in time between the execution cycles of the main script. This can be important if the function modifies anything that is used in the surrounding script.

The event has one argument ***event*** that besides the standard Java script properties has the following properties:

Data – the new value for the variable.

target – a reference to the variable.

3.6 Array Object

The array object has a property for each element in the array. They are called ”0”, ”1”, ”2”, etc. You can use brace notation to retrieve these properties:

```
someArray[3]
```

When you iterate over an array with a for-in loop, it is guaranteed that the loop will go through the array members in numerical order.

```
var str = "";
for(i in someArray)
{
        str += "someArray["+i+"]= ";
        str += someArray[i];
        str += "\n";
}
```

Let us say that the array has 3 members with the values ”a”, ”b” and ”c”. The variable str will then contain the following string after you have run the script:

```
someArray[0] = a
someArray[1] = b
someArray[2] = c
```

3.6.1 Properties in Array Objects

The array object has the following properties:

length

Indicates the length of the array object. You can change the length of the array by setting this property. However, you cannot make the length be less than zero.

3.6.2 The Array Constructor

new Array(length)

Creates a new array object with the length specified. If ***length*** is omitted, an array object with length 0 is created.

new Array(item1, item2, item3, …)

Creates a new array object that contains the elements ***item1***, ***item2***, ***item3***, …

3.6.3 The Array Prototype

The following functions are available for use in an array object.

Array.prototype.join(separator)

Concatenates the elements in this array object, separated by the string specified. If a **separator** is omitted, **”,”** is used as the separator. The resulting string is returned.

Array.prototype.pop()

The array object's last element is removed from the array and returned.

Array.prototype.push(item1, item2, item3, …)

The specified elements ***item1***, ***item2***, ***item3***, … are added to the array object in the correct order. The new length of the array object is returned.

Array.prototype.toString()

Converts this array object to a string exactly as if ***join()*** had been called without an argument. The string is returned.

3.7 Process Object

Process objects have a few functions that you can use to obtain specific information about processes and manipulate processes that you started from scripts.

3.7.1 Properties in Process Objects

Process objects have no special properties.

3.7.2 The Process Constructor

new Process(fileName[, currentDirectory])

Creates a new process object and starts a process with the file specified.

currentDirectory is optional. If supplied the process created gets this directory as startup directory.

3.7.3 The Process Prototype

Process.prototype.isAlive()

Finds out whether the process that the process object is monitoring is still being run.

Process.prototype.terminate()

Terminates the process that the process object is monitoring.

Example:

The following script starts some kind of background application and terminates it after 5 minutes if it is still running.

Remember that Ethiris Server has no access to a user interface (*Desktop*) and hence should not start an application with a user interface since it can not be displayed.

```
if(myProc == undefined)
   myProc = new Process("BackgroundWork.exe");

if(myTimer == undefined)
   myTimer = new Timer(5*60*1000);

if(myProc.isAlive() && myTimer.timeout())
   myProc.terminate();
```

3.8 ProcessRunning object

ProcessRunning is a helper function used to find out if a process with a specific name is currently running in the same computer as Ethiris Server. In the case of a cluster, it is the member that currently is master that will run the script and hence it is in that computer the function will be run.

The function can be used like this to figure out if Notepad is running in the server computer:

```
Debug.Print("Notepad is running: " +
ProcessRunning("notepad.exe"));
```

ProcessRunning is preferably used together with the Process object.

3.9 Object Object

The object *object* is the simplest type of object in the script engine.

3.9.1 Properties in Object Objects

Object objects have no built-in properties. You select the properties that are to exist by allocating to them.

3.9.2 The Object Constructor

new Object()

Creates a new object *object*.

new Object(value)

Creates a new object of a type that depends on ***value***. If the value is a number, a number object is created. If the value is a string, a string object is created. If the value is a Boolean value, a Boolean object is created.

Object()

When the object constructor is called as a function, a new object *object* is created.

Object(value)

When the object constructor is called as a function, it converts the ***value*** to an object and returns the result.

3.9.3 The Object Prototype

The following functions are available for use in an object *object*.

Object.prototype.hasOwnProperty(property)

Decides whether this object has the property ***property***. It does not check whether the property exists in any object in the prototype chain.

Object.prototype.isPrototypeOf(object)

Decides whether this object exists in the prototype chain for ***object***.

Object.prototype.propertyIsEnumerable(property)

Decides whether the property ***property*** in this object is enumerable (i.e. does not have the attribute *DontEnum*). It does not check the object in the prototype chain.

Object.prototype.toString()

Converts this object to a string and returns the string. The object is converted to a string in the following steps:

1. Let ***class*** be this object's class.
2. Return the string value **”[object ” + class + ”]”.**

Please note that only the script engine has access to an object's class. For object objects, the class is the string **”Object”**.

Object.prototype.valueOf()

Returns this object.

3.10 Function Object

Function objects represent functions that you create in the script engine and built-in functions.

Function objects can often be used as constructors in ***new*** expressions.

3.10.1 Properties in Function Objects

Function objects have the following properties:

length

Indicates the typical number of arguments the function expects. If the function is called with fewer arguments, those that are missing will normally be assumed to have the value ***undefined***. This property has the attributes DontDelete, ReadOnly, and DontEnum.

prototype

An object *object.* When the function is used in ***new*** expressions, a new object is created, the prototype of which is the value of the ***prototype*** property. This property has the attribute DontDelete.

3.10.2 The Function Constructor

You normally create functions by using the keyword ***function***.

Example:

```
// Function declaration
function factorial(n)
{
   if(n <= 0) return 1;
   return n*factorial(n-1);
}

// Variable is allocated value of function expression
var factorial = function factorial(n)
{
   if(n <= 0) return 1;
   return n*factorial(n-1);
}

// Anonymous function
var factorial = function(n)
{
   if(n <= 0) return 1;
   return n*arguments.callee(n-1);
}
```

You can also call the function constructor.

new Function(p1, p2, …, pn, body)

Creates a new function object that has formal parameters with names ***p1***, ***p2***, … ***pn***. The function body is set to ***body***. The formal parameters can be omitted or combined to form one or more strings.

Example:

The following call creates a function that adds up three numbers and calls it to calculate the sum of 1, 2 and 3.

```
var sum = new Function("a", "b", "c", "return a+b+c");
return sum(1, 2, 3);
```

Example:

The following expressions produce the same result:

```
new Function("a", "b", "c", "return a+b+c")
new Function("a, b, c", "return a+b+c")
new Function("a,b", "c", "return a+b+c")
```

Function(p1, p2, …, pn, body)

When the function constructor is called as a function, a new function object is created just as if you had used ***new***.

3.10.3 The Function Prototype

The following functions are available to call in function objects.

They are used, for example, when you want to call a function that exists as a property in an object with a this-object other than the object.

Function.prototype.apply(thisArg, argArray)

Performs a function call with this function object.

If ***thisArg*** is ***null*** or ***undefined***, the global object is used as ***this***-object for the function call. Otherwise, ***thisArg*** is used as ***this***-object.

If ***argArray*** is ***null*** or ***undefined***, no arguments are sent to the function call. If ***argArray*** is neither an array object or an arguments object, a TypeError exception is thrown. Otherwise, the arguments ***argArray[0]***, ***argArray[1]***, … ***argArray[argArray.length-1]*** are sent to the function.

Function.prototype.call(thisArg, arg1, arg2, …)

Performs a function call with this function object.

If ***thisArg*** is ***null*** or ***undefined***, the global object is used as ***this***-object for the function call. Otherwise, ***thisArg*** is used as ***this***-object.

The voluntary parameters ***arg1***, ***arg2***, etc. are used as arguments for the function call.

3.11 String Object

String objects are created when you convert a string to an object.

Strings are converted automatically to string objects when you use dot notation. Example:

```
"abcdefghijklmnopqrstuvwxyz".length
```

converts the string specified to a string object and retrieves the object’s ***length*** property.

3.11.1 Properties in String Objects

length

The number of characters in this string.

3.11.2 The String Constructor

new String()

Creates a new string object that contains an empty string.

new String(value)

Converts ***value*** to a string and creates a new string object that contains the string.

String(value)

Converts ***value*** to a string value. If value is omitted, an empty string is returned, `""`.

String.fromCharCode(char0, char1, …)

Returns a string that contains as many characters as the number of arguments. Each argument specifies one character in the string, from left to right.

3.11.3 The String Prototype

The following functions are available to call in string objects.

String.prototype.charAt(pos)

Returns a string that contains the character in position ***pos*** in this string.

String.prototype.charCodeAt(pos)

Returns a non-negative number less than 65536 that represents the character code in the UNICODE character table for the character in position ***pos*** in this string.

String.prototype.concat(string1, string2, …)

Returns a string that contains the characters in this string followed by the characters in ***string1***, ***string2***, etc.

String.prototype.indexOf(searchString, pos)

Searches through this string for ***searchString***, starting in position ***pos***, and returns a number that indicates the position in which it found ***searchString***. If the search fails, –1 is returned. If ***pos*** is omitted, the search starts in position 0.

String.prototype.lastIndexOf(searchString, pos)

Searches through this string from the back, starting in position ***pos***, and returns the position in which it finds ***searchString***. If the search fails, –1 is returned. If ***pos*** is omitted, the search starts at the end of the string.

String.prototype.replace(searchValue, replaceValue)

Let string be the result of converting the ***this***-object to a string. Let searchString be ***searchValue*** converted to a string. Let newstring be ***replaceValue*** converted to a string.

Searches through this string for the first incidence of searchString. Returns a string in which the substring found is replaced with a string that is obtained from newstring by replacing characters in newstring with replacement text as shown in the table below.

Character	Replacement text
$$	$
$&	The substring found
$`	The part of the string preceding the substring found
$´	The part of the string after the substring found

Table of characters that may occur in the argument replaceValue, and their respective replacement texts.

Example:

```
"Hello, world!".replace("Hello","See you later")
```

produces the result **”See you later, world!”**.

To be able to replace with **”$”**, you have to write **”$$”** as the character **”$”** has special significance in ***replaceValue***:

```
"£5, £13, £20".replace("£","$$")
```

produces the result **”$5, £13, £20”**, as only the first incidence is replaced.

```
"Jan-Åke Jansson".replace("Jans", "$&$´ is Jakob
$&$´´s $´")
```

produces the result **”Jan-Åke Jansson is Jakob Jansson´s sonson”.**

String.prototype.slice(start, end)

Returns the substring of this string that starts in position ***start*** and runs up to but does not include ***end***. If ***end*** is omitted, the substring that starts in position ***start*** and goes up to and includes the end of the string is returned.

If ***start*** is negative, ***length+start*** is used, where ***length*** is the length of this string. If ***end*** is negative, ***length+end*** is used.

String.prototype.substring(start, end)

Returns the substring of this string that starts in position ***start*** and runs up to but does not include ***end***. If ***end*** is omitted, the substring that starts in position ***start*** and goes up to and includes the end of the string is returned.

If any of the arguments is ***NaN*** or negative, it is replaced with 0. If any of the arguments is larger than the length of the string, it is replaced with the length of the string. If ***start*** is larger than ***end***, they are switched.

String.prototype.toString()

Returns this string object’s string value, i.e. the same as valueOf.

String.prototype.valueOf()

Returns this string object’s string value, i.e. the same as toString.

3.12 Boolean Object

Boolean objects are created when you convert Boolean values to objects.

Please note that Boolean objects are always converted to *true* if you use them in ***if*** statements and other constructions that convert to Boolean values.

To find out the Boolean value that the Boolean object encapsulates, you have to use Boolean.prototype.valueOf().

Example:

```
var myBoolean = new Boolean(false);
if(myBoolean)
{
   sResult = "We always get here";
}
if(myBoolean.valueOf())
{
   sResult = "We never get here";
}
```

3.12.1 Properties in Boolean Objects

Boolean objects have no special properties.

3.12.2 The Boolean Constructor

Boolean objects are created when you convert Boolean values to objects. You can also create new Boolean objects using the Boolean constructor.

new Boolean(value)

Converts *value* to a Boolean value and returns a new Boolean object that contains the value.

Boolean(value)

Converts ***value*** to a Boolean value.

3.12.3 The Boolean Prototype

The following functions are available for use in a Boolean object.

Boolean.prototype.toString()

If this Boolean object's Boolean value is *true*, the string `"true"` is returned. Otherwise, `"false"` is returned.

Boolean.prototype.valueOf()

Returns this Boolean object's Boolean value.

