

ETHIRIS®

VIDEO MANAGEMENT SOFTWARE

Integration med Ethiris ▶

Camera 4

Camera 5

Camera



Camera



Camera 11

Camera

Copyright © 2020 Kentima AB

Mångfaldigande av innehållet i denna manual, helt eller delvis, är enligt lagen om upphovsrätt till litterära och konstnärliga verk förbjudet utan medgivande av copyright-innehavarna. Förbudet gäller varje form av mångfaldigande genom tryckning, kopiering, bandinspelning, överföring till elektroniskt media etc.

Produktion och layout: Kentima AB

Version: 12.0

Första utgåva: November 2020

Tryck: Kentima AB

Varumärken: **Ethisis** och **WideQuick** är registrerade varumärken. Alla andra märken tillhör sina respektive ägare.

Kentima AB

Postadress:

Box 174

245 22 STAFFANSTORP

Sverige

Besöksadress:

Kastanjevägen 4

245 44 STAFFANSTORP

Sverige

E-post:

info@kentima.com

Internet:

www.kentima.com

Innehåll

1	Introduktion	1:1
1.1	Introduktion	1:1
1.1.1	Användning	1:1
1.2	Översiktlig beskrivning	1:1
2	ActiveX	2:1
2.1	Ethis ActiveX control	2:1
2.1.1	Översikt	2:1
2.1.2	Installation	2:2
2.1.3	Testkör exempelapplikationen	2:3
2.1.4	ActiveX control gränssnitt	2:7
3	Fjärrstyrning av Ethis Client	3:1
3.1	Ethis Client Remote control	3:1
3.1.1	Översikt	3:1
3.1.2	Installation	3:1
3.1.3	Prova exempelapplikationen	3:2
3.1.4	Fjärrkontrollens gränssnitt	3:9
4	Modbus OPC Server	4:1
4.1	Modbus OPC Server	4:1
4.1.1	Översikt	4:1
4.1.2	Installation	4:1
4.1.3	Konfiguration	4:1
5	Mottagare	5:1
5.1	Mottagare	5:1
5.1.1	Översikt	5:1
5.1.2	Mottagare allmänt	5:2
5.2	Mottagartyper	5:3
5.2.1	TCP-inkommande	5:3
5.2.2	TCP-utgående	5:4
6	Förklaring av termer	6:1
7	Index	7:1

1 Introduktion	1:1
1.1 Introduktion	1:1
1.1.1 Användning.....	1:1
1.2 Översiktlig beskrivning.....	1:1

1 Introduktion

1.1 Introduktion

Denna manual har som syfte att ge användaren hjälp med att integrera sitt Ethisis system med andra sorters system.

Det finns totalt sex olika manualer för Ethisis. Förutom denna finns även *Installera Ethisis*, *Admin – Konfiguration för Ethisis*, *Ethisis Client användarguide*, *Kom igång med Ethisis* och *Kom igång med Ethisis Mobile*.

Denna del av manualen beskriver några exempel på hur man integrerar Ethisis med andra system. För information angående installation av Ethisis, se *Installera Ethisis* manualen. För information angående att komma igång med sitt Ethisis system efter installation, se *Kom igång med Ethisis*.

För djup information om de olika funktionerna i Ethisis, *Admin – Konfiguration för Ethisis* samt *Ethisis Client användarguide* manualerna.

1.1.1 Användning

Huvudsyftet med Ethisis är kameraövervakning som utförs på två sätt. Ena sättet är att manuellt övervaka livebilder från olika kameror. Det andra sättet sker genom inspelning av bilder från anslutna kameror. Inspelning kan ske kontinuerligt från en eller flera kameror eller i form av bildsekvenser när ett larmvillkor är uppfyllt. De inspelade bilderna kan spelas upp i efterhand m h a ett videobandspelarliknande gränssnitt.

1.2 Översiktlig beskrivning

Ethisis är ett system för övervakning som använder sig av nätverkskameror och analoga kameror tillsammans med videoencodrar från olika leverantörer.

Produkten är indelad i flera program, där *Ethisis Server* och *Ethisis Client* är de viktigaste. Serverdelen hanterar alla kameror och sparar bilder på hårddisken. Klientdelen visar livebilder och inspelade bildsekvenser.

Det finns också ett program kallat *Ethisis Admin*, vilket används för att konfigurera alla Ethisis-moduler i systemet. I *Ethisis Admin* konfigurerar man *Ethisis Server* genom att t ex definiera vilka kameror som är anslutna till varje *Ethisis Server*, när video ska spelas in, vilken bildfrekvens och upplösning man ska använda, etc. Man definierar också användargränssnittet för de olika *Ethisis Clients* i systemet genom *Ethisis Admin*.

Teoretiskt sett kan ett oändligt antal kameror anslutas till varje Ethisis server, samt visas i önskat antal Ethisisklienter. Dock sätter bandbredd och skärmupplösning gränser för hur många anslutna kameror som är lämpliga.

Det finns olika licensnivåer för Ethis som tillåter olika många anslutna kameror. För att tillgodose behovet av stora anläggningar med många kameror, är Ethis designat med fokus på skalbarhet. Med skalbarhet menas att det är möjligt att dela upp sitt system på flera Ethisservrar och på så sätt sprida arbetet på flera datorer.

En sammanfattning av de olika Ethiskomponenterna ses nedan:



Ethis Server

Ethis Server är kärnan i ett Ethissystem. Den hanterar all kommunikation med kamerorna och spelar in video på hårddisk. *Ethis Server* kör som en tjänst under operativsystemet och startas automatiskt när datorn startas. Ett Ethissystem kan omfatta en eller flera Ethisservrar.



Ethis Client

Ethis Client används för att se på livevideo och inspelad video. En Ethisklient kan anslutas till en eller flera Ethisservrar för att få tillgång till kameror. Ett Ethissystem kan ha en eller flera Ethisklienter.



Ethis Admin

Ethis Admin används för att konfigurera de olika delarna i ett Ethissystem. *Ethis Server* och *Ethis Client* konfigureras med detta gemensamma verktyg. Från vilken dator som helst i systemet, kan *Ethis Admin* köras och användas för att konfigurera alla Ethiskomponenter på plats.



Ethis Mobile

Ethis Mobile är en app för att kunna koppla upp dig mot dina Ethissystem via mobilen. Via *Ethis Mobile* kan du titta på livebilder från olika kameror i anläggningen samt se larm i systemet.



Ethis Server OPC Server

Ethis Server OPC Server är en separat Ethiskomponent som används för att låta andra system få tillgång till information i *Ethis Server*. Andra system med en OPC-klient kan ansluta till en eller flera Ethisservrar och läsa/skriva till alla signaler i *Ethis Server*, t ex starta en inspelning eller styra en PTZ-kamera.



Ethis ActiveX

Ethis ActiveX är en komponent som används för att visa livevideo från en kamera som är ansluten till en Ethisserver. Den här komponenten kan användas i vilket system som helst som kan hantera standard ActiveX-komponenter.



Ethis Viewer

Ethis Viewer är ett separat program som används för att se exporterad video från ett Ethissystem.



Ethis Signature Validator

Ethis Signature Validator är en separat programvara som används för att verifiera äktheten i exporterade videofiler och jpg-bilder från Ethis.

2 ActiveX

2:1

2.1 Ethisis ActiveX control.....	2:1
2.1.1 Översikt	2:1
2.1.2 Installation.....	2:2
2.1.3 Testkör exempelapplikationen	2:3
2.1.4 ActiveX control gränssnitt	2:7

2 ActiveX

2.1 Ethisis ActiveX control

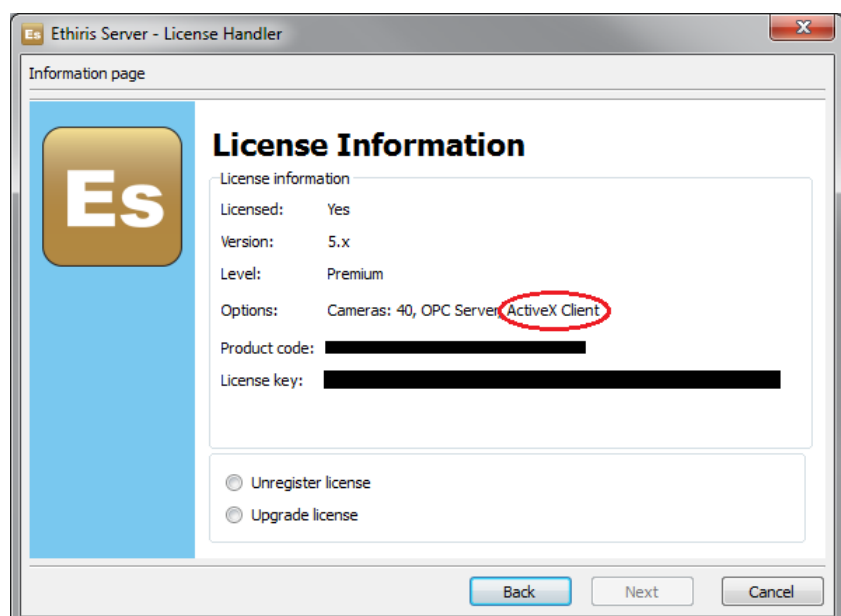
2.1.1 Översikt

Syftet med Ethisis ActiveX control är att låta andra system visa livevideo från kameror som är anslutna till en Ethisis Server. Vilket system som helst som kan hantera ActiveX-kontroller kan använda Ethisis ActiveX control.