3.13 Number Object

Number objects are created when you convert numbers to objects.

3.13.1 Properties in Number Objects

Number objects have no special properties.

3.13.2 The Number Constructor

Number objects are created when you convert numbers to objects. You can also create new number objects using the number constructor.

new Number(value)

Converts *value* to a number and returns a new number object that contains the value. If *value* is omitted, a number object that contains the value *+0* is returned.

Number(value)

Converts *value* to a number and returns the number.

Number.MAX_VALUE

A constant that contains the highest positive finite number that the script engine can represent.

Number.MIN_VALUE

A constant that contains the lowest positive number that the script engine can represent.

Number.NEGATIVE_INFINITY

A constant that has the value of negative infinity. You can also get negative infinity by writing *-Infinity*.

Number.POSITIVE_INFINITY

A constant that has the value of positive infinity. You can also get positive infinity by writing *Infinity*.

Number.NaN

A constant that represents ”Not-a-Number” values in accordance with the IEEE standard. You can also get NaN by writing *NaN* as the global object also contains a property called *NaN* and the value of ”Not-a-Number” values in accordance with the IEEE standard.

3.13.3 The Number Prototype

The following functions are available for use in a number object.

Number.prototype.toExponential(fractionDigits)

Returns a string that contains the number represented in exponential notation with a digit before the decimal point and *fractionDigits* after the decimal point. If *fractionDigits* is omitted, as many value digits are used as is necessary to specify this number uniquely.

Number.prototype.toFixed(fractionDigits)

Returns a string that contains the number represented in fixpoint notation with *fractionDigits* after the decimal point. If *fractionDigits* is undefined, the value 0 is used.

Number.prototype.toPrecision(precision)

Returns a string that contains the number represented either in exponential notation with a digit before the decimal point and *precision-1* digits after the decimal point or in fixpoint notation with *precision* value digits.
If *precision* is *undefined*, this number is converted to a string with *toString()*.

Number.prototype.toString(radix)

If *radix* is the number 10 or *undefined*, this number is converted to a string.

If *radix* is an integer from 2 to 36, but not 10, the result is a string that contains this number represented in *radix* notation. Example:

```
255.0.toString(16)
```

becomes ”FF”.

Radix 16 is often also called hexadecimal notation. Radix 2 is often called binary notation and radix 8 octal notation.

Number.prototype.valueOf()

Returns this number object’s number value.

3.14 The Math Object

The math object contains properties that are used primarily in mathematical calculations. You get the math object from the property *Math* in the global object.

3.14.1 Properties in the Math Object

The math object contains both values and functions.

E

The value of ***e***, the radix of the natural logarithm function.

LN10

The natural logarithm of 10.

LN2

The natural logarithm of 2.

LOG10E

The radix 10 logarithm of ***e***.

LOG2E

The radix 2 logarithm of ***e***.

PI

The value of ***π***, the ratio between the circumference of a circle and its diameter.

SQRT1_2

The square root of ½.

SQRT2

The square root of 2.

abs(value)

Calculates the absolute amount of ***value***.

acos(value)

Calculates the anticosine of ***value***.

asin(value)

Calculates the antisine of ***value***.

atan(value)

Calculates the antitangent of ***value***.

atan2(y, x)

A variant of atan that takes two arguments. Thanks to this, the function's range of values is the interval [0, 2 π].

ceil(value)

The ceiling function. Rounds ***value*** up.

cos(radians)

Calculates the cosine of ***radians***, which must be an angle stated in radians.

exp(value)

The exponential function. Calculates ***e*** raised to the power ***value***.

floor(value)

The floor function. Rounds ***value*** down.

log(value)

The natural logarithm function. Calculates the radix-*e* logarithm of ***value***.

max(value1, value2, …)

Returns the highest value of ***value1, value2***, etc. The function can take one or more parameters.

min(value1, value2, …)

Returns the lowest value of ***value1, value2***, etc. The function can take one or more parameters.

pow(base, exponent)

The power function. Calculates ***base*** raised to the ***exponent***.

random()

Returns a rectangularly distributed pseudo-random number in the interval [0, 1].

round(value)

Rounds *value* to the nearest integer

sin(radians)

Calculates the sine of ***radians***, which must be an angle stated in radians.

sqrt(value)

Calculates the square root of ***value***.

tan(radians)

Calculates the tangent of ***radians***, which must be an angle stated in radians.

3.15 Error Object

Error objects are thrown as exceptions when an error occurs in a script. You can also create error objects by using the error constructor and use them in your own *throw* expressions.

3.15.1 Properties in Error Objects

message

This property contains an error message that indicates why an exception was thrown.

Example:

```
try
{
   var myVariable = 7;
   sResult = myVarible + 3;
}
catch(e)
{
   sResult = e.message;
}
```

The script above will assign sResult the text:

”Reference error: GetValue failed for property name ”myVarible” because base is null on line 4”.

The reason for this error message is that you spelled ”**myVariable**” incorrectly on line 4.

name

Indicates the type of error, for example **”Error”**, **”RangeError”**, etc.

3.15.2 The Error Constructor

You can create your own error objects using the error constructor.

new Error(message)

Creates a new Error object with the message ***message***.

Error(message)

When you call the error constructor as a function, a new error object is created just as if you had called it with ***new***.

3.15.3 The Error Prototype

Error.prototype.toString()

Returns this error object’s error message.

3.16 COMObject Object

COMObject objects represent COM components in Microsoft® Windows.

3.16.1 Properties in COMObject Objects

The properties of a COMObject object depend on the COM component it represents. See the documentation for the COM component in question.

These properties can be both values and functions.

Example with COM objects that use the component ”Ethiris Client Remote Client” to select the view ”Building” in the Ethiris client in the event of motion in front of camera Utb2:

```
// Event function called when a view is changed in the
client
function ViewChanged(event)
{
   sResult = "View changed to: " + param1;
}

var comObj;
if(comObj == undefined)
{
   // Create an instance of the COM object
   comObj = new
COMObject("EthirisClientRemoteClient.Control");

   // Connect locally on port 1237
   comObj.Connect("127.0.0.1", 1237);


   // Add event listener for ViewChange
   comObj.addEventListener("ViewChange",
"ViewChanged", false);
}


// If motion in camera Utb2
if(Utb2.motion2.Motion)
{
   // Select the view "BuildingA" in the client
   comObj.SelectClientView("BuildingA");
}
```

3.16.2 The COMObject Constructor

new COMObject(ProgID)

Creates a new COMObject object that exposes a COM component to the script engine. The parameter *ProgID* is a program ID stated as a string.

COMObject(id)

When the COMObject constructor is called as a function, a new COMObject object is created just as if you had used ***new***.

3.16.3 The COMObject Prototype

The COMObject prototype has no special properties.

3.17 Sequence Object

Sequence objects are used primarily in **for-in** loops.

3.17.1 Properties in Sequence Objects

Sequence objects can only have properties, the names of which are integers in a specific interval that was specified when the sequence object was created.

Example 1:

```
for (i in Sequence(5, 12))
{
        arr[i] = 0;
}
```

This loop sets the elements with index 5 to 12 in the array ***arr*** to 0. The variable *i* will thus have the value 5 the first time the loop is run, 6 the next time, etc. If we see Sequence(5, 12) as an object that has the properties 5, 6, 7, 8, 9, 10, 11 and 12, this loop works almost exactly like the built-in for-in loop in ECMAScript. The difference is that ***i*** becomes an integer instead of a string. Sequence(1, 1000000) will also not take up any more memory than Sequence(1,2). If you write Sequence(12,5), the control variable will get the values in reverse order instead.

Example 2:

```
var result = 0;
for (i in Sequence(1, 10))
{
        result = result + i;
}
```

As *i* is always an integer, the *result* will have the value (1+2+3+4+5+6+7+8+9+10):

55

not the value

”12345678910”

which you would have got if *i* had been a value of type String. The plus operator (+) performs the string concatenation in ECMAScript if either link is of type string.

3.17.2 The Sequence Constructor

You can create new sequence objects using the sequence constructor. The fact that a sequence object goes from the integer ***start*** to the integer ***end*** means that it has properties with the names ***start***, ***start***+1, ***start***+2, … ***end***-1, ***end***.

Example:

```
var mySequence = new Sequence(1, 3);
```

results in a new object with the properties 1, 2 and 3 being created. These properties all initially have the value *undefined.*

You can store values in these properties but not in properties with any names other than 1, 2 and 3.

```
mySequence[1] = "hi";
sResult = mySequence[1];
mySequence[0] = "hello";
sResult += ", " + mySequence[0];
```

The first call for an alert will assign *sResult* the text ”hi”. The second call, however, will generate the text ”undefined” as you cannot create new properties in sequence objects except those specified at the time at which the sequence was constructed.

new Sequence(end)

Creates a new sequence object that goes from 0 to ***end***. If ***end*** is not an integer, it is rounded down to the nearest integer.

new Sequence(start, end)

Creates a new sequence object that goes from start to end. If any of the parameters is not an integer, it is rounded down to the nearest integer.

If ***start*** is larger than ***end***, the properties are enumerated in reverse order:

```
var s = "";
for(i in Sequence(3, 1))
{
   s += i + " ";
}
s += "Takeoff!";
sResult = s;
```

The script above will assign the text ”3 2 1 Takeoff!” to sResult.

Sequence(…)

When the sequence constructor is called as a function, a new sequence object is created just as if you had used ***new***.

3.17.3 The Sequence Prototype

The sequence prototype has no special properties.

3.18 Timer Object

Timer objects have a few functions that you can use to find out how long a script has been running and when a specific time has passed in a script.

Example of a timer with a 1-minute interval:

```
var myTimer;
if(myTimer == undefined)
   myTimer = new Timer(60*1000);

if(mytimer.timeout())
{
   sResult = "60 seconds gone";
   myTimer.start(60*1000);
}
```

3.18.1 Properties in Timer Objects

Timer objects have no special properties.

3.18.2 Events in Timer Objects

The Timer Object has one event you can listen to in your script.

onTimeout

This event is fired when the Timer object has been running for the time defined in its construction.

Example:

```
var myTimer;
if(myTimer == undefined)
   myTimer = new Timer(60*1000);

myTimer.onTimeout = function()
{
   sResult = "60 seconds gone";
}
```

Using the event instead of the timeout() method has the advantage of only being called once without having to use some kind of edge boolean that would make the script more cluttered.

Timers can also be set up to be cyclic automatically. This is more precise than resetting the timer manually since the reset won’t be locked to the execution interval of the script. This is only useful with events because the timer is technically only in a ”timed out” state for a very short time, so the timeout () method will likely miss it and never trigger.

Example:

```
//The Boolean in the timer's constructor indicates
//that it is cyclic
var myTimer;
if(myTimer == undefined)
        myTimer = new Timer(60*1000, true);

myTimer.onTimeout = function()
{
   sResult = "60 seconds gone";
}
```

This example is functionally identical to the first timer example shown, but using events instead of the timeout() method.

3.18.3 The Timer Constructor

new Timer()

Creates a new empty timer object.

new Timer(milliSec)

Creates a new timer object with the time specified.

new Timer(milliSec, cyclic)

Creates a new timer object with the time specified and its cyclic nature defined. A timer is by default not cyclic.

new Timer(milliSec, cyclic, stopped)

Creates a new timer object with the time specified and its cyclic nature defined and whether it should be created in a stopped state. A timer is by default not cyclic and not stopped.

3.18.4 The Timer Prototype

Timer.prototype.checkInterval()

Restarts the timer if the set time has elapsed and returns true if this is the case.

Timer.prototype.checkInterval(milliSec)

Restarts the timer if the specified time has elapsed and returns true if this is the case.

Timer.prototype.elapsed()

Returns the number of milliseconds that have elapsed since the timer was started and up until it was stopped if it is stopped.

Timer.prototype.isRunning()

Returns true if the timer is running and false if it is stopped.

Timer.prototype.start()

Starts/restarts the timer with the set time.

Timer.prototype.start(milliSec)

Starts/restarts the timer with the specified time.

Timer.prototype.stop()

Stops the timer if it was not already stopped. Returns the number of milliseconds elapsed since the timer was started.

Timer.prototype.timeout()

Checks whether the set time has passed.

Timer.prototype.timeout(milliSec)

Checks whether the specified time has passed.

3.19 Date Object

Date objects store points in time and can be helpful when you need to perform calculations on date and time values.