Det andra systemet kan köra lokalt på samma dator som Ethisis Server eller så kan det köra på en annan dator. Det viktiga att komma ihåg är att Ethisis ActiveX control ska installeras på samma dator som systemet som ska använda ActiveX-kontrollen kör.

För att kunna ansluta till en Ethisis Server via ActiveX-kontrollen måste Ethisis Servern ha licensoptionen *ActiveX*. Denna är automatiskt inkluderad i en licens på nivå *Premium* och kan köpas till för nivåerna *Extended* och *Advanced*. ActiveX är inte tillgänglig för en Ethisis Server som kör på licensnivå *Basic*.

För att kontrollera om du har optionen *ActiveX*, starta *Kentima License Handler*, välj *Ethisis – Server* i produktlistan och klicka *Next*. I dialogen *License Information* finns en rad med *Options* för denna licens.



Figur 2.1 Se till att licensen inkluderar optionen ActiveX Client

Klicka *Cancel* när du är klar.

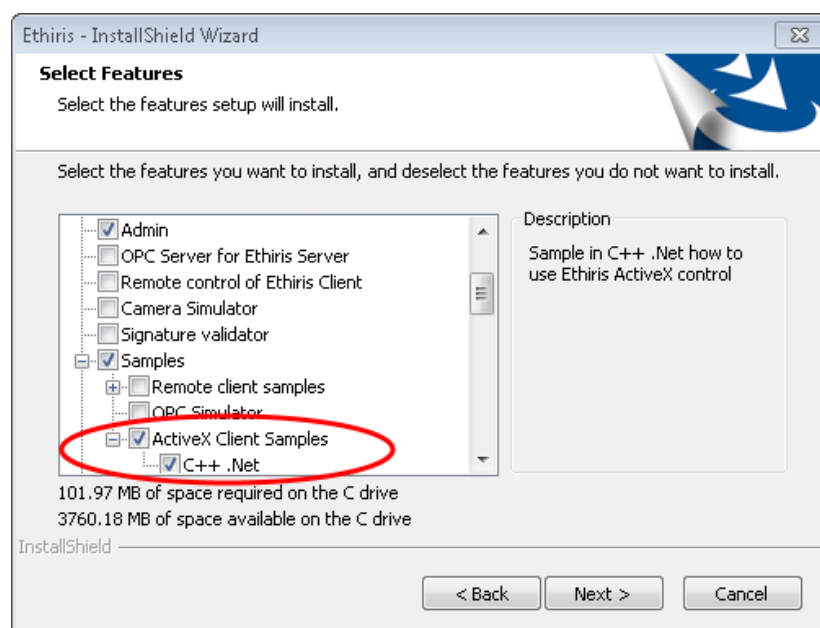
I Ethis setup finns en specifik *Setup Type* kallad *ActiveX Client* som kan väljas för att endast installera ActiveX-kontrollen. Om du vill installera andra Ethiskomponenter tillsammans med ActiveX-kontrollen måste du välja *Setup Type Custom* och sedan välja komponenten *ActiveX Client* tillsammans med övriga önskade komponenter.

Det finns också ett exempelprojekt i C++ .Net som visar hur man kan använda ActiveX-kontrollen.

2.1.2 Installation

Kör Ethis setup på den dator där du vill ha ActiveX-kontrollen. I följande beskrivning av exemplet förutsätter vi att du även installerar exemplet *ActiveX client sample/C++.Net* och att Visual Studio 2017 är installerat på datorn om du vill titta på källkodsfilerna.

ActiveX-kontrollen kommer så småningom att ansluta sig till en Ethis Server någonstans i nätverket. Ethis Server behöver inte vara installerad på samma dator som ActiveX-kontrollen är, men det fungerar lika bra lokalt, med både ActiveX-kontrollen och Ethis Server på samma dator.



Figur 2.2 Välj komponenterna *ActiveX Client Sample/C++.Net* & *ActiveX Client*.

ActiveX Client 32-bit (själva ActiveX-kontrollen) väljs automatiskt när exemplet väljs.

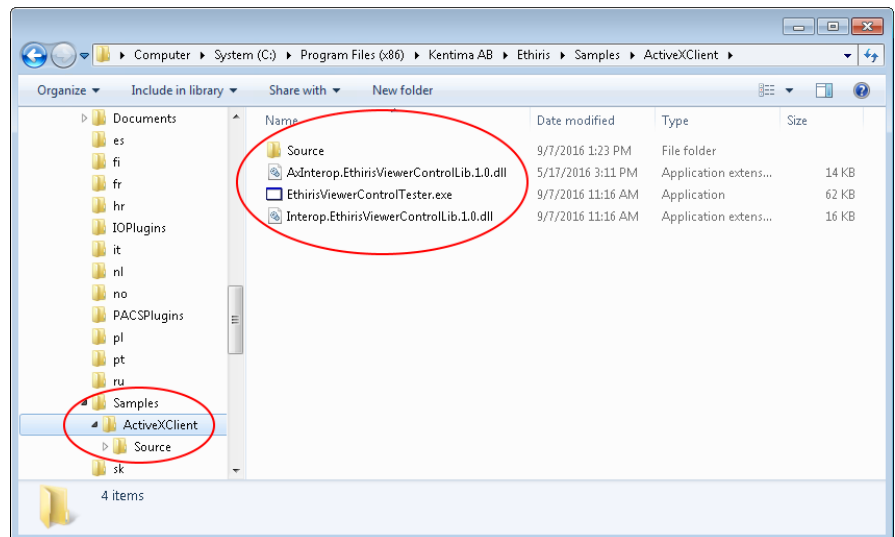
2.1.3 Testkör exempelapplikationen



Exempel-
applikationen är
inte avsedd att
direkt användas i
en riktig tillämpning!

Efter installation av exemplet hittar du exempelfilerna i en underkatalog till Ethisis installationskatalog. I allmänhet är denna `C:\Program Files (x86)\Kentima AB\Ethisis\Samples\ActiveXClient`.

Observera att exempelapplikationen bara är till för att visa hur man kan använda *ActiveX Client*. Den är inte avsedd att direkt användas i en riktig tillämpning!