Internally, the Date object stores time indications as a number of milliseconds since midnight beginning January 1st, 1970. To get useful information about for example with which year a certain number of milliseconds corresponds, there are a number of operations you can execute on Data objects.

Midnight beginning January 1st, 1970 is known as the epoch, or sometimes the Unix epoch since Unix operating systems use this point in time as the starting point of a certain time scale. In chronology, an epoch is a point in time that has been chosen as the starting point for a certain time scale. Events that happened at an earlier time can be indicated by using negative numbers. The datastore in Ethiris contains no data type which stores time indications directly. The best way to store a time indication in the data store is to use a Double variable which contains the number of milliseconds since the epoch. If you wish to execute an operation on the time indication, you can create a Date object using the Double variable, make the required changes, and save the Date object back to the variable.

Example:

```
var date = new Date(StartTime);
date.setYear(2010);
StartTime = date;
```

In the example, we assume the variable *StartTime* holds point in time and we adjust the year to be 2010 without affecting moth, day or time parts.

3.19.1 Properties in Date Objects

Date objects have no properties.

3.19.2 The Date Constructor

new Date()

To create a new Date object, write

```
var myDate = new Date();
```

It is suitable to save the new Date object somewhere, in this case in the variable myDate. The new object will contain the date of today and the present time.

When you use the Date constructor to create a new Date object and don’t send any parameters to the constructor, a Date object is created containing the date of today and the present time.

new Date(value)

To create a new Date object which represents a certain number of milliseconds after the epoch, you can call the constructor using a parameter which isn't a string.

```
var anotherDate = new Date(1234151251211);
```

If the parameter is a string, the time will be parsed from it. The string must be in one of three formats.

- ISO-standard – ”YYYY-MM-DD HH:mm:ss.zzz”
- Compact format - ”YYYYMMDDHHmmsszzz”
- American standard - ”[Day] Mon DD YYYY HH:MM:SS:ZZZ”

where *Day* is ignored and can be excluded, and is the name of the day in three letters, e.g. ”Wed”. *Mon* is the name of the month in three letters, e.g. ”Oct”. *DD* is the date of the month, *YYYY* is the year in four digits, MM is the month in two digits, DD, *HH*, *mm*, *ss* is day, month, hours, minutes and seconds respectively in two digits and zzz is milliseconds in three digits.

```
var someDate = new Date(”2006-10-11 13:21:39.168”);
```

The Date constructor which takes a string is compatible with the function *Date.prototype.toString().* This means that Date objects can be converted into strings and back again without losing any information.

new Date(year, month[, date[, hours[, minutes[, seconds[, ms]]]]])

You can also indicate at least another two parameters to the Date constructor to create a Date object which represents a certain point in time. The parameters you exclude will get the value 0. The following script creates a Date object which contains the time for Christmas Eve 1963. It turns out to be a Tuesday.

```
var christmas1963 = new Date(1963, 12, 24);
Debug.Print(christmas1963);
```

Date()

If the Date constructor is called as a function, a string will be returned containing the date of today. All parameters will be ignored.

Note

Do not forget to write ***new*** Date(…) if you intend to create a new Date object. If you leave ***new*** out, a string containing today's date will always be created, regardless of which parameters you send in.

Date.parse(string)

The function Date.parse() converts its argument to a string and tries to interpret the string as a time in one of the following formats:

- ISO-standard – ”YYYY-MM-DD HH:mm:ss.zzz”
- Compact format - ”YYYYMMDDHHmmsszzz”
- American standard - ”[Day] Mon DD YYYY HH:mm:ss.zzz”

Where *[Day]* is ignored and can be excluded, and is the name of the day in three letters, e.g. ”Wed”. *Mon* is the name of the month in three letters, e.g. ”Oct”. *DD* is the date of the month, *YYYY* is the year in four digits, *HH*, *mm*, *ss* and *zzz* is hours, minutes, seconds and milliseconds respectively.

3.19.3 The Date prototype

Date.prototype.toString([format[, timezone]])

Date objects can easily be converted into strings. The following call converts myDate to a string and displays it in the debug window.

```
Debug.Print (myDate.toString());
```

You can provide toString() with one or two optional arguments in order to get the resulting string in the format you prefer. The first optional parameter ***format*** specifies the format of the resulting string:

- If ***format*** is ”compact” the resulting string is in the compact format: ”YYYYMMDDHHmmsszzz”.
- If ***format*** is ”long” the resulting string is in the format: Day Mon DD ”YYYY HH:mm:ss:zzz”.
- Any other value, including no value, for ***format*** results in a string in ISO format:
  ”YYYY-MM-DD HH:mm:ss.zzz”.

YYYY	Year in four digits
MM	Month in two digits
DD	Day of the month in two digits
HH	Hours in two digits
mm	Minutes in two digits
ss	Seconds in two digits
zzz	Milliseconds in two digits
Day	the name of the day in three letters e.g. ”Wed”.
Mon	the name of the month in three letters e.g. ”Oct”.

The Date constructor which takes a string is compatible with the function *Date.prototype.toString()*. This means that Date objects can be converted into strings and back again without losing any information.

The second optional parameter ***timezone*** specifies what time specification to use during conversion, if value ”utc” is specified the resulting string will be specified in ”Coordinated Universal Time”, all other values, including not providing this param, will result in local time.

Date.prototype.toDateString()

Returns the date part of the Date object as a string. The call

```
myDate.toDateString()
```

would return a string like ”2006-10-11”

Date.prototype.toTimeString()

Returns the time part of the Date object as a string.

```
myDate.toTimeString()
```

would return something like ”13:21:39.168”

Date.prototype.toUTCString()

Returns a string for the Date object according to UTC time. Generates the same kind of output as *toString()* without parameters.

Date.prototype.toLocaleDateString()

Returns a string for the date part of the Date object, using the systems locale.

Date.prototype.toLocaleString()

Returns a string of the Date object, generated using the systems locale.

Date.prototype.toLocaleTimeString()

Returns a string for the time part of the Date object, using the systems locale.

Date.prototype.getTimezoneOffset()

Returns the difference in minutes between local time and UTC time.

Example:

If your timezone is GTM+1 then -60 will be returned if winter time is active, or -120 if summer time is active.

Date.prototype.valueOf()

The following call converts myDate to a number, i.e. the number of milliseconds since the epoch, and displays it in the debug window:

```
Debug.Print (myDate.valueOf());
```

The Number constructor converts its parameter to a number when it is called as a function. The following script is equivalent to the above:

```
Debug.Print (Number(myDate));
```

Date.prototype.getTime()

Another way to convert a Date object to a number is by using the function *getTime*().

```
myDate.getTime()
```

would return something like: 1160573057929.

Date.prototype.getYear(),
Date.prototype.getMonth(),
Date.prototype.getDay(),
Date.prototype.getHours(),
Date.prototype.getMinutes(),
Date.prototype.getSeconds(),
Date.prototype.getMilliseconds(),
Date.prototype.getUTCYear(),
Date.prototype.getUTCMonth(),
Date.prototype.getUTCDay(),
Date.prototype.getUTCHours(),
Date.prototype.getUTCMinutes(),
Date.prototype.getUTCSeconds(),
Date.prototype.getUTCMilliseconds()

Date objects have a number of functions available whose names all begin with ”*get*”. These are used to break down the Date object in smaller time units.

The following script displays information about the Date object in the debug window:

```
// Create a string containing information about
// the Date object
var s = "Local date:\n";
s += "Year: " + myDate.getYear() + "\n";
s += "Month: " + myDate.getMonth() + "\n";
s += "Day: " + myDate.getDay() + "\n";
s += "Hours: " + myDate.getHours() + "\n";
s += "Minutes: " + myDate.getMinutes() + "\n";
s += "Seconds: " + myDate.getSeconds() + "\n";
s += "Milliseconds: " +
myDate.getMilliseconds()+"\n\n";
s += "UTC date:\n";
s += "Year: " + myDate.getUTCYear() + "\n";
s += "Month: " + myDate.getUTCMonth() + "\n";
s += "Day: " + myDate.getUTCDay() + "\n";
s += "Hours: " + myDate.getUTCHours() + "\n";
s += "Minutes: " + myDate.getUTCMinutes() + "\n";
s += "Seconds: " + myDate.getUTCSeconds() + "\n";
s += "Milliseconds: " + myDate.getUTCMilliseconds();

// Show the string in the debug window;
Debug.Print(s);
```

.

Date.prototype.setTime()

You can also change the time value the Date object will store. You can choose to set the time as a number of milliseconds since the epoch by calling setTime(). The following example displays which time is positioned 1 000 000 000 milliseconds after the epoch.

```
myDate.setTime(1e9);
Debug.Print(myDate);
```

1×10^9 is written in the script language as 1e9. One billion milliseconds after the epoch turns out to be Monday, January 12th, 1970 at 13:46:40.

Date.prototype.setYear(), Date.prototype.setMonth(), Date.prototype.setDay(), Date.prototype.setHours(), Date.prototype.setMinutes(), Date.prototype.setSeconds(), Date.prototype.setMilliseconds(), Date.prototype.setUTCYear(), Date.prototype.setUTCMonth(), Date.prototype.setUTCHours(), Date.prototype.setUTCMinutes(), Date.prototype.setUTCSeconds(), Date.prototype.setUTCMilliseconds()

Date object have a number of functions available whose names all begin with ”*set*”. These are used to change some part of the Date object.

Example:

```
// sets the year to 2010 and the month to December
myDate.setYear(2010);
myDate.setMonth(12);

// the same thing can be done in UTC
myDate.setUTCYear(2010);
myDate.setUTCMonth(12);
```

3.20 File Object

File objects have a few functions that you can use to read from and write to text files.

Example that reads the contents from the file ”TestIn.txt” and writes the contents to the file ”TestOut.txt”:

```
var fileIn  = new File("TestIn.txt", OPEN_ALWAYS);
var fileOut = new File("TestOut.txt", CREATE_ALWAYS);
fileOut.write(fileIn.read());
```

3.20.1 Properties in File Objects

File objects have no special properties.

3.20.2 The File Constructor

new File(fileName, creationDisposition)

Creates a new file object with the specified file name and any of the following flags.

CREATE_NEW creates a new file. Fails if the file already exists.

CREATE_ALWAYS creates a new file. If it already exists, it is overwritten.

OPEN_EXISTING opens an existing file. Fails if it does not exist.

OPEN_ALWAYS opens the file. If it does not exist, a new file is created.

TRUNCATE_EXISTING opens the file and truncates it so its size is 0 bytes. Fails if the file does not exist.

3.20.3 The File Prototype

File.prototype.atEnd()

Returns whether the file pointer is at the end of the file (EOF)

File.prototype.close()

Closes an open file.

File.prototype.getSize()

Returns the size of the file in bytes.

File.prototype.isOpen()

States whether the file is open.

File.prototype.lastModification()

Returns date and time for the last modification of the file in a date object.

File.prototype.position()

Returns the current position of the file pointer relative to the beginning of the file.

File.prototype.read([maxSizeToRead])

Reads all data from the file or up to *maxSizeToRead* bytes.

File.prototype.readLine()

Reads data from the file until EOL is found or 8192 bytes have been read and returns the string including the EOL character(s). EOL is CR and/or NL.

File.prototype.seek(relativePos[, reference = FILE_BEGIN])

Sets the file pointer at the position ***relativePos*** relative to the ***reference***. If you don't define ***reference***, the position is set relative to the beginning of the file. The method also returns the new file position relative to the beginning of the file.

File.prototype.setFilePointer(reference[, relativePos = 0])

Sets the file pointer at the position ***relativePos*** relative to the ***reference***. ***reference*** can be ***FILE_BEGIN, FILE_CURRENT*** or ***FILE_END***. If you don't define ***relativePos***, 0 will be used, which means that the file pointer is set to the beginning of the file, will remain where it is or be set to the end of the file. The method also returns the new file position relative to the beginning of the file.

File.prototype.write(data)

Writes ***data*** to the current position in the file.

3.21 Clients object

The Clients object is a helper object that is available from license level *Advanced* and upwards. The object represents a number of Ethiris Clients and is used for sending commands to one or several Ethiris Client.

Using 4 different ”filters” you can create a command object that in some sense has a dynamic list of desired clients. Strictly speaking, the command object contains a filter that is matched against all connected clients at the time when a command is sent. The command is sent to all connected clients matching the filter. The 4 filters are: *All*, *GetClientByAddress*, *GetClientByConfigName* and *GetClientByHostName*. These are described below.