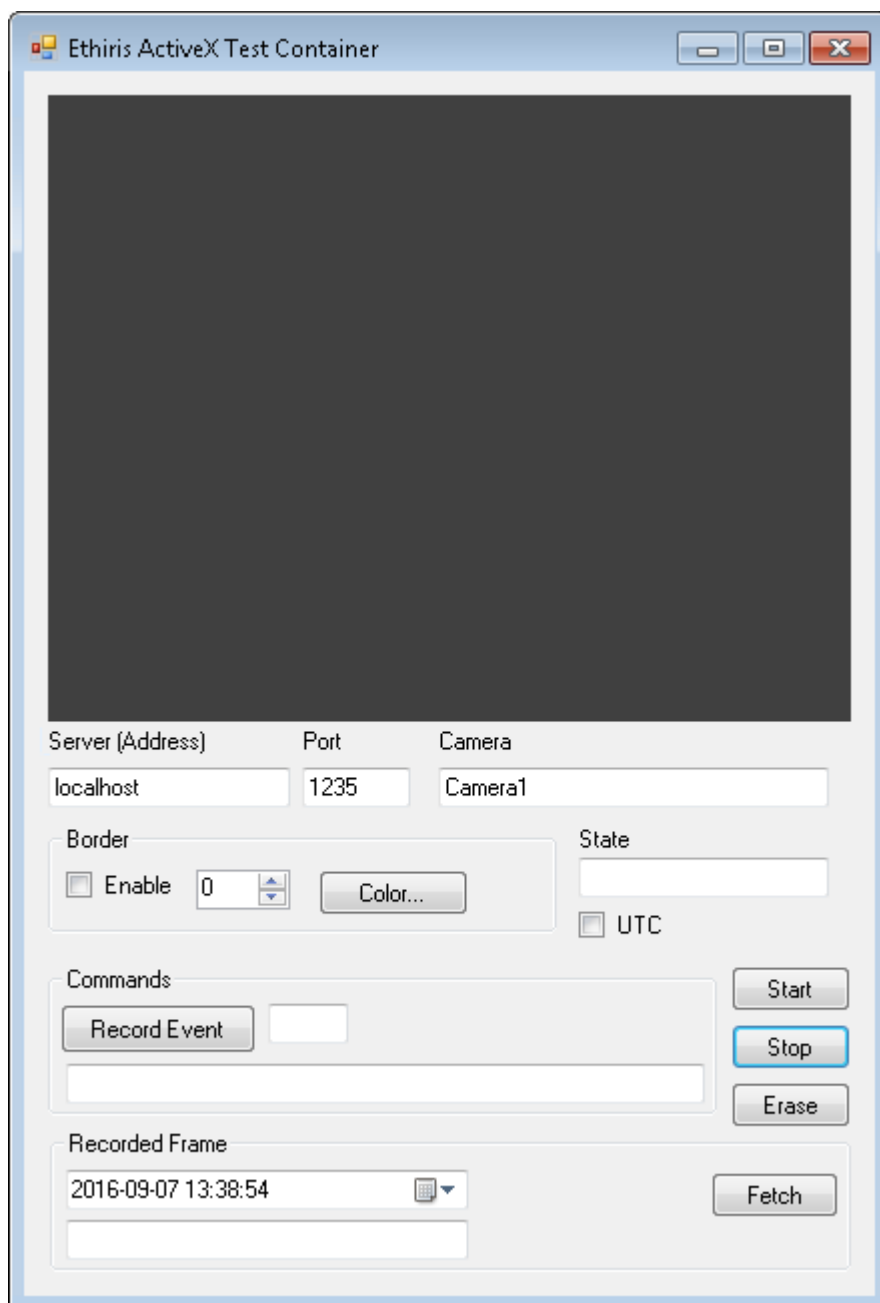


Figur 2.3 ActiveX exemplet i Windows Explorer.

Källkodsfilerna ligger i underkatalogen *Source*. Jag förutsätter att om du känner till Visual Studio 2017, så vet du hur källkodsfilerna ska användas. Förutom källkodsfilerna finns en kompilerad version av exemplet kallad *EthisisViewerControlTester.exe*.

Låt oss testa exemplet!

Dubbelklicka på *EthisisViewerControlTester.exe* filen i Windows Explorer. Applikationen startar och ser ut som i *Figur 2.4*.



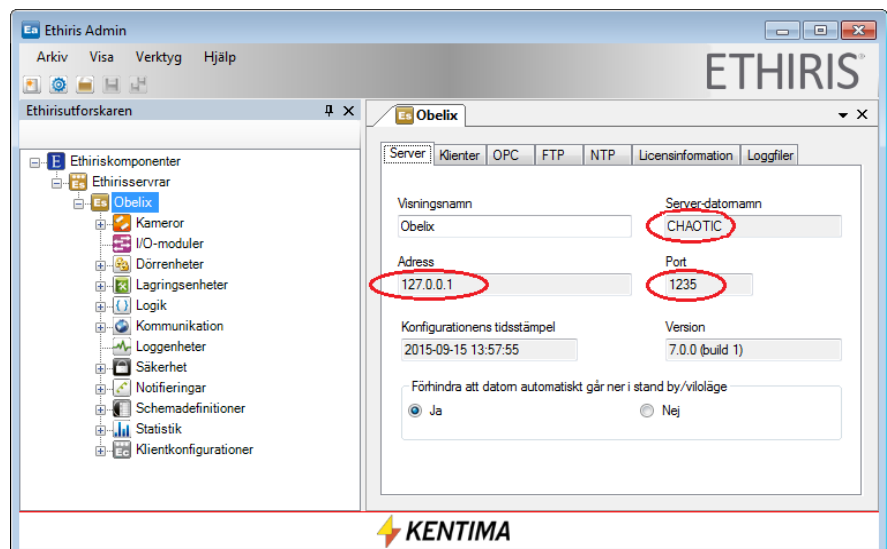
Figur 2.4 Exempelapplikationen ActiveX har just startats.

Överst i formuläret är ActiveX-kontrollen, i viloläge för tillfället.

Tanken är att vi ska ange namnet/IP-adressen till en Ethisis Server och namnet på en kamera. Då kan vi starta/stoppa livevideo och skicka kommandon till ActiveX-kontrollen.

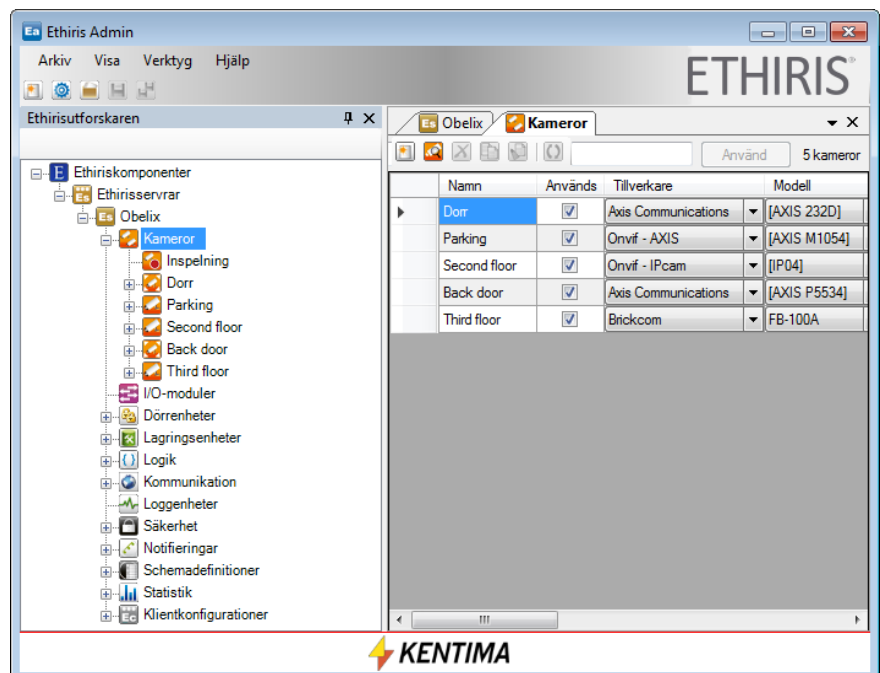
Om du har anslutit en Ethisis Server i Ethisis Admin, kan du titta där för att få information om namnet/IP-adressen till Ethisis servern och namnet på de anslutna kamerorna.

I Figur 2.5 är panelen Ethisis Server öppnad genom att ha dubbelklickat på noden Ethisis Server (inringad) i trädvyn. I panelen till höger är *Serverdatornamn*, *Adress* och *Port* också inringade.



Figur 2.5 Ethis Server konfiguration i Ethis Admin.

Nu, dubbelklick på noden *Kameror* i trädvyn öppnar tillhörande panel och visar en lista med de anslutna kamerorna.