When you got your command object you can send various commands such as *SelectView*, *ShowDynamicView*, *ShowPopupView,* etc. These are also described below.

3.21.1 Methods in the Clients object

All()

This ”filter” is used for creating a command object where commands will be sent to all Ethiris Clients that are currently connected to the Ethiris Server. Note that the number of clients may vary from time to time when clients disconnect and new clients connect to the server.

Example:

```
var clients;
var oldTrigger;

if (clients == undefined)
{
          // Get all clients
          clients = Clients.All();
}

if (trigger && !oldTrigger)
{
          // Reload configuration in all
          // clients when 'trigger' is true
          clients.ReloadConfiguration(true);
}

oldTrigger = Boolean(trigger);
```

This script creates a command object for all connected clients and assigns it to the variable *clients*. When *trigger* becomes *true* the command *ReloadConfiguration(true)* is sent to all the connected clients and thus forces a reload of their configurations.

GetClientByAddress(address)

This ”filter” is used for creating a command object where commands will be sent to the Ethiris Clients that are connected to the Ethiris Server and runs on a computer with the IP address passed as an argument

Example:

```
var clients;
var oldTrigger;

if (clients == undefined)
{
          // Get specific client
          clients =
Clients.GetClientByAddress("192.168.31.123");
}

if (trigger && !oldTrigger)
{
          // Show view 'View One' when trigger is true
          clients.SelectView("View One");
}

oldTrigger = Boolean(trigger);
```

This script creates a command object for all connected clients with IP address *192.168.31.123*, which should be only one. When trigger becomes true a command for selecting a view called *View One* is sent.

GetClientByConfigName(name)

This ”filter” is used for creating a command object where commands will be sent to the Ethiris Clients that are connected to the Ethiris Server and runs a configuration with the name passed as an argument

Example:

```
// Get clients running ‘Test’ conf
clients = Clients.GetClientByConfigName("Test");
```

GetClientByHostName(name)

This ”filter” is used for creating a command object where commands will be sent to the Ethiris Clients that are connected to the Ethiris Server and runs on a computer with the name passed as an argument

Example:

```
// Get the client on host ‘Chaotic’
clients = Clients.GetClientByHostName("Chaotic");
```

3.21.2 Methods in a command object

When you have got a command object for sending commands to one or several clients there are different commands that can be sent to the clients. These are described below.

For each command that is sent, you get a corresponding *Event* called *onCommandCompleted* which, among other things, contains information about the result of the command.

SelectView(viewName, [sectionName])

This method is used for selecting a view in the clients that match the filter in the command object.

viewName is mandatory and is the name of the view you want to select.

sectionName is optional to pass as an argument. It is the name of the section where the desired view is defined.

Return value is a transaction object that in turn contains *transaction ID* and the number of clients the command was sent to. The Transaction object is described a little later in the manual. If you care about neither transaction ID nor the number of clients the command was sent to, you don't need to handle the return value of the method.

Example:

```
// Select view
trans = clients.SelectView("MyView", "MySection");
```

SelectSection(sectionName, [viewName])

This method is used for selecting a section in the clients that match the command object filter.

sectionName is mandatory and is the name of the section you want to select.

viewName is optional. It is the name of a view you want to select at the same time you select a section. A possible view has to be defined in the selected section.

Return value is a transaction object.

Example:

```
// Select section
// We don't care about the transaction object
clients.SelectSection("MySection");
```

ShowDynamicView(monitor, [cameras])

This method is used for opening a dynamic live view in the clients that match the command object filter. The live view is always maximized at the selected monitor.

monitor is mandatory and is supposed to be a number between 1 – 8 that specifies at which monitor the live view will be opened. Via this method, a maximum of one window at a time can be open. If the window is already open when the command is sent with the argument *cameras* the content in the window will be exchanged to the new cameras.

cameras is optional. If this argument is omitted, the dynamic window will be closed on the selected monitor. When the argument *cameras* is used it is supposed to be an array with one or several cameras. Each element in the array represents a camera. Each element can represent a camera in three different ways; *Camera object (the camera script name)*, *CameraID* or *Camera name (the camera name in Ethiris Client)*. Note that you can have "gaps" in the array, which means that the corresponding camera view will be empty.

E.g. the array ["Cam1", "Cam2", , "Cam4", , "Cam6"] creates a view with 6 camera views of which two are empty.

Return value is a transaction object.

Example:

```
// Show dynamic view with 3 cameras on monitor 2
// Camera 1 is a camera object
// Camera 2 is by ID
// Camera 3 is by name in client
clients.ShowDynamicView(2, [ Door, Garage.ID,
"Reception" ]);
```

Note the square brackets surrounding the cameras'[' and ']' respectively. These create an array with, in this case, three elements.

ShowPopupView(popupName [, viewName [, cameras[, focusCamera]]])

This method is used for opening a pre-defined popup window in the clients that match the filter in the command object.

popupName is mandatory and should be the name of a pre-defined popup window that is defined in the client configuration. In the popup windows definition is which monitor, size, and position of the window defined.

viewName is optional. If neither this nor *cameras* are passed as the argument the popup window will be closed in the client. If viewName is passed, it's supposed to be the name of a pre-defined client view in the client configuration. This view will be displayed in the popup window. The popup window will be opened if it's not already open. Note that the selected view may have several camera views of which each one can be either undefined or have a defined content, such as that a live camera will be displayed.

cameras is also optional. It's an array with one or several cameras in the same way as for *ShowDynamicView*. If *cameras* is not passed to the method, either the view specified with *viewName* is displayed (if viewName is passed to the method) or the popup window will be closed (if neither viewName nor cameras is passed to the method). If *cameras* is passed in, the cameras in the array will be displayed in the following way: If *viewName* also is passed, the cameras in the array will populate the undefined camera views, i.e. those camera views that are empty. Thus you can mix pre-defined camera views with cameras specified in the array *cameras*. If *viewName* is not passed into the method, the cameras specified in the cameras array will be displayed in an automatically created view with the required number of camera views, somewhat like the command *ShowDynamicView* with the difference that the window is pre-defined.

focusCamera is also optional. It defines which camera should have focus (green rectangle around its cameraview) in the popup window that is created. If none is defined, the first cameraview will get focus.

Return value is a transaction object.

Example:

```
// Show popup view
clients.ShowPopupView("PopWin", "MyView", [ Door ]);
```

ShowInstantReplayView(monitor|popupName [, camera [, <instant replay time>]])

This method is used for opening (or closing) a window with an instant replay view.

monitor|popupName is mandatory and should be the name of a pre-defined popup window that is defined in the client configuration or a number referring to the monitor you want to show the window on. Valid values are:

- -1 – show the window on the same monitor as the main windows of the Ethiris Client.
- 0 – show in default live panel.
- 1 – 8 – show on the monitor that corresponds to the number. The highest valid value is the number of monitors connected to the client computer.

If you only specify this parameter, the instant replay window referred to by the parameter will be closed.

camera is optional. This is the camera you want to show instant replay for. Assume you have a camera named *Camera* in your server configuration. You can refer to the camera in three ways:

- By specifying the camera object in the script:
  Camera
- By specifying the camera id:
  Camera.ID
- By specifying the name of the camera in the client configuration:
  <name of the camera in the client configuration>"

<instant replay time> is optional. This is the initial instant replay time to use. If you skip this parameter, the last used value for this camera will be used or 15s if the value has never been changed or *Time before* for event recording if less than 15s.

Return value is a transaction object.

Example:

```
// Show instant replay view with camera Camera in
// popup window PopWin.
// Use default instant replay time
clients.ShowInstantReplayView("PopWin", Camera);

// Show instant replay view with camera Camera on
// same monitor as Ethiris main window
// Use instant replay time of 30s
clients.ShowInstantReplayView(0, Camera.ID, 30);

// Close instant replay view on
// popup window PopWin.
clients.ShowInstantReplayView("PopWin");
```

ReloadConfiguration([forceReload])

This method is used for reloading the configuration in the clients that match the command object filter.

forceReload is optional and is supposed to be a *Boolean* value, i.e. *true* or *false*. Now, *false* is default, so it only makes sense to pass in *true*. If you don't force the reload, the client itself decides if the configuration has to be reloaded. It will if the configuration has been changed. If you choose to force the reload, the client will reload its configuration regardless of whether it is necessary or not.

Return value is a transaction object.

Example:

```
// Reload configuration
clients.ReloadConfiguration();
```

PlaySound(soundName, [stop])

This method is used for starting playback of a sound in the clients that match the command object filter.

soundName is the name of the sound as defined in the client configuration. If the defined sound has the *Loop* option checked, the sound will keep on playing until a stop command is sent. Otherwise, the sound will be played once. Note that the script is executed over and over again. If the *PlaySound* command is executed every cycle in the script, the sound will be played for each round of execution. Therefore it's advisable to use edge detection as in the following example.

stop is optional and is supposed to be a *Boolean* value, i.e. *true* or *false*. Now, *false* is default, so it only makes sense to pass in *true*. If you don't pass a value the sound will be played. If you pass *true* the playback of the sound will be stopped.

Return value is a transaction object.

Example:

```
// Play sound on motion in front of 'FenceCam'
var client = Clients.GetClientByHostName("Chaotic");
var edgePlaySound;

if (FenceCam.MotCars.Motion && !edgePlaySound)
        client.PlaySound("CarSound");

edgePlaySound = FenceCam.MotCars.Motion;
```

StopSound(soundName)

This method is used for stopping playback of a sound in the clients that match the command object filter.

soundName is the name of the sound as defined in the client configuration. This command is usually used when the sound is in *loop mode*.

Return value is a transaction object.

Example:

```
// Stop sound using a push button ('buttonStopSound')
var client = Clients.GetClientByHostName("Chaotic");
var edgeStopSound;

if (buttonStopSound && !edgeStopSound)
        client.StopSound("CarSound");

edgeStopSound = Boolean(buttonStopSound);
```

LoadCamerasInPlayerByTimestamp(timestamp, [cameras])

This method is used to load a set of cameras in the client's player.

timestamp is a string representing the date and time you want the player to start at.

cameras is an array with one or several cameras to load.

Both or either of these parameters can be used. If *timestamp* is omitted, then the player will load the supplied cameras at the time the player is currently at. If ***cameras*** is omitted the player will go to the specified time with the currently loaded camera(s).

Return value is a transaction object.

Example:

```
// Three different ways to open cameras in the player
var client = Clients.GetClientByHostName("Chaotic");

client.LoadCamerasInPlayerByTimestamp(["Camera 2",
"Camera 3"]);

client.LoadCamerasInPlayerByTimestamp("2012-08-24
10:30:00");

client.LoadCamerasInPlayerByTimestamp(("2013-03-27
14:45:00", ["Camera 4"]);
```

StartPlayer()

This method is used to start the player in the client.

Return value is a transaction object.

Example:

```
// Start Player
var client = Clients.GetClientByHostName("Chaotic");

client.StartPlayer();
```

PausePlayer()

This method is used to pause the player in the client.

Return value is a transaction object.

Example:

```
// Pause Player
var client = Clients.GetClientByHostName("Chaotic");

client.PausePlayer();
```

MoveToNextRecording()

This method is used to seek forward to the next available recording for the currently loaded camera(s).

Return value is a transaction object.

Example:

```
// Move to next recording
var client = Clients.GetClientByHostName("Chaotic");

client.MoveToNextRecording();
```

MoveToPrevRecording ()

This method is used to seek back to the previously available recording for the currently loaded camera(s).

Return value is a transaction object.

Example:

```
// Move to previous recording
var client = Clients.GetClientByHostName("Chaotic");

client.MoveToPrevRecording();
```

GetDisplayedCameras()

This method is used to get an updated list of cameras that are currently displayed in the Client. The list will be sent by using the onDisplayedCameras event.

Exempel:

```
// Get Cameras displayed in the Client
clients.GetDisplayedCameras();
```

3.21.3 Events in the Clients object

There are a number of different events you can listen to in your script. Common for all of them is that you get access to an *event* object that has a number of properties. Exactly which properties are available is determined by the type of event.

onConnect

This event is fired when an Ethiris Client is connected to the server and is prepared to receive commands for remote control.

Example:

```
Clients.onConnect = function(event)
{
        sResult = event.HostName + " - ";
        sResult += event.ConfigurationName;
}
```

This example will assign to the variable *sResult*, both the name of the computer where the client runs and has connected to the server and furthermore the name of the configuration that is loaded in the client.

The following properties are available for the *event* object:

IpAddress contains the IP address of the computer where the connected client runs.