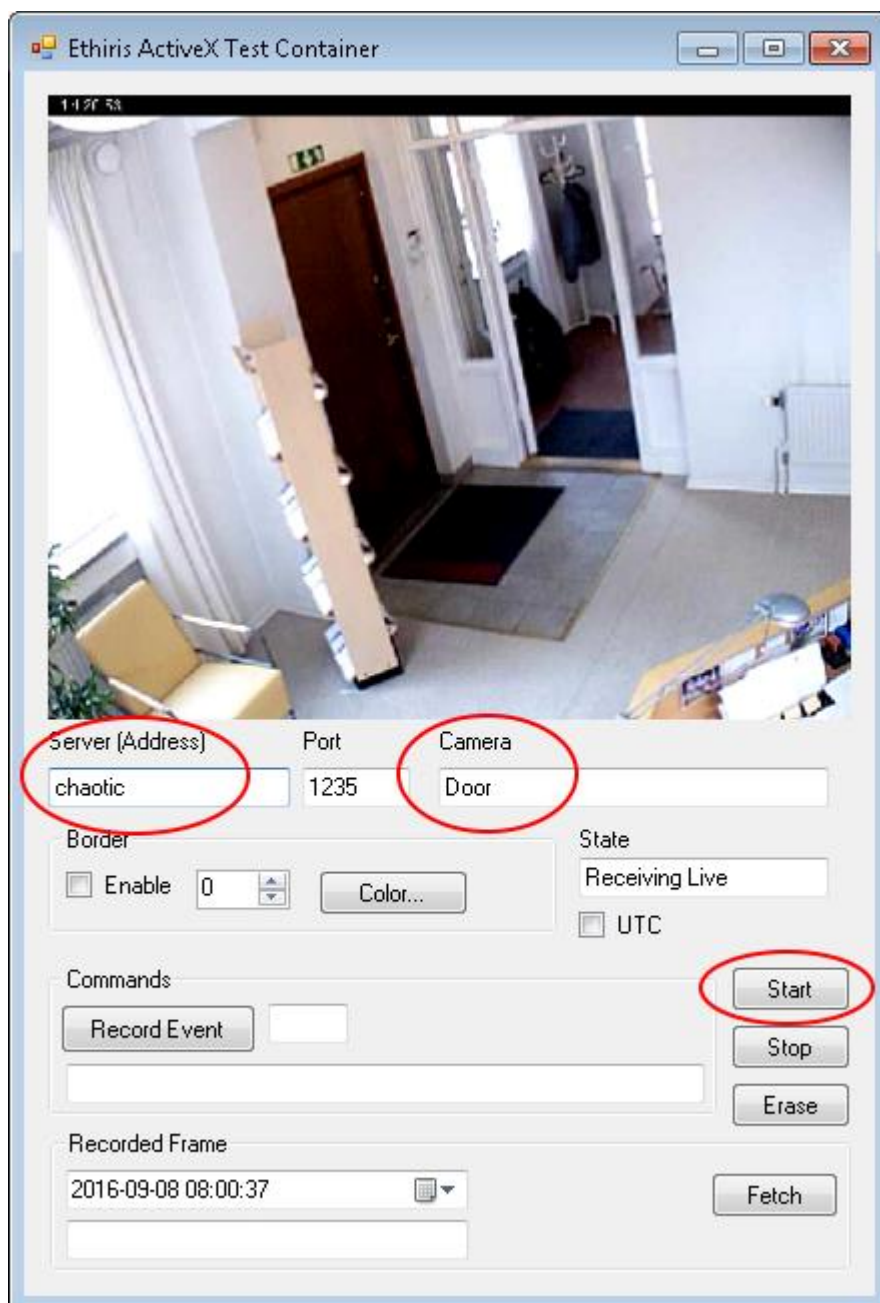
Figur 2.6 Listan med nätverkskameror i Ethis Admin.

Då har vi tillräckligt med information för att testa ActiveX exemplet.

Ange lämpligt *namn*/*IP-adress* till en Ethis Server och ett *kameranamn* i under *Camera* i exempelapplikationen. I mitt fall anger jag *chaotic* som namn för Ethis Server och *Door* som kameranamn.

Klicka på knappen *Start* för att starta livevideo.

Om Ethis Server kör på samma dator, kan du låta *localhost* stå kvar eller ange *127.0.0.1* som namn för Ethis Server eftersom båda dessa är alias för den lokala datorn.



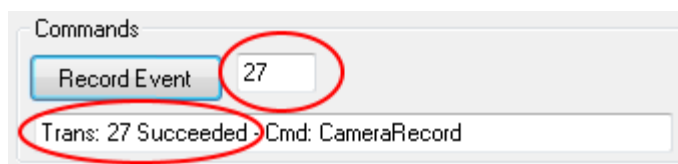
Figur 2.7 Livevideo i testapplikationen.

Klicka bara på knappen *Stop* för att stoppa livevideo.

I sektionen *Commands* kan du starta händelseinspelning för en kamera. Klicka på knappen *Record Event* (*Händelsestyrd inspelning*). Angiven kamera kommer starta händelseinspelning.

Till höger om knappen *Event Record* finns ett fält som visar *Transaction ID* (*TransaktionsID*) för senaste kommandot. I fältet meddelandefältet undertill, visas information om det senaste kommandot, såsom *TransaktionsID*, *Status* och *Kommando*.

För tillfället finns endast ett tillgängligt kommando, *RecordEvent*. I framtiden kan ytterligare kommandon bli tillgängliga och då får transaktionsid en större betydelse för att kunna identifiera olika kommandon.



Figur 2.8 Avsnittet Commands (Kommandon) från testapplikationen.

I sektionen, *Border (Kant)*, kan du välja att visa en kant för ActiveX-kontrollen och bestämma *Border Color (Kantfärg)* och *Border Width (Kantbredd)*.

2.1.4 ActiveX control gränssnitt

När du använder Ethis ActiveX control så använder du gränssnittet till kontrollen. Följande är en referens till de olika tillgängliga egenskaperna, metoderna, händelserna och uppräkningsarna. Dessa egenskaper, metoder och händelser är Ethis specifika. Utöver dessa finns också en mängd standardegenskaper, metoder och händelser för en ActiveX-kontroll, såsom *Width (Bredd)* och *Height (Höjd)*.

Egenskaper

Namn	Typ	Beskrivning
BorderColor	OLE_COLOR	Bestämmer färgen på kanten (om den är synlig).
BorderWidth	Long	Bestämmer tjockleken på kanten (om den är synlig).
BorderVisible	Boolean	Bestämmer om kanten är synlig eller inte.
Camera	String	Namnet på en förvald kamera i Ethis Server.
ErrorDescription	String	Feltext (om i felläge).
LogItems	Array	Reserverad för framtida användning
Port	Long	Förvald port i Ethis Server att ansluta till. Denna är nästan alltid 1235.
Server	String	Adress till en förvald Ethis Server att ansluta till(IP-adress eller DNS-namn).
VideoState	eVideoState	Aktuell status för videokommunikationen med Ethis Server. Se <i>Uppräkningar</i> senare i detta dokument.
UTC	Boolean	Bestämmer om tidsstämplar tolkas som UTC-tid (sant) eller lokal tid (falsk).

Metoder

Erase() as Long

Rensar bildytan på kontrollen.

RecordEvent([Camera as String], [Server as String], [Port as Long]) as Long

Startar en händelseinspelning för kameran *Camera* på Ethis Servern *Server*. Parameterna är valfria, men om de utelämnas måste egenskaperna *Camera*, *Server* och *Port* sättas. Returvärdet är *TransactionID* för kommandot.

RecordedFrame(Timestamp as String, [Margin as long], [Camera as String], [Server as String], [Port as Long]) as Long

Visar en inspelad videobild nära den specificerade tidpunkten, inom marginalen i ms (+/-) från kameran *Camera* på Ethis Servern *Server*. Parameterna *Margin*, *Camera*, *Server* och *Port* är valfria, men om de utelämnas måste egenskaperna *Camera*, *Server* och *Port* sättas. Standardvärdet för *Margin* är 1000 ms. Returvärdet är *TransactionID* för kommandot. Parametern *Timestamp* skall anges i strängformatet "YYYY-MM-DD hh:mm:ss[.fff]".

StartLive([Camera as String], [Server as String], [Port as Long])

Startar hämtning av livevideo från kameran *Camera* på Ethis Servern *Server*. Parameterna är valfria, men om de utelämnas måste egenskaperna *Camera*, *Server* och *Port* sättas. Resultatet rapporteras tillbaka via händelsen *VideoStateChanged*.

Stop()

Stoppas hämtning av livevideo från aktuell kamera. Resultatet rapporteras tillbaka via händelsen *VideoStateChanged*.

Händelser

CommandComplete(Success as Boolean, TransactionID as Long, Command as String, Message as String)

Denna händelse skickas från ActiveX-kontrollen när ett kommando som har skickats till den tidigare är klart. Följande parametrar skickas i händelsen:

Success – sant eller falskt. Om sant, så utfördes kommandot framgångsrikt.

TransactionID – Innehåller ID för kommandots transaktion. Detta är samma ID som returneras från metoden *RecordEvent* som beskrivits ovan.

Command – Det interna namnet på kommandot. För tillfället finns endast ett tillgängligt kommando, *CameraRecord*.

Message – Om allt är OK så är denna parameter tom. Den kan innehålla ett meddelande från ActiveX-kontrollen med information om kommandot.

LogValueBoolean (Name as String, Timestamp as String, Value as Boolean)

LogValueDouble (Name as String, Timestamp as String, Value as Double)

LogValueInteger (Name as String, Timestamp as String, Value as Long)

LogValueString (Name as String, Timestamp as String, Value as String)

Dessa händelser är reserverad för framtida ny funktionalitet och används inte i denna version.

RecordedFrameDisplayed (Timestamp as String, Width as long, Height as long)

Denna händelse skickas från ActiveX-kontrollen när en inspelad videobild har visats. Följande parametrar skickas i händelsen:

Timestamp – Tidsstämpel för visad bild i strängformatet "YYYY-MM-DD hh:mm:ss.fff".

Width – Bredd i pixel på visad bild.

Height – Höjd i pixel på visad bild.

VideoStateChanged(NewState as eVideoState)

Denna händelse skickas från ActiveX-kontrollen när det sker en förändring i statusen för livevideo, t ex när mottagning av livevideo från kameran startar eller när livevideo stoppas. Följande parametrar skickas i händelsen:

NewState – Det finns totalt 7 olika status som beskrivs nedan i avsnittet *Uppräkningar*. Parametern *NewState* beskriver den nya statusen för livevideo för ActiveX-kontrollen.