HostName contains the name of the computer where the connected client runs.

ConfigurationName contains the name of the configuration that is loaded in the connected client.

ClientType contains a value representing the type of client that has connected to the server. The possible values exist as constants directly for the *Clients* object. These are *ETHIRIS_ADMIN*, *ETHIRIS_CLIENT*, *ETHIRIS_ACTIVEX_VIEWER*, *ETHIRIS_MOBILE_CLIENT,* and *WIDEQUICK_CLIENT*.

Connected contains *true* if the client is connected, and false if it’s not connected.

onDisconnect

This event is fired when an Ethiris Client is disconnected from the server.

Example:

```
Clients.onDisconnect = function(event)
{
        sResult = event.HostName + " - ";
        sResult += event.ConfigurationName;
}
```

The example above will assign to *sResult* the name of the computer where the disconnected client ran and furthermore the name of the configuration that was loaded in the client.

The properties of the *event* object are exactly the same as for *onConnect* described above.

onCommandCompleted

This event is fired when a command has been received and handled by an Ethiris Client. You can use this event for controlling the result of the command.

Example:

```
Clients.onCommandCompleted = function(event)
{
          sResult = event.TransactionID;

          if (event.Result < Clients.RESULT_OK)
          {
                    // Something went wrong...
          }
}
```

This example will assign the command's transaction ID to *sResult*. Then there is a check that the result is OK. Read more about the various result codes under *Constants* below. Generally speaking, *RESULT_OK* has a value of 0. A negative result indicates an error. A positive result indicates a warning.

The following properties are available for the *event* object:

IpAddress, ***HostName***, ***ConfigurationName,*** and ***ClientType*** are the same as above. Besides these there also are:

TransactionID is a running number that contains the current transaction ID. This will match the ID returned in the Transaction object when the corresponding command was sent.

Result contains a value for the result of the command. See below for a list of possible values.

onButtonPressed

This event is fired when a user clicks a button in a camera view in the client. This event is fired regardless if a variable is connected to the button's click-event or not. The upside of this event is that you in script get information about in which client the user has clicked a button. This can be used to do different things depending on in which client the button has been clicked.

The event is fired both when the button is pressed and when it is released.

Exemple:

```
Clients.onButtonPressed = function(event)
{
          //Do something
}
```

The following properties are available for the *event* object:

IpAddress, ***HostName***, ***ConfigurationName,*** and ***ClientType*** are the same as above. Besides these there also are:

ButtonName is the text that is assigned to the button if any.

CameraID *is* id for the camera that is shown in the view where the button is shown.

CameraName is the name in the server of the camera that is shown in the view where the button is shown.

Name is the name of the client as defined in *Remote Clients.*

Pressed id *true* if the button is clicked and false if the button is released.

ViewName is the name of the view where the button id defined.

onViewClicked

This event is fired when a user clicks somewhere in a camera view in the client. The upside of this event is that you in script get information about in which client the user has clicked a camera view. This can be used to do different things depending on in which client the button has been clicked.

The event is fired both when the mousebutton id pressed and released.

Exemple:

```
Clients.onViewClicked = function(event)
{
          //Do something
}
```

The following properties are available for the *event* object:

IpAddress, ***HostName***, ***ConfigurationName, ClientType, CameraID, CameraName, Name, Pressed*** and ***ViewName*** are the same as above. Besides these there also are:

FrameTime is the timestamp of the camera image shown in the camera view where the user clicked.

ImageClickPositionX is the x-coordinate in the unscaled camera image where the user clicked.

ImageClickPositionY is a y-coordinate in the unscaled camera image where the user clicked.

ImageSizeX is the width of the unscaled camera image.

ImageSizeY is the height of the unscaled camera image.

MouseButton is a string that tells which mouse button the user clicked on. Possible alternatives are *Left, Middle* and *Right*.

Panel is the name of the panel in which the user has clicked. Possible alternatives are *Live* and *Player.*

PanelSizeX is the width of the panel or the window where the view is shown that the user clicked in.

PanelSizeY is the height of the panel or the window where the view is shown that the user clicked in.

ViewClickPositionX is the x-coordinate in the camera view where the user has clicked.

ViewClickPositionY is the y-coordinate in the camera view where the user has clicked.

ViewName is the name of the current view where the user has clicked.

ViewPositionX is the x-coordinate of the view's top left corner in the panel.

ViewPositionY is the y-coordinate of the view's top left corner in the panel.

ViewSizeX is the width of the view where the user has clicked.

ViewSizeY is the height of the view where the user has clicked.

onPlayerLoaded

This event fires when one or more cameras are loaded in the player.

Exemple:

```
Clients.onPlayerLoaded = function(event)
{
          //Do something
}
```

The following properties are available for the *event* object:

IpAddress, ***HostName***, ***ConfigurationName, ClientType, Name,*** and ***ViewName*** are the same as above. Besides these there also are:

Cameras is a list of cameras there are loaded in the player. Note that it is cameras that exist in the server configuration, i.e. if a virtual camera is loaded in the player, the corresponding source camera (the 360 camera) will be added to the list.

Time is the timestamp of the timelines in the player. That means that the timestamp in the camera images might differ slightly from this time.

onPlayerTime

This event fires when the time changes for the timelines or player state changes in the player. The state of the player is for example if the player is playing back recorded video or if it is in pause.

Exemple:

```
Clients.onPlayerTime = function(event)
{
          //Do something
}
```

The following properties are available for the *event* object:

IpAddress, ***HostName***, ***ConfigurationName, ClientType, Name,*** and ***ViewName*** are the same as above. Besides these there also are:

PlayerState is the state of the playback in the player. Possible alternatives are *Pause* and *Play*.

Time is the timestamp of the timelines in the player. This means that the timestamp if the images might differ slightly.

onDisplayedCameras

This event fires when called by GetDisplayedCameras() or when a change of what cameras are shown in the client occurs.

Exempel:

```
Clients.onDisplayedCameras = function(event)
{
        for(var j in event["ClientCameraNames"])
        {
                var s = "Camera Displayed: "
                s +=event["ClientCameraNames"][j];
                Debug.Print(s);
        }
}
```

The following properties are available for the *event* object:

ClientCameraNames contains the names of the displayed cameras.

3.21.4 Constants in the Clients object

Client types

These constants are used in the events *onConnect* and *onDisconenct* for the property *ClientType* in the *event* object.

ETHIRIS_ADMIN - 1

ETHIRIS_CLIENT - 2

Reserved - 3

ETHIRIS_ACTIVEX_VIEWER - 4

ETHIRIS_MOBILE_CLIENT - 5

WIDEQUICK_CLIENT - 6

Result

These constants are used in the event *onCommandCompleted* for the *Result* property in the *event* object. The first three constants are merely warnings, which means that the command has been executed as far as possible, but with some minor shortcomings. E.g. if one of the cameras specified in the *cameras* array is missing, the remaining cameras will be displayed and the missing camera will be indicated in the camera view with the text *Camera missing*.

The constants after *RESULT_OK* are considered errors, which means that the command can't be executed.

RESULT_VIEW_NOT_FOUND – 3. The view name passed as an argument does not exist in the client configuration.

RESULT_CAMERA_NOT_FOUND – 2. One of the cameras in the array *cameras* does not exist in the client configuration.

RESULT_WILL_RELOAD – 1. Is valid for the command *ReloadConfiguration* and means that the client has detected that its configuration has been updated and that the client will reload the configuration eventually.

RESULT_OK - 0. The command has been executed by the client without problems.

RESULT_NO_COMMAND – -1. Reserved for future use. This should not happen.

RESULT_MISFORMED_COMMAND – -2. Reserved for future use. This should not happen.

RESULT_UNKNOWN_COMMAND – -3. Reserved for future use. This should not happen.

RESULT_POPUPWINDOW_MISSING – -4. The argument *popupName* is missing or contains a name whose corresponding popup window does not exist in the client configuration. This result is valid for the command *ShowPopupView*.

RESULT_MONITOR_MISSING – -5. The argument *monitor* is missing or contains a value that does not match an existing monitor.

RESULT_SECTION_MISSING – -6. The argument *sectionName* is missing or contains a name of a section that does not exist in the client configuration.

RESULT_VIEW_MISSING – -7. The argument *viewName* is missing or contains the name of a client view that does not exist in the client configuration.

RESULT_SOUND_MISSING – -8. The argument sound*Name* is missing or contains a name of a sound that does not exist in the client configuration.

3.22 Transaction object

The Transaction object is an object that is returned from all of the five different commands that can be sent via a command object. The commands are: *SelectView*, *SelectSection*, *ShowDynamicView*, *ShowPopupView* and *ReloadConfiguration*. The Transaction object contains information about the transaction ID of the command and the number of clients the command was sent to.

3.22.1 Properties of the Transaction object

TransactionID

This is a running number that is increased by one each time a command is sent.

Example:

```
var clients;
var oldTrigger;
var transObj;

if (clients == undefined)
{
   // Get all clients
   clients = Clients.GetClientByHostName("Chaotic");
}

if (trigger && !oldTrigger)
{
   // Show dynamic view on monitor 2
   transObj = clients.ShowDynamicView(2, ["Door" ]);
}

oldTrigger = Boolean(trigger);

Clients.onCommandCompleted = function(event)
{
   // Check transaction ID
   if (transObj.TransactionID == event.TransactionID)
   {
      if (event.Result >= Clients.RESULT_OK)
      {
         // ShowDynamicView command successful...
      }
   }
}
```

In the example above, the transaction object is used in the event *onCommandCompleted* to verify that it is the correct command that has been completed. There may be several commands running simultaneously for several clients.

NumClients

This property contains the number of clients that the command was sent to.

Example:

```
transObj = clients.ReloadConfiguration(true);
sResult = transObj.NumClients + " clients updated";
```

3.23 Cameras object

The Cameras object is an object that contains an array of all the cameras that are defined in the server configuration, both IP cameras and analog cameras connected via video encoders.

Each element in the array is a reference to a real camera object and can thus be used like any other camera object in the script. All camera variables are available such as *RecordEvent* and *RecordContinuous*.

In the example below the cameras object, is used for starting event recording on every camera in the server.

Example:

```
for(var cam in Cameras)
{
   Cameras[cam].RecordEvent = true;
}
```

3.23.1 Events in Cameras object

The Cameras object has the following events:

onEvent(event)

This event bubbles up from each camera object and will get called when the event arrives for any camera. See *3.24* below for a description of the event and its arguments. If you need to perform the same action on events from several similar cameras this is a convenient way to handle it all in one place.

The example shows how to use people detection analytics in a Flir thermal camera to start event recording on the same camera.

```
Cameras.onEvent = function(event)
{
     var pd = event.Data["Data"]["State"];

     if(pd != undefined)
     {
         event.target.RecordEvent = (pd == "true");
     }
}
```

Here we are looking for the property *State* on the object returned from property *Data* on the *event.Data* array. If this property is present the variable *pd* will get the value ”true” or ”false” as a result of the analytics module. We can use this to start/stop recording on the camera. If the *event.Data* array does not contain an object named *Data* with a property *State*, it means the camera sent some other notification. In this case, the variable *pd* will get the special value *undefined* and we should do nothing.

The *event.target* property holds a reference to the camera object representing the camera that sent the notification. On this object, we have access to all properties of the cameras, like *RecordEvent*, and we use this to trigger recording on the camera.

The code given as an example below can be used to find out what notifications, events and metadata, that arrive from any camera.

```
Cameras.onEvent = function(event)
{
    Debug.Print("");

    var s = event.target.Name + " - Topic = " +
event["Topic"];
    Debug.Print(s);

    for(var i in event.Data)
    {
        for(var j in event.Data[i])
        {
            var s = event.target.Name;
            s += " - Data[" + i + "][" + j + "] = ";
            s += event.Data[i][j];
            Debug.Print(s);
        }
    }
}
```

3.24 Camera object

The camera object represents a single camera as defined in the server configuration.

3.24.1 Methods in the Camera object

SaveImage(path)

This method is used to save the last frame from the camera as a jpg-file to disk. If the *path* is a complete path including a filename ending with .jpg, that filename will be used each time the image is saved. If *path* only contains a path (not filename), the filename will automatically be created in the following format: <cameraname>_<timestamp>.jpg.

This means that all images saved will get a unique name. Ethiris Server will never erase any of these images; you will have to handle that outside of Ethiris. Bear in mind that if unique filenames are used, there might be a lot of files in the destination folder.

Example:

```
if (save_jpg)
{
    save_jpg = false;

    //Save last image from camera as a jpg-file
    Camera_1.SaveImage("c:\Images\LastImage.jpg");
}
```

This script saves the last frame from the camera under the name LastImage.jpg. Each time the image is saved, it will overwrite the current file on disk.

3.24.2 Events in Camera object

The Camera object has the following events:

onEvent(event)

Each camera in the data store can generate an event when certain things happen in the camera. This can be things like alarms, tampering detection or I/O signals connected to the camera. Or it could be the results from analysis functions locally defined in the camera, for example, motion detection, people counter or other functions.

Different camera models generate different notifications, so one must know what to look for.

Note – currently event notifications are implemented only for ONVIF cameras in Ethiris and only for cameras where the Events (*E*) check box is checked in Ethiris Admin.

The event can be tied to a function that will get called each time the camera sends new notifications.

An example of usage could be if we want to detect when a PTZ camera is moving. An ONVIF compatible PTZ camera from Axis Communication will send notifications when the camera starts and stops moving in pan, tilt or zoom direction:

```
Camera_1.onEvent = function(event)
{
   var moving = event.Data["Data"]["is_moving"];

   if(moving != undefined)
   {
      if(moving == 1)
         Debug.Print("Camera moving");
      else
         Debug.Print("Camera stopped");
   }
}
```

The event argument ***event*** has a few properties besides the standard Java script event argument properties:

Data – this is an array of notification data sent from the camera. Each notification has a set of properties that is specific for that notification type, it may be different for each camera model, camera setup and notification type.

target – a reference to the camera object that represents the camera that generated the notifications. Here we can access all the camera object properties like *Name*, *RecordEvent*, *Recording,* and many others.

3.25 CameraCollection object

The CameraCollection object represents a dynamic collection of cameras in the server configuration and becomes a reference to the real camera objects it contains. This makes several variables available, for instance, Enable, RecordEvent and RecordContinious. These are set for all the cameras in the collection at the same time. In the example below the object is used to start an event recording on all the added cameras.

Example:

```
var camColl = new CameraCollection(Cam_1, Cam_2);
camColl.RecordEvent = true;
```

3.25.1 Methods in the CameraCollection object

Add([cameras])

This method is used to add one or more cameras to a camera collection.

cameras is mandatory and should contain an array with one or more cameras to add. Every element in the array represents a camera using a Camera object (the cameras script name). If any of the supplied cameras already exist in the collection, no change is made to them, but the remaining is added.

E.g. the array [Camera_1, Camera_2, Camera_3] becomes a collection of three cameras.

Return value is the number of Camera objects in the collection.

Remove([cameras])

This method is used to remove one or more cameras from a camera collection.

cameras is mandatory and should contain an array with one or more cameras to add. Every element in the array represents a camera using a Camera object (the cameras script name).

E.g. the array [Camera_1, Camera_4] removes Camera_1 from the collection, but Camera_4 is not in the collection, hence it is ignored.

Return value is the number of Camera objects in the collection.

Example:

```
camColl = new CameraCollection(Cam_3, Cam_4);
Debug.Print("Amount: " + camColl.Add(Cam_1));
Debug.Print("Amount: " + camColl.Remove(Cam_4));
```

This script creates a collection with two cameras. Then a third camera is added, and the number of cameras in the collection is printed. Then one camera is removed and the new number of cameras in the collection is printed.

3.26 Storages object

The Storages object is a helper object. The object represents a number of cameras with available recorded material, matching the search criteria.

3.26.1 Methods in the Storages object

GetCamerasWithRecordings(tsStart, tsEnd)

This method is used to locate cameras with recorded material available between the defined start and end times.

tsStart is a mandatory string representing the date and time you want to use as a startingpoint for locating recorded material.

tsEnd is a mandatory string representing the date and time you want to use as a endpoint for locating recorded material.

Return value is a camera object.

Example:

```
var cams = Storages.GetCamerasWithRecording(tsStart,
tsEnd);
Debug.Print("GetCamerasWithRecording: " +
cams.length);
for(var i in cams)
   Debug.Print(i + " | " + cams[i].ID);
```

And can also be used for instance to load these cameras in the client's player.

Example:

```
clients.LoadCamerasInPlayerByTimestamp(tsStart, cams);
```

3.27 Database object

The database object represents a single database (connection) as defined in the server configuration.

3.27.1 Methods in the Database object

ExecuteQuery(sqlStatement)

This method is used to execute a SQL statement. The SQL statement supports all SQL commands that can be written as a string. Keep in mind that this command is very powerful! Therefore, make sure you know what you do before you run a command on a database containing data.

This command executes the SQL statement synchronously. This means that the script will wait on this line until the database has executed the SQL statement. If it is a complex statement, this could take some time. Consider using *AsyncQuery* instead. There is a max running time on the script. If script execution in Ethiris takes too long, script execution will be aborted with a runtime error. The max running time for the script is 20s.

sqlStatement is mandatory. This is the SQL statement that should be executed. Consult the manual of your database to get help regarding writing SQL statements for your database.

Return value is a *database query* object that in turn contains a transaction ID and the result of the SQL statement.

Example:

```
var res = DB.ExecuteQuery("insert into mytable (cola,
colb) values (10, 20);");
```

This script assumes that there is a defined database (connection) named *DB* and in the database used there is a table called *myTable* with two columns called *cola* and *colb.* The variable *res* contains a *database query* object that contains the result of the statement as well as status information on how it went, if something went wrong or similar.

The script creates a new post in the *myTable* table.

AsyncQuery (sqlStatement)

This method is used to execute a SQL statement. The SQL statement supports all SQL commands that can be written as a string. Keep in mind that this command is very powerful! Therefore, make sure you know what you do before you run a command on a database containing data.

This command executes the SQL statement asynchronously. This means that the script immediately continues to the next line of code. The actual database call will run in the background and when the execution is finished, an event is generated that can be listened for either on the result object returned from the *AsyncQuery* method or on the database object itself.

sqlStatement is mandatory. This is the SQL statement that should be executed. Consult the manual of your database to get help regarding writing SQL statements for your database.

Return value is a *database query* object that in turn contains a transaction ID and the result of the SQL statement. This object will generate an event when execution is finished or en error occurs.

Example:

```
var res = DB.AsyncQuery("select * from mytable;");
```

This script assumes that there is a defined database (connection) named *DB* and in the database used there is a table called *myTable.* The variable *res* contains a *database query* object that contains the result of the statement as well as status information on how it went, if something went wrong or similar.

This SQL statement returns all rows in the table *myTable.*

Note!

If you want to start multiple asynchronous database questions in parallel, you must save the *database query* objects in different variables. This because it is in these objects that the actual database call is happening.

```
var res1 = DB.AsyncQuery("select * from mytable1;");

var res2 = DB.AsyncQuery("select * from mytable2;");

var res3 = DB.AsyncQuery("select * from mytable3;");
```

In this way, you can get events when each database request has been completed and the result is available.

3.27.2 Events in Database object

The Database object has the following events:

onQueryFinished(event)

Each database in the data store can generate an event when an asynchronous database call is complete.

An easy way to find out when an asynchronous database query is complete is to use the *onQueryFinished* event.

In the example below, the values from all fields from the last post in the table are written to the *Debug* window.

```
DB.onQueryFinished = function(event)
{
     for(var c in event.target.Columns)
     {
          Debug.Print(event.target.Columns[c] + " = " +
event.target.Rows[event.target.RowCount-1][c]);
     }
}
```

The event argument *event* has a few properties besides the standard Java script event argument properties:

currentTarget - This is a reference to the database object the event is generated on (in this case *DB*).

target - This is a reference to the *database query* object that was returned from the *AsyncQuery* call. It is in this object that the result of the database query is available. See also the description of the *database query* object below.

3.28 Database query object

Database query object is an object returned from the *AsyncQuery* call on a database object. The database query object contains information about the query and will eventually contain the result of the SQL statement.

The database query object must exist as long as the database query is running and hence must be stored in a local variable in the script.

3.28.1 Methods on Database query object

Fetch() - Is used to fetch the next row from the result. If there are any more rows to return, *Fetch()* will return *true* and store the next row in the property *currentRow,* otherwise, *Fetch()* will return *false.* See the example under *CurrentRow* below.

3.28.2 Properties on Database query object

TransactionID - This is a running number that is increased each time an asynchronous query is executed, regardless of which database object they are executed on. *TransactionID* can be used to connect the id of the query with the correct event on the database object.

Example:

```
if (get_data)
{
    get_data = false;
    var res = DB.AsyncQuery("select * from myTable;");

    Debug.Print("Query started: " +
res.TransactionID);
}

DB.onQueryFinished = function(event)
{
    Debug.Print("Query finished: " +
event.target.TransactionID);

    for(var c in event.target.Columns)
    {
        Debug.Print(event.target.Columns[c] + " = " +
event.target.Rows[event.target.RowCount-1][c]);
    }
}
```

In the example above, tranaction id is written both when *AsyncQuery* is called and in the event *onQueryFinished.*

ColumnCount - The number of columns in the result from the database query. If the query doesn’t return any columns, the value will be 0.

Columns - A list of column names that are returned. The highest index in the list is *ColumnCount - 1 if ColumnCount* is greater than 0. Otherwise, the list is empty.

To display the names fo all columns that are returned in the query, do like this:

```
DB.onQueryFinished = function(event)
{
    for(var c in event.target.Columns)
    {
        Debug.Print(event.target.Columns[c]);
    }
}
```

CurrentRow - Contains a list of values for the current row returned by. Not until the method *Fetch()* has been called and returns *true* will *CurrentRow contain any values*. To iterate all rows returned from a query, you can do like this:

```
DB.onQueryFinished = function(event)
{
    Debug.Print("Query finished: " +
event.target.TransactionID);

    while(event.target.Fetch())
    {
        for(var c in event.target.CurrentRow)
        {
            Debug.Print(event.target.Columns[c] + " =
" + event.target.CurrentRow[c]);
        }
    }
}
```

Keep in mind that this is only an example. In reality, you wouldn't just print all values in the debug window, but instead, process them somehow. Also, try to write your SQL statement in a way that as few rows as possible are returned. This is to optimize performance in your system.

Error - Is an object with two properties containing an error code and a message describing the error, if any, of the SQL statement.

Error.Code - This is the error code returned. If no error, the value will be 0.

Error.Message - Is a string containing the error message. If no error, the string is empty.

RowCount - The number of rows returned by the query. If no rows are returns, the value is 0.

Status - Is the state of the query. See below for the constants in the *database query* object describing the different states that can exist. The state can have any of three values. Only if *Status* is *STATUS_FINISHED,* the result is available in the other properties.

3.28.3 Events in the Database query object

There is an event you can listen t in the *database query* object.

onQueryFinished(event)

Each database query object can generate an event when the asynchronous database query has finished.

An easy way to find out when an asynchronous database query is complete is to use the *onQueryFinished* event.

In the example below, the values from all fields from the last post in the table are written to the *Debug* window.

Note that the *database query* object doesn't exist before the call to *AsyncQuery* has been issued, that's why the extra if-statement is needed.

```
if (get_data)
{
    get_data    = false;
    var res = DB.AsyncQuery("select * from myTable;");

    Debug.Print("Query started: " +
res.TransactionID);
}

if (res != undefined)
{
    res.onQueryFinished = function(event)
    {
        Debug.Print("Query finished: " +
event.target.TransactionID);

        for(var c in event.target.Columns)
        {
            Debug.Print(event.target.Columns[c] + " =
" + event.target.Rows[event.target.RowCount-1][c]);
        }
    }
}
```

The event argument *event* has a few properties besides the standard Java script event argument properties:

currentTarget - This is a reference to the *database query* object the event is generated on (in this case *res*).

target - This is a reference to the *database query* object that was returned from the *AsyncQuery* call. It is in this object that the result of the database query is available. See also the description of the *database query* object above.

Note that the event will bubble up to the database level. You only need to listen to the event on one level, either on the *database query* object or on the *database* object.

3.28.4 Constants in the Database query object

Status

These constants can be used to determine which state the *database query* object is in.

STATUS_ERROR	-1
STATUS_RUNNING	0
STATUS_FINISHED	1

3.29 RemoteClients object

The RemoteClients object is an object that contains an array of all remote clients that are defined in the server under the node *Remote Clients*.

Each element in the array is a reference to a real remote client object and thus can be used as any other remote client object in the script. All remote client variables are available such as *Name* and *Connected*.