Uppräkningar

eVideoState kan innehålla 7 olika värden från 0 till 6 som beskrivs nedan:

vsIdle – 0. Ingen video hämtas och ActiveX-kontrollen är vilande.

vsConnectingLive – 1. Anslutning till kameran för hämtning av livevideo pågår.

vsConnectedLive – 2. Anslutning för hämtning av livevideo är etablerad.

vsCreatingSessionLive - 3. Anrop är utfört för att skapa en session. Detta förfarande har att göra med det interna Ethirisprotokollet.

vsRequestedLive – 4. ActiveX-kontrollen har skicka begäran om livevideo från en kamera.

vsReceivingLive – 5. Mottagning av livevideo. Detta är den normala statusen när livevideo skickas till ActiveX-kontrollen.

vsError – 6. Någonting är fel.

vsConnectingRecordedFrame – 7. Anslutning till servern för hämtning av inspelad video.

vsConnectedRecordedFrame – 8. Anslutning för hämtning av inspelad video är etablerad.

vsCreatingSessionRecordedFrame – 9. Anrop är utfört för att skapa en session. Detta förfarande har att göra med det interna Ethirisprotokollet.

vsRequestedRecordedFrame – 10. ActiveX-kontrollen har skickat begäran om inspelad video från en kamera.

vsReceivingRecordedFrame – 11. Mottagning av inspelad video.

3 Fjärrstyrning av Ethisis Client	3:1
3.1 Ethisis Client Remote control.....	3:1
3.1.1 Översikt	3:1
3.1.2 Installation.....	3:1
3.1.3 Prova exempelapplikationen.....	3:2
3.1.4 Fjärrkontrollens gränssnitt	3:9

3 Fjärrstyrning av Ethisis Client

3.1 Ethisis Client Remote control

3.1.1 Översikt

Syftet med Ethisis Client Remote control är att låta andra system kontrollera en eller flera Ethisis Client med avseende på vad som visas live eller som inspelad video.

Det andra systemet kan köra lokalt på samma dator som Ethisis Client eller så kan det köra på en annan dator. Det viktiga att hålla i minnet är att *Ethisis Client Remote control* ska installeras på samma dator som systemet som ska använda kontrollen kör.

I Ethisis setup finns en specifik *Setup Type* som heter *Remote control of Ethisis Client* som kan väljas när man endast installerar fjärrkontrollen. Om du även vill installera andra Ethisiskomponenter tillsammans med fjärrkontrollen, måste du välja *Setup Type Custom* och sedan välja komponenten *Remote control of Ethisis Client* tillsammans med övriga önskade komponenter.

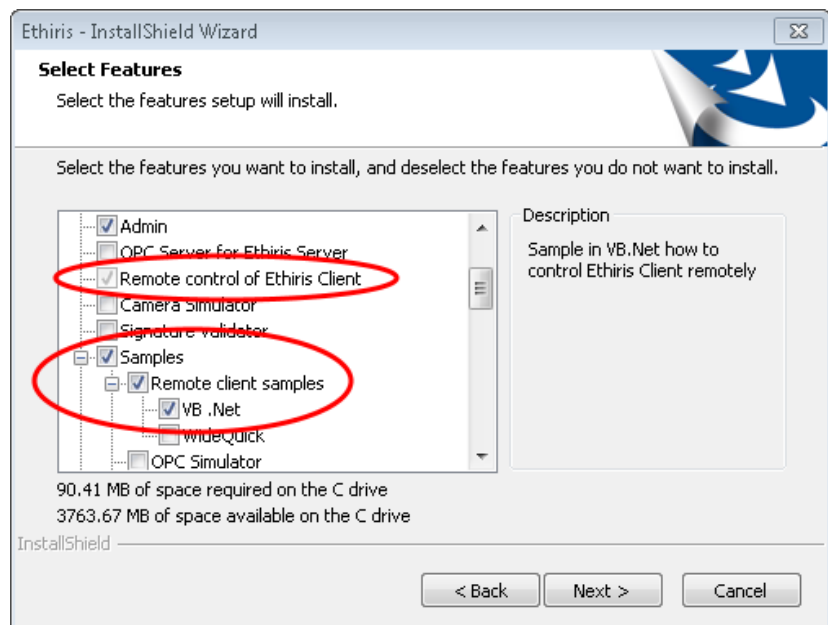
Det finns också två *samples (exempel)* som visar hur fjärrkontrollen ska användas; Ett i Visual Basic .Net 2017 och ett i WideQuick.

3.1.2 Installation

Kör Ethisis setup på den dator där du vill ha fjärrkontrollen. I följande beskrivning av exemplet i VB.Net, utgår vi ifrån att du också installerar *Remote client samples/VB.Net* och att Visual Basic .Net är installerat på datorn.

Fjärrkontrollen kommer så småningom att ansluta till en Ethisis Client någonstans i nätverket. Ethisis Client behöver inte nödvändigtvis vara installerad på samma dator som fjärrkontrollen är, men det går alldeles utmärkt att ha den installerad lokalt med både Ethisis Client och fjärrkontrollen på samma dator.

Om du vill ha exemplet gjort i *WideQuick*, klicka bara i checkrutan för WideQuick i dialogen *Select Features* i installationsguiden.



Figur 3.1 Välj komponenterna Remote client sample & Remote control of Ethis Client.

Komponenten *Remote control of Ethis Client* (själva fjärrkontrollen) är automatiskt vald när något av tillhörande exempel väljs.

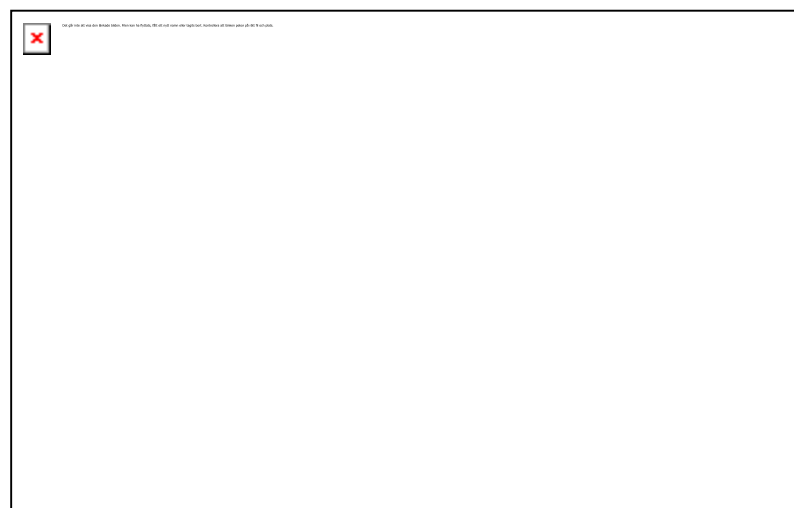
3.1.3 Prova exempelapplikationen



Exempel-
applikationen är
inte avsedd att
direkt användas i
en riktig tillämpning!

Efter installation av exemplet hittar du exempelfilerna i en underkatalog till Ethis installationskatalog. I allmänhet är denna `C:\Program\Kentima AB\Ethis\Samples\RemoteClient\VB.Net`.

Observera att exempelapplikationen bara är till för att visa hur man kan använda *Remote control of Ethis Client*. Den är inte avsedd att direkt användas i en riktig tillämpning!

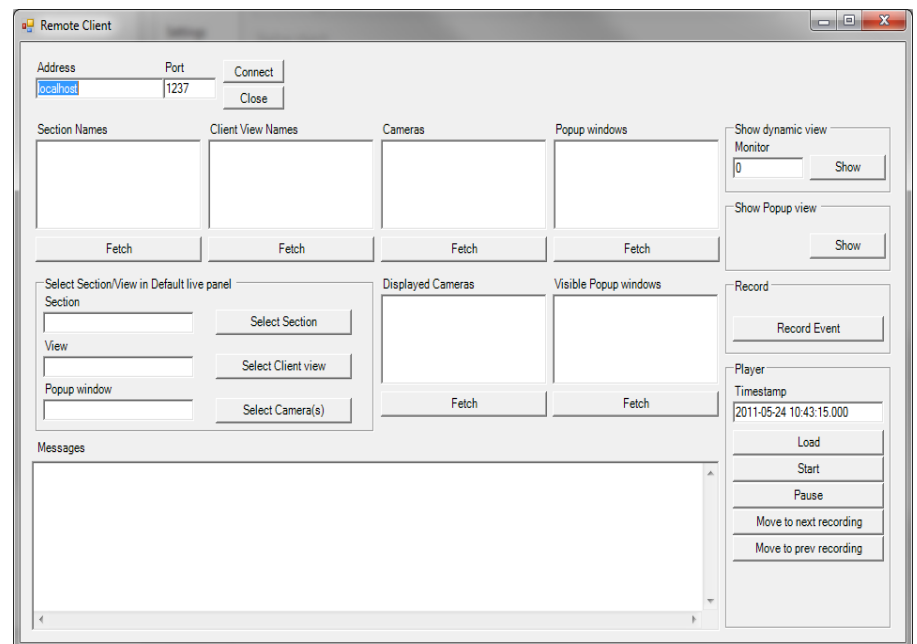


Figur 3.2 Exempelfiler för fjärrstyrning av klienten för VB.Net i Windows Utforskaren.

Jag tar för givet att om du känner till Visual Basic .Net, så vet du hur källkodsfilerna ska användas. Förutom källkodsfilerna finns en färdigkompilerad version av exemplet som heter *EthisClientRemoteSample.exe*.

Låt oss prova denna!

Dubbelklicka på *EthisisClientRemoteSample.exe* i Windows Utforskaren. Applikationen startar och ser ut som i Figur 3.3

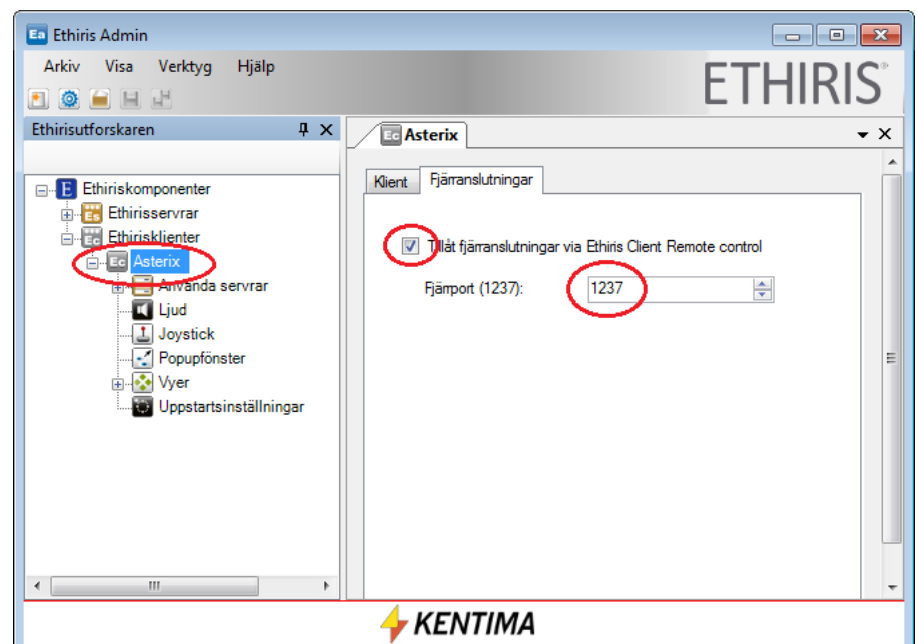


Figur 3.3 Exempelapplikationen i VB.Net har just startats.

Överst i formuläret finns fält för *Address* (Adress) och *Port* (Port). Standardvärdena är *localhost* och *1237*. *localhost* innebär den lokala datorn. Om den Ethisis Client du vill fjärrstyra kör på en annan dator måste du ange *DNS-namnet/IP-adressen* till den datorn istället.

Innan du kan ansluta till en Ethisis Client måste du konfigurera klienten så att den lyssnar på inkommande anrop på port 1237.

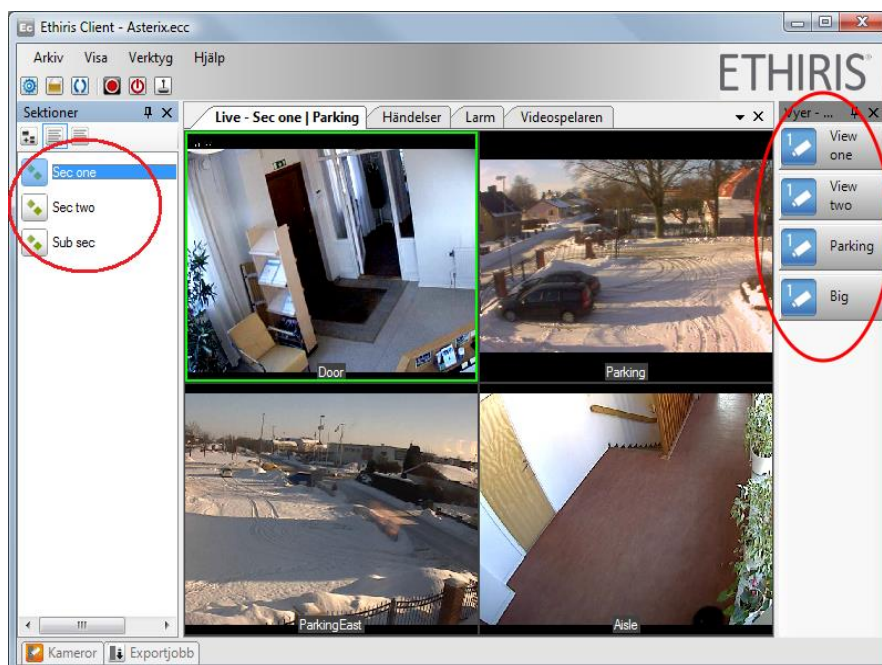
I Ethisis Admin, ser det ut så här:



Figur 3.4 Aktivera fjärranslutning via port 1237 i klientkonfigurationen.

Spara klientkonfigurationen och **starta** sedan *Ethisis Client*.

I mitt fall, ser Ethisis Client ut som nedan:

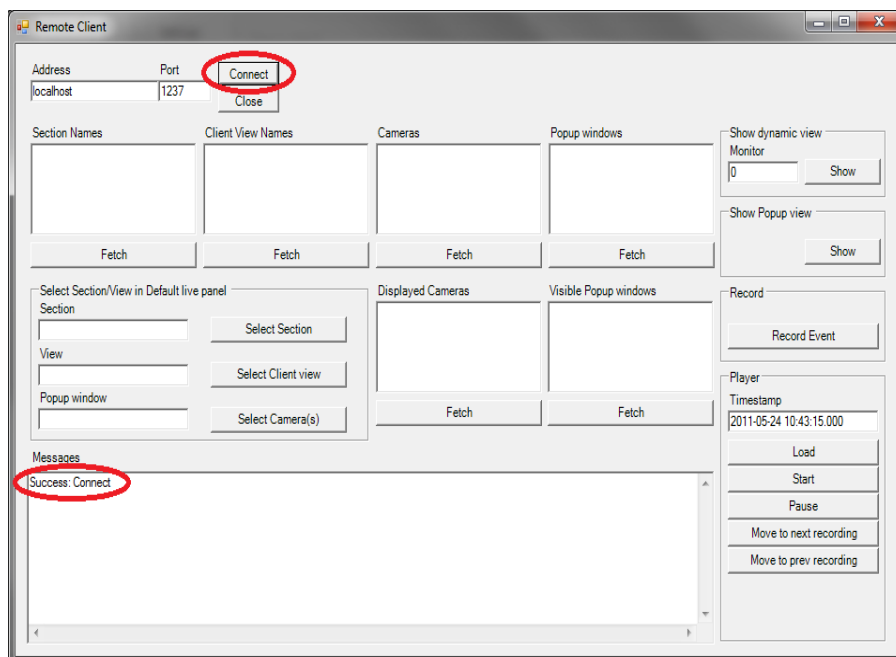


Figur 3.5 Ethis Client med 3 olika sektioner och 4 olika vyer i första sektionen.

I exemplet ovan finns det tre sektioner, som heter *Sec one*, *Sec two* och *Sub sec* och det finns fyra vyer i den första (valda) sektionen som heter *View one*, *View two*, *Parking* och *Big*.

Nu är det dags att ansluta till Ethis Client från exempelapplikationen.

Klicka på knappen *Connect (Anslut)*. Det ska dyka upp ett *message (meddelande)*, som informerar dig om att anslutningen lyckades.



Figur 3.6 Anslutning till Ethis Client lyckades.

Direkt under knapparna *Connect (Anslut)* och *Close (Stäng)* finns 6 listor; *Section Names (Sektionsnamn)*, *Client View Names (Vynamn)*, *Cameras (kameror)*, *Displayed Cameras (Visade kameror)*, *Popup windows (popupfönster)* och *Visible Popup windows (synliga popupfönster)*.

Under varje lista finns en knapp *Fetch* (*Hämta*). Klicka på knappen *Fetch* för att hämta tillhörande lista från Ethiris Client.

Section Names – Klick på knappen *Fetch* hämtar aktuell lista med sektionensnamn från aktuell klientkonfiguration.