In the example below the object is used for getting the names of all the defined remote clients.

Example:

```
sResult = "";
for(var rc in RemoteClients)
{
   if (sResult != "")
      sResult += ", ";

   sResult += RemoteClients[rc].Name;
}
```

3.30 AccessControllers-object

The AccessControllers- object is an object that contains an array of all the Access Controllers that are defined in the server configuration under the node Access *Controllers*.

Each element in the array is a reference to a real AccessController object and can thus be used as any other AccessController object in the script. All AccessController variables are available.

3.30.1 Methods in the AccessControllers-object

AccessDorr(doorID)

This method is used to manually give access to a door from the script. Note that it is always possible to manually give access to a door from the script.

Example:

```
//Manually give access to the door Door 1
AccessControllers.AccessDoor(AccessController_1.Door_1
.ID);
```

3.30.2 Events in the AccessControllers-object

There are two different events that can be caught in the script. Both provide an event object with different properties.

onAccessPointEvent

This event is raised when an event is generated for an Acces Point by an Access Controller.

Example:

```
AccessControllers.onAccessPointEvent = function(event)
{
    Debug.Print(event.DoorName);
}
```

In the example, the name of a door that has an access point that generated an event will be printed.

The ***event*** object provided has the following properties:

DoorControllerName is the name of the door controller to which the door is connected that generated the event.

DoorID is the ID of the door where the access point is located that generated the event.

DoorName is the name of the door where the access point is located that generated the event.

EventType is the type of event generated. Possible values are:

- Access Denied – access was denied.
- Access Granted – access was granted, the door was unlocked.
- Access Not Taken – access was granted but the door was not opened within the access time.
- Access Taken – the door was opened after access was granted.

onDoorStateChanged

This event is raised when an event is generated for a door by an Access Controller.

Example:

```
AccessControllers.onDoorStateChanged = function(event)
{
    Debug.Print(event.DoorName);
}
```

The example will print the name of the door that generated the event in the Access Controller.

The ***event*** object has these properties:

AlarmState is the current alarm status of the door. Possible values are:

- Normal – normal state, no alarm status.
- Forced Open – the door has been opened without having been granted access, it has been forced open.
- Open Too Long – the door has been open too long, longer than the configured max time.

DoorControllerName is the name of the Door Controller to which the door that generated the event belongs.

DoorID is the ID of the door that generated the event.

DoorMode is the current status of the door. Possible values are:

- Accessed – access to the door has been granted.
- Blocked – the door is in a blocked state, access is not allowed.
- Locked – the door is locked.
- Unlocked – the door is (permanently) unlocked.
- Locked Down – the door is locked and blocked in this state.
- Locked Open – the door is unlocked and blocked in this state.
- Double Locked – the door is locked with double lock.

DoorName is the name of the door that generated the event.

PhysicalState is the physical state of the door. Possible values are:

- Open – the door is open.
- Closed – the door is closed.

WarningState is the current warning state of the door. Possible values are:

- Normal – normal state, no warning.
- Open Too Long – the door has been open too long.

4 Ethiris Camera Simulator

4.1 Overview

The purpose of the simulator is to have the opportunity to display Ethiris without needing to connect real cameras. Ethiris Camera Simulator comes pre-configured with 4 video sequences and 4 cameras. The simulator can generate video streams for up to 24 cameras simultaneously, it can also deliver several streams for each camera to Ethiris Server.

In addition to the 4 pre-installed video stream sources, the user can add more stream sources, these can be of several types:

- A single file in the form of exported video sequences from Ethiris Client in Ethiris export format - *.ethe.
- A folder with several *.ethe files.
- A single JPEG image file.
- A folder containing several JPEG files.

Each of these are best produced by exporting from Ethiris Client.

If you use a folder of several files they will be played one after the other in alphabetic order sorted by file name.

The video sources are played repeatedly until you stop the simulator or one of the cameras.

To Ethiris Server the simulator behaves like a 24 channel video server, after installation only the channels 1, 2, 3 and 4 are enabled.

The simulator listens to port *1236* for connections by default.

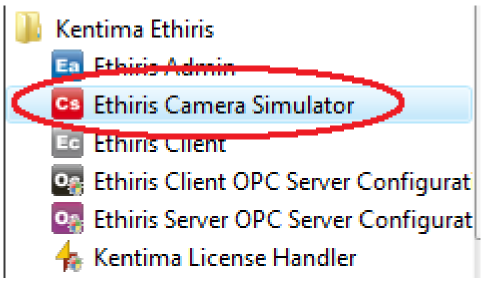
4.2 Starting the Simulator

When you have installed the simulator, it is extremely easy to start it so that it starts to send frames from the simulated cameras just as if they were real cameras.

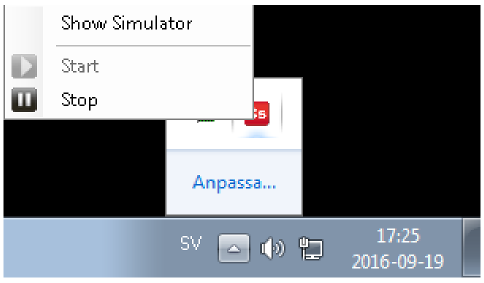
Click the *Start menu* and **select** *Programs*.

Select the program group *Kentima Ethiris*.

1. **Click** the program icon *Ethiris Camera Simulator*.

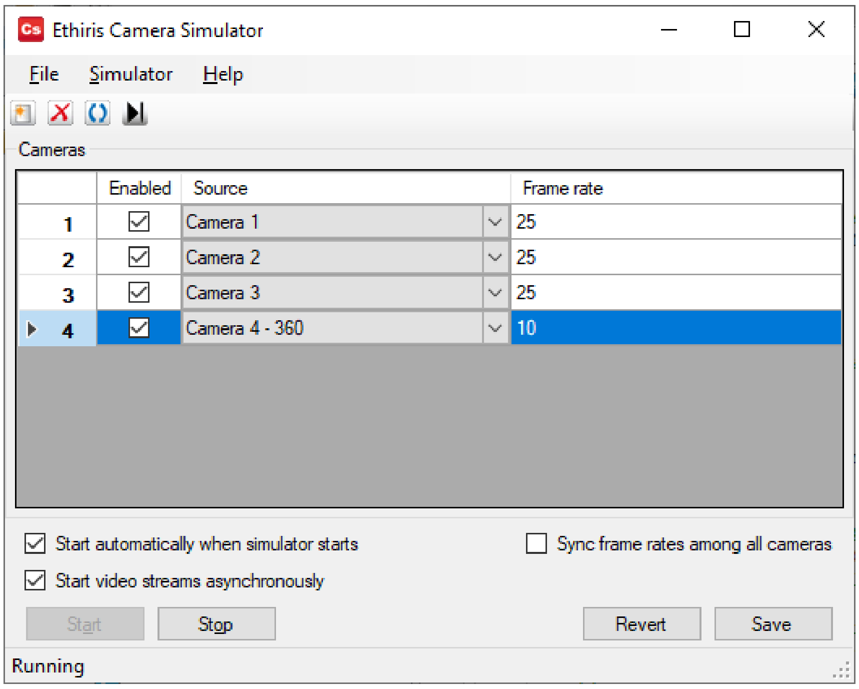


2. The camera simulator will start hidden in the system tray. **Click** on the system tray icon on the taskbar and then **right-click** on the *Ethiris Camera Simulator* icon to display its' popup menu.



3. **Click** the menu item *Show Simulator* to display the simulator main window.

4.3 The simulator interface



	Enabled	Source	Frame rate
1	✓	Camera 1	25
2	✓	Camera 2	25
3	✓	Camera 3	25
4	✓	Camera 4 - 360	10

[✓] Start automatically when simulator starts
[ ] Sync frame rates among all cameras
[✓] Start video streams asynchronously

Running

Figure 4.1 Ethiris camera simulator.

At the top of the main window is a menu bar with three alternatives. The *File* menu has the following choices:

- **Revert** – discards all changes made since the last *Save*.
- **Save** – saves all current settings so they will be restored when the simulator is started the next time.
- **Hide to Tray** – closes the main window but keeps the simulator running. It can be reached from the system tray.
- **Exit** – will close all video streams and terminate the simulator.

The *Simulator* menu has the following choices:

- **Configure Stream Sources…** - will open a new window where you can add new video sources in the form of an exported video from Ethiris Client in *.ethe files or JPEG images. See 4.4

Configure Stream Sources described below.

- **Add Camera** – creates one additional camera in the list. Up to 24 cameras can be added.
- **Remove Camera** – removes the highest-numbered camera from the list.
- **Start** – starts the simulator, this will make all cameras with a checkbox in the column *Enabled* available for streaming to Ethiris Server.
- **Stop** – will stop the simulator, all cameras will become unavailable for Ethiris Server.

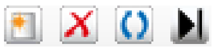


Figure 4.2 Simulator Toolbar.

Add Camera.	Creates one additional camera in the list. Up to 24 cameras can be added.
Remove Camera	Removes the highest-numbered camera from the list.
Restart enabled video streams asynchronously	Restarts all active cameras in order, except for the first one.
Next Frame	Cameras with a Frame rate set to 0, will be stepped forward one frame with each press. Waiting for too long between two frames will result in a communication error with the camera.

The main window shows a list of the cameras that are defined in the simulator. The list has columns where you can modify the settings of the simulated cameras.

- **Enabled** can be used to individually control whether each camera is active or not.
- **Source** is a combo box where you can select which of the defined video sources should be used for each camera.
- **Frame rate** specifies the number of frames per second to send to Ethiris Server for each camera. This is regardless of the frame rate that was used by the camera when the stream source was recorded.

  For the video from the simulated camera to look normal when displayed, you should use the same setting as the camera when recorded, however, for testing purposes, you can adjust it freely.

Any changes take effect immediately, however, they are only temporary until you click the *Save* button.

If you have several cameras defined and you want to change one setting to the same value for all cameras, there is a quick way. If you double click on the column header for one of the columns, all cameras will get the same value for that column as the camera that is select.

For example, the frame rate for camera 3 is set to 25 frames per second. If you select camera 3 by clicking somewhere on row 3 and then double click the "*Frame rate*" column header, all cameras will be set to 25 frames per second.

At the bottom of the main window there are some push buttons:

- **Start** – starts the simulator, this will make all cameras with a checkbox in the column *Enabled* available for streaming to Ethiris Server.
- **Stop** – will stop the simulator, all cameras will become unavailable for Ethiris Server.
- **Revert** – will remove all changes made after the last time you pushed the *Save* button. This includes changes made in the *Configure Stream Sources* dialog.
- **Save** – will persist all settings so they remain when the simulator is started the next time. This includes changes made in the *Configure Stream Sources* dialog.

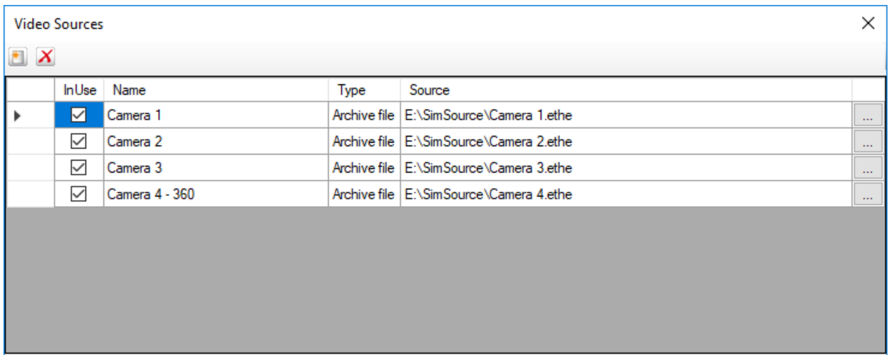
There are also three options in the form of checkboxes.

- **Start automatically when simulator starts** – Is per default enabled so that the simulator starts the streaming of video upon startup.
- **Start video streams asynchronously** – Is per default **not** enabled. Its function is to start the streaming of video with a random delay, between 250 and 500ms, so that cameras using the same source are not shown synchronously.
- **Sync frame rates among all cameras** – Is per default **not** enabled. When set, the marked channels Frame rate is set to all channels. And any change to Frame rage, regardless of channel, will be synchronized among all channels.

The status bar indicates that the simulator is started by showing the text ”*Running*”, otherwise the text ”*Stopped*” is shown.