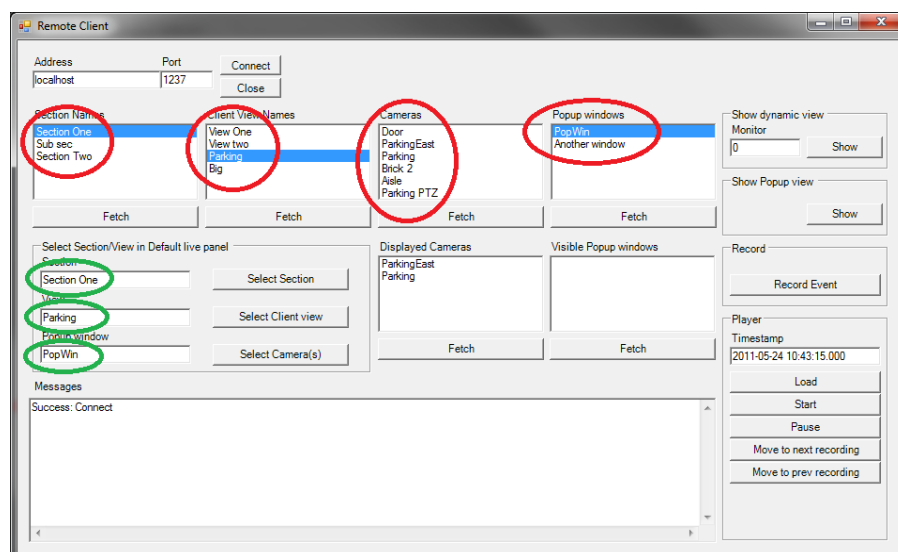
Client View Names – Klick på knappen *Fetch* hämtar namnen på de vyer som ingår i den för tillfället valda sektionen i Ethiris Client. Om du först väljer en sektion i listan *Section Names* i exempelapplikationen, så får du en lista med vynamn för de vyer som ingår i den valda sektionen i exempelapplikationen istället för från den valda sektionen i Ethiris Client..

Cameras – Klick på knappen *Fetch* hämtar aktuell lista med namnen på de kameror som ingår i aktuell klientkonfiguration. Listan är samma som listan i verktygsfönstret *Cameras* (*Kameror*) i Ethiris Client.

Displayed Cameras – Klick på knappen *Fetch* hämtar en lista med namnen på de kameror som för tillfället visas live i Ethiris Client.

Popup windows – Klick på knappen *Fetch* hämtar en lista med namnen på de popupfönster som är definierade i Ethiris Client.

Visible Popup windows – Klick på knappen *Fetch* hämtar en lista med namnen på de popupfönster som för tillfället visas i Ethiris Client.



Figur 3.7 Exempelapplikationen efter att ha klickat på *Fetch* knapparna.

När du väljer en *sektion* genom att klicka på dess namn i listan *Section Names*, kopieras sektionensnamnet till fältet *Section* till vänster om knappen *Select Section* (*Välj sektion*). Samma gäller för *Client View Names* (*Vynamn*) och *Popup windows* (*popupfönster*). Dessa fält är inringade med grönt i exemplet ovan.

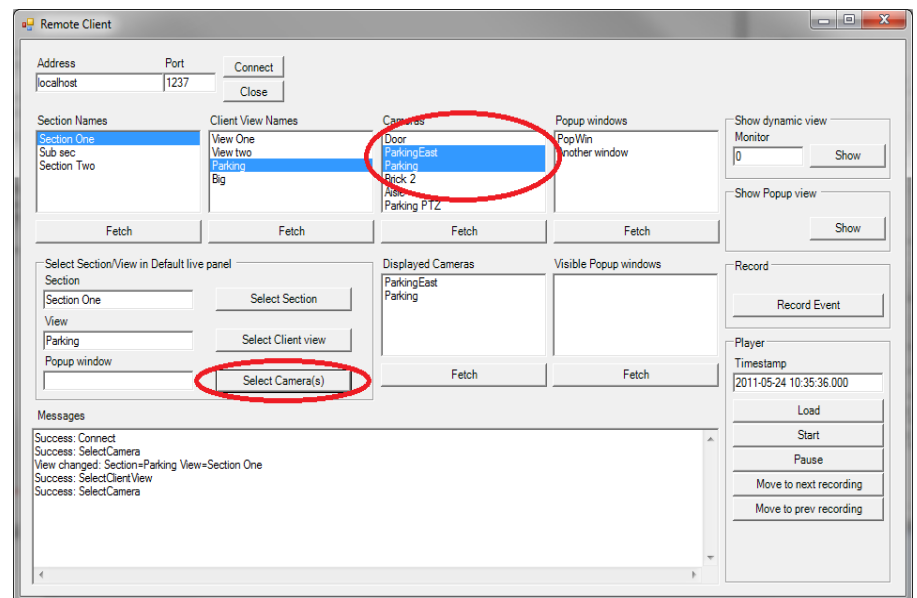
I exemplet ovan är sektionen *Sec one* och vyn *Parking* valda.

Klicka på knappen *Select Section* (*Välj sektion*) för att byta sektion i Ethiris Client och klicka på knappen *Select Client view* (*Välj klientvy*) för att byta vy i Ethiris Client.

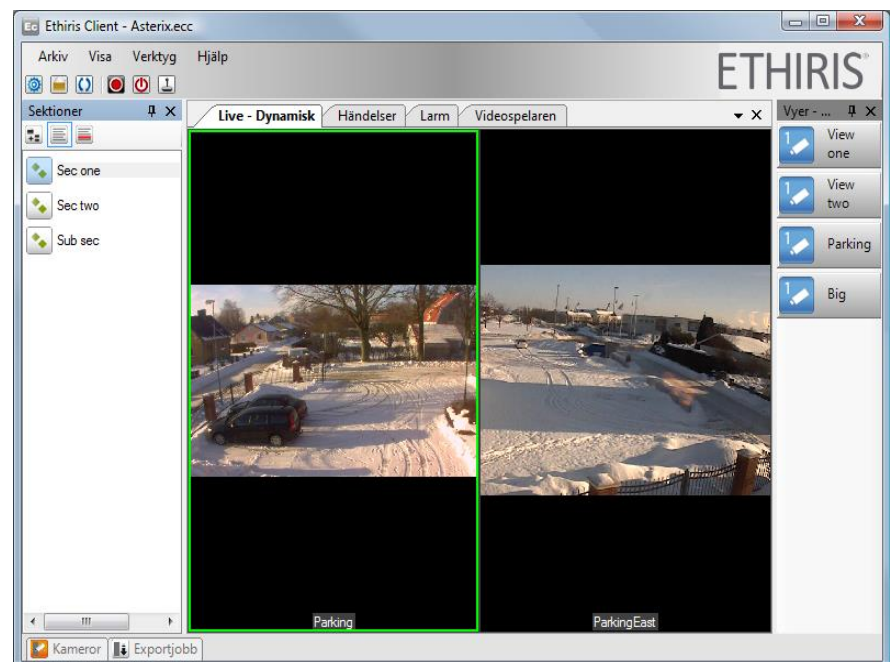
Kameralistan

Listan *Cameras* (*Kameror*) betar sig något annorlunda eftersom du kan välja flera kameror i listan samtidigt. Klicka på en kamera för att välja den. Klicka igen för att välja bort den. Genom att klicka på flera kameror, en i taget, kan du välja flera kameror i listan. Se exemplet nedan.

Om du nu klickar på knappen *Select Camera(s) (Välj kameror)* kommer en dynamisk vy att skapas i Ethis Client och visa kamerorna i *Default Live panel*.



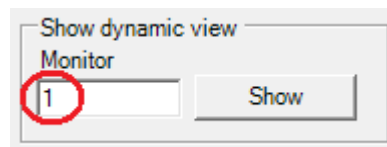
Figur 3.8 Två kameror valda.



Figur 3.9 En dynamisk vy med de två valda kamerorna i Ethis Client.

Klick på knappen *Select Camera(s) (Välj kameror)* har samma effekt som att klicka på knappen *Show (Visa)* med *Monitor 0* i avsnittet *Show dynamic view (Visa dynamisk vy)* i exempelapplikationen.

Du kan ange ett annat monitornummer såsom 1 för monitor 1. Klick på knappen *Show (Visa)* kommer då att skapa en dynamisk vy i ett eget fönster på den specificerade monitorn.

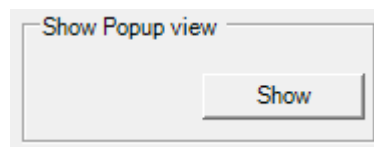


Figur 3.10 Visa en dynamisk vy på monitor 1.

Den dynamiska vyn kommer att visa kameror som är valda i listan *Cameras*.