4.4 Configure Stream Sources

The menu alternative *Simulator->Configure Stream Sources…* opens a dialog where you can specify the stream sources that should be available for selection in the main window. The name of the items in this list will automatically populate the stream source selection list in the main window.



	InUse	Name	Type	Source
▸	✓	Camera 1	Archive file	E:\SimSource\Camera 1.ethe
	✓	Camera 2	Archive file	E:\SimSource\Camera 2.ethe
	✓	Camera 3	Archive file	E:\SimSource\Camera 3.ethe
	✓	Camera 4 - 360	Archive file	E:\SimSource\Camera 4.ethe

Figure 4.3 The Configure Stream Sources dialog.

There are two buttons at the top:

Creates a new stream source in the list.

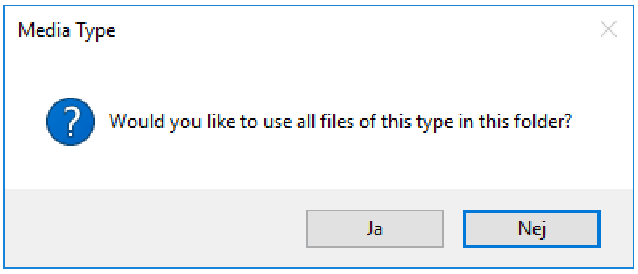
Deletes the stream source that is selected in the list. If any camera is using this stream source, that cameras source will become the first source.

The list of stream sources has these columns:

- **In use** – specifies if the stream source should be available for selection as a source for a camera.
- **Name** –the name to display in the stream source selection box for the cameras. This is automatically set to the name of the source file, and can then be edited.
- **Type** – identifies the type of storage used for the source, this can be one of:
    - **None** - an empty source, you need to select a source file/folder. Will produce no video.
    - **Archive file** – a file exported from Ethiris Client in Ethiris native format (*.ethe). This is the recommended format to use.
    - **Archive folder** – a folder containing several archive files. The files will be used sequentially.
    - **Jpeg file** – a single JPEG image file, this will produce a still image.
    - **Jpeg folder** – a folder containing a number of JPEG files that are used sequentially, sorted alphabetically by name.

  The value for this column can not be set manually, it's selected automatically when you select a file or folder in the *Source* column.
- **Source –** The file or folder to use as a stream source. You should not write the file/folder name here but use the browse button ... at the end of the line to browse for the file/folder.

Note – you must always select a single file, even if you want to use several files in a folder.


Media Type

Would you like to use all files of this type in this folder?

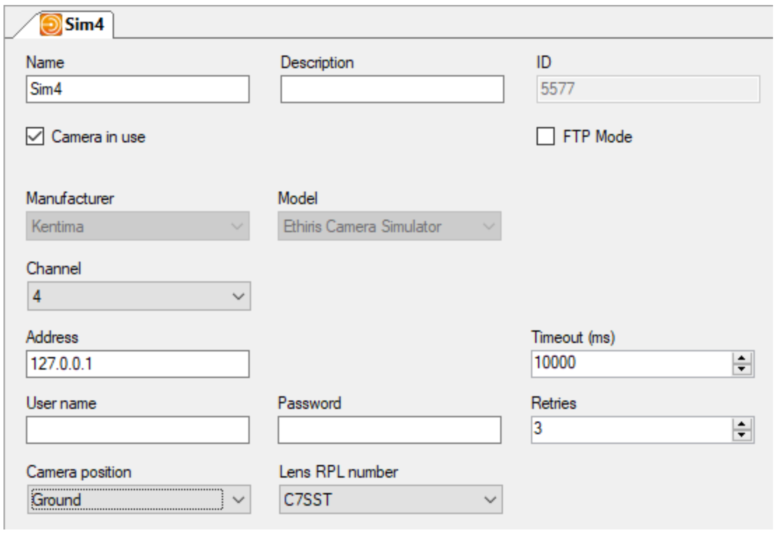
When you have selected a file, the simulator will display a dialog that asks if you want to use only the selected file or all files of the same type in the folder.

4.5 Simulator Camera in Ethiris

1. **Start** *Ethiris Admin*.
2. **Add** a new camera. For information on how to do this, see

Cameras node on page 2:72.

3. **Name** the camera *Sim1*.
4. **Select** *Kentima* as the Manufacturer.
5. **Select** *Ethiris Camera Simulator* as the Model.
6. **Enter** 127.0.0.1 (or *localhost)* as the address for connecting to the simulator that has just started on your own computer.
7. The dialog should look as in Figure 4.4 when you have finished. **Click** the Save button to save the new camera.


Sim4

Name: Sim4	Description:	ID: 5577
☑ Camera in use		☐ FTP Mode
Manufacturer: Kentima	Model: Ethiris Camera Simulator	
Channel: 4		
Address: 127.0.0.1		Timeout (ms): 10000
User name:	Password:	Retries: 3
Camera position: Ground	Lens RPL number: C7SST	

Figure 4.4 Dialog for adding a new camera (simulator).

Note! It is only camera 4 (Sim4) that is panomorph. For the dewarping to work correctly, you have to set the RPL number to C7SST for this camera. The other simulator cameras should be set to *Not panomorph.*

Camera position for camera Sim4 should be set to *Ground,* as the camera was placed on the table when the video was recorded.

Now that the simulator has been started and its cameras have been added to the server configuration, the simulator cameras can be used in an Ethiris client just like any other camera. For more information on how to use cameras in the client, see the *Ethiris Client User´s Guide*.

5 Explanation of Terms

Camera view

A view that can display frames from a camera. The live view can display up to 64 camera views simultaneously.

Client view

A client view can contain between 1 and 64 camera views. Each client view is linked to a function button that is displayed in the live view.

Cluster of Ethiris Servers

A cluster of Ethiris Servers means that one or more different physical machines can act as one unit with redundancy. If a member is disconnected, the others can automatically handle the member´s tasks with a barely noticeable impact on any potential clients. This function requires that every member of the cluster has a Premium level.

COM

Component Object Model. A binary communication standard from Microsoft that can be used in communication between various program components. COM is the basis of OPC.

Command port

In communication via TCP/IP, the recipient of data must be identified with both an IP address and a port number. We call the port number on which an Ethiris server listens for commands from clients the command port. By default, this is 1235. If necessary, the number can be changed in the system configuration for each Ethiris server. A port number can assume values between 1 and 65535.

Continuously recorded frames

A type of recorded frames that are recorded continuously from a camera at a specific time interval between each frame. The video is recorded regardless of whether alarm events occur.

Data store

The data store is located in Ethiris Server and contains all variables and their current state in the system.

Document window

This is a window that is used for presenting various types of information such as properties for various objects.

Floating window

A floating window has been torn away from the main frame and is floating freely. In a floating state, a window can be positioned on another monitor and can be resized as desired.

Function button

A function button in this context is a button in the live view that, when you click it, either select a predefined layout of camera views or starts recording from the camera selected.

FTP

File Transfer Protocol. In Ethiris it is possible for certain camera models to specify that on alarm the images shall be sent by FTP. In that case, images are not sent continuously which is the normal behavior, but images are sent only when an alarm is detected in the camera. An event is created in Ethiris just as it was a normal event triggered recording. In this way, you can conserve a lot of bandwidth in your system.

H.264

A video compression standard built on similar technology as MPEG-4. H.264 is also called MPEG-4 part 10. It is even more compact then MPEG-4, about half the size for a comparable quality. But on the other hand, it is very demanding for the hardware, especially when it comes to high-resolution frames.

Hotspot

A type of camera view that displays frames from the camera selected and/or the current alarm camera.

I/O

Input / Output. Exist both as digital and analog signals. Several camera models have a number of digital inputs and outputs that can be used to connect extern equipment to the system.

IP address

Each unit in a computer network has a unique IP address that consists of 4 groups of digits. Each group can be 0-255, for example, 192.168.30.29.

License key

A key that is received from Kentima Solutions after payment has been made and a registration key has been submitted. The license key is used to ”unlock” Ethiris via the license dialog.

Live frames

Video from a camera that is being filmed right now, as opposed to recorded frames.

Main frame

The main frame is the area in Ethiris Admin available for various windows such as the *Ethiris Explorer* and document windows.

MJPEG

Motion JPEG, a video compression format where each frame is a separate JPEG image.

MPEG-4

A video compression format where a complete image (i e I-Frame) is followed by a number of smaller images depicting the difference compared to the last image (P-Frames or B-Frames). This is a more compact format compared to MJPEG.

Network camera

A type of camera that can be directly connected to a network. The camera device has a unique IP address.

OPC

OLE for Process Control. A communication standard developed within the automation industry for communication between different systems.

Pin

A tool window can be pinned which means that it will not slide into the edge when it loses focus, but it stays open.

Preset position

For a camera with a PTZ function, you can define preset positions in Ethiris. A preset position is a named position with preset coordinates for pan, tilt, and zoom. A preset position can be activated manually in Ethiris client by an operator, but also automatically through a script in the Ethiris server. PTZ tours are built up by a list of preset positions.

Product code

A key following each Ethiris license. The key shall be entered at installation time. Both the Ethiris Client and Ethiris Server requires a product code at installation.

Protocol

The selected protocol specifies what method to use when Ethiris requests images from a camera. Today the choices are MJPEG, MPEG-4, and H.264.

PTZ

Pan, Tilt & Zoom. The standard designation for cameras with these functions.

Receive port

When communicating over RTP the images are sent from the camera to a specific port number on the computer where Ethiris server runs. If you don't specifically set a port number Ethiris will randomly select an available port number. In some cases, e.g. if you have to open the port in a firewall, you want to specify this port number.

Registration key

The key that uniquely identifies an instance of a program. The key is generated by clicking the *Register* button in the license dialog. This key must be sent to Kentima Solutions AB to unlock Ethiris.

Round

A type of camera view in which a list of cameras is created for automatic switching between cameras at a specific time interval.

RTP

Real-Time Protocol.

Scale mode

Indicates how the frame in a camera view is to be displayed. The options are: *Keep Aspect Ratio*, *Fill Display area* and *Original size*. The default scale mode is *Keep Aspect Ratio* where the image is as large as possible without distorting the proportions (aspect ratio).

Section

A part of the system with its own set of view buttons and function buttons. Each section is represented by an item in *Sections Explorer*. A *Section* can consist of other sections making it possible to build hierarchies.

Tab group

Both *Document windows* and *Tool windows* can be grouped together and become tabs in a group. In this state, only one window at a time in the group can be visible. Click a tab to make the corresponding window visible.

TCP/IP

Transmission Control Protocol/Internet Protocol.

Test license

When Ethiris is installed, a test license is received with which you can test the product free of charge.

Tool window

A Tool window is used for supplying the user with various tools such as a list of available variables. In Ethiris Admin there is one Tool window. This is Ethiris Explorer. There are also tool windows in some panels like the *Variable Browser* in the *Script* panel.

Trigger level

The level of motion required in a frame for it to be interpreted as a motion alarm. Indicated in % of pixels with motion in relation to all pixels monitored in the frame.

Unpin

When you *unpin* a tool window, it will slide into the edge of the main frame when it loses focus, leaving visible only a tag with the name of the window.

User interface

The elements in a computer program with which a user works when using the program. What is visible on the screen such as forms, dialogs, etc.

User operation

There are 30 various user operations for which you can require authorization in Ethiris. Examples of user operations are ”View live video from camera” and ”Update server configuration”.

Video encoder

A device that can be directly connected to a computer network. Its purpose is to convert one or more video signals to a format that can be sent on the network, for example, Jpeg.

Video request port

When communicating over RTP the camera might require video requests on a port different from the normal communication port over TCP. In that case, you can specify the correct video request port. The default value for this port is automatically selected by Ethiris based on the selected camera model.

Viewing area

The area in the live view that is used to display frames from different cameras, i.e. excluding menu and function buttons.

6 Index

KENTIMA PRODUCT LINES

SECURITY

VIDEO MANAGEMENT SOFTWARE

NETWORK VIDEO RECORDER

PSIM SOFTWARE

AUTOMATION

HMI/SCADA SOFTWARE

INDUSTRIAL COMPUTERS

OPERATOR PANELS

Mailing address:
PO BOX 174
SE-245 22 Staffanstorp
Sweden

Visiting address:
Kastanjevägen 2
245 44 Staffanstorp
Sweden

Phone: +46 (0)46-25 30 40
Fax: +46 (0)46-25 03 10

E-mail: info@kentima.com
www.kentima.com

KENTIMA
Automation and Security Products