Popupfönster

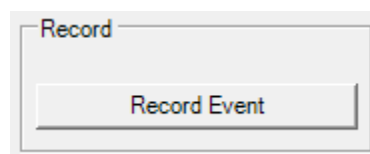
Klicka på knappen *Show* för att visa den fördefinierade vy som anges under *View (Vy)* i det fördefinierade popupfönster som anges under *Popup window (popupfönster)*. Dessutom visas de kameror som är markerade i listan *Cameras (Kameror)* i odefinierade kameravyer i vyn.



Figur 3.11 Visa popupfönster.

Det finns många möjligheter med denna funktion. Om ingen vy är vald, visas de kameror som är valda i listan *Cameras (Kameror)* i en dynamisk vy i det popupfönster som är markerat. Om inga kameror är valda, kommer den markerade vyn att visas i valt popupfönster precis som den är definierad. Om varken vy eller kameror är valda, kommer det markerade popupfönstret att stängas.

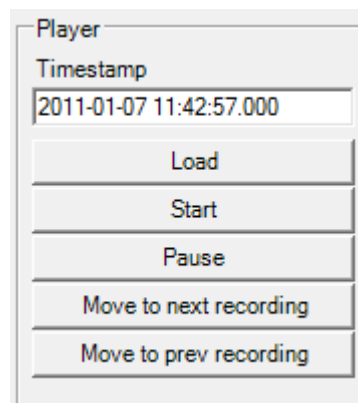
Det finns också ett avsnitt *Record* i exempelapplikationen med en *Record Event (Händelseinspelning)* knapp.



Figur 3.12 Knappen *Record Event* i exempelapplikationen.

Klicka på knappen *Record Event* för att starta en händelseinspelning på kamerorna som är valda i listan *Cameras*.

Slutligen, finns det ett *Player (Videospelare)* avsnitt i exempelapplikationen.



Figur 3.13 Avsnittet *Player* i exempelapplikationen.

När du klickar på knappen *Load (Ladda)*, visar Ethis Client fliken *Videospelaren*, laddar den med kamerorna som är valda i listan *Cameras* och sätter tidslinjan till tiden som står i fältet *Timestamp (Tidsstämpel)* i exempelapplikationen.

Nu kan du manövrera videospelaren med de andra knapparna i exempelapplikationen; *Start (Start)*, *Pause (Paus)*, *Move to next recording (Hoppa till nästa videosegment)* och *Move to prev recording (Hoppa till föregående videosegment)*.

Testa dem!

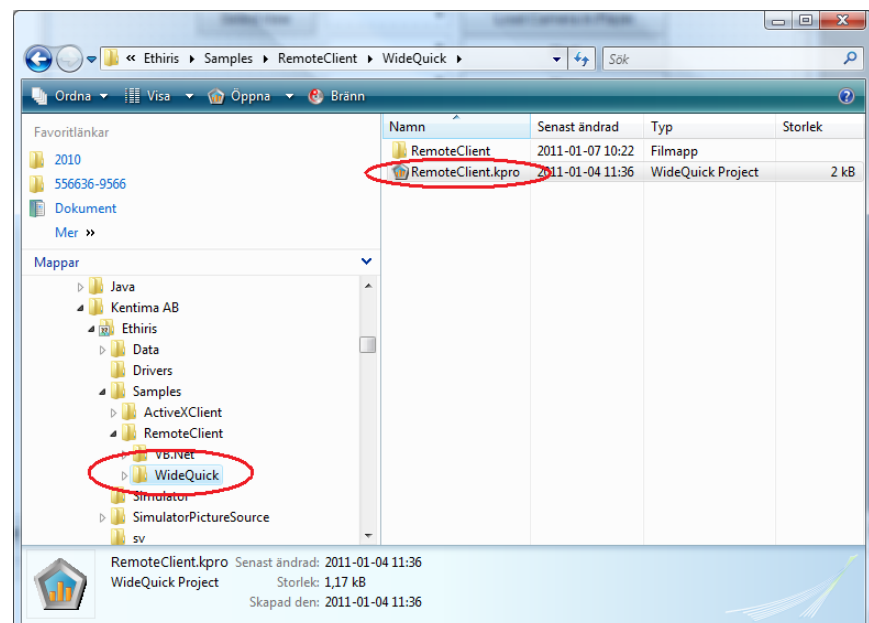
WideQuick exempel



Exempelapplikationen är inte avsedd att direkt användas i en riktig tillämpning!

Om du valde *WideQuick* exemplet i installationen, kommer exempelprojektet att placeras i en underkatalog till Ethis installationskatalog. I allmänhet är denna *C:\Program Files\Kentima AB\Ethis\Samples\RemoteClient\WideQuick*.

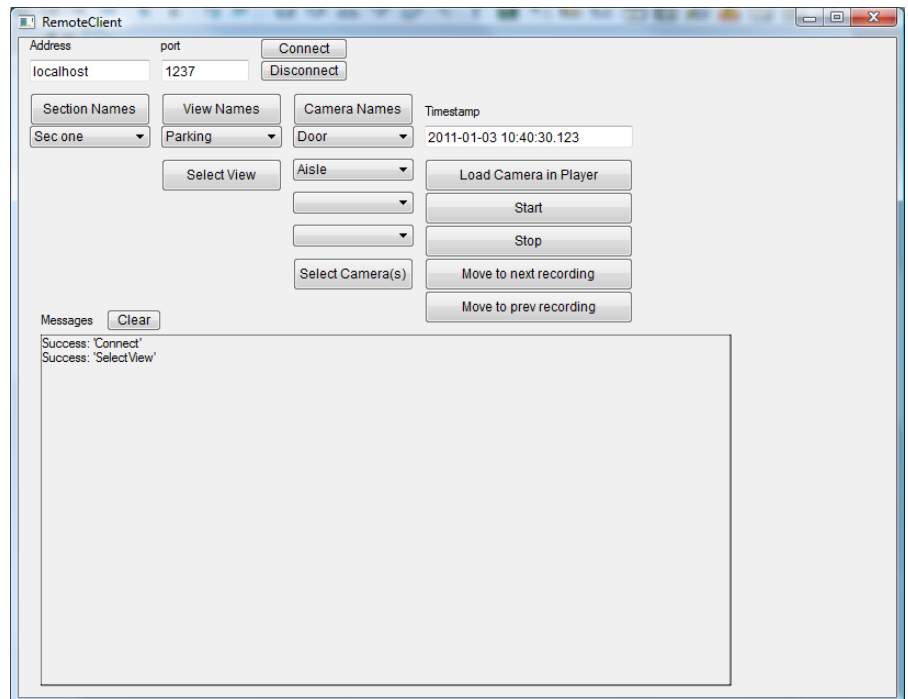
Observera att exempelapplikationen bara är till för att visa hur man kan använda *Remote control of Ethis Client*. Den är inte avsedd att direkt användas i en riktig tillämpning!



Figur 3.14 Exempelprojektet för fjärrstyrning med WideQuick i Windows Utforskaren.

Starta helt enkelt *WideQuick Designer* och öppna projektet *RemoteClient.kpro*.

När du kör projektet i förhandsgranskning, så ser det ut ungefär som exemplet i VB.Net.



Figur 3.15 WideQuick förhandsgranskning med exempelprojektet.

De flesta metoder och händelser är implementerade i exempelprojektet. Var snäll och se exemplet i VB.Net ovan för information om syftet med de olika listorna och knapparna.

3.1.4 Fjärrkontrollens gränssnitt

När du använder Ethisis Client Remote control så använder du kontrollens gränssnitt. Följande är en beskrivning av de olika tillgängliga egenskaperna, metoderna, händelserna och uppräkningsarna.

Egenskaper

Det finns inga egenskaper för Ethisis Client Remote control.

Metoder

Close()

Stänger anslutningen till Ethisis Client.

Connect (Address as String, Port as Long)

Ansluter till den Ethisis Client instans som enligt parametrarna ska fjärrstyras.

Address – En sträng som innehåller IP-adress eller DNS-namn till datorn som kör Ethisis Client.

Port – Portnumret som den Ethisis Client lyssnar på för fjärrstyrningskommandon. Detta är som standard 1237.

GetCameraNames()

Initialiserar överföring av en array med kameranamn för alla tillgängliga kameror i klienten. Resultatet returneras i händelsen *CameraNames*.

GetDisplayedCameras()

Initialiserar överföring av en array med kameranamn för alla kameror som för tillfället visar livebilder i klienten. Resultatet returneras i händelsen *DisplayedCameras*.

GetPopupWindowNames()

Initialiserar överföring av en array med strängar innehållande namnen på de popupfönster som är definierade i klienten. Resultatet returneras i händelsen *PopupWindowNames*.

GetSectionNames()

Initialiserar överföring av en array med strängar innehållande namnen på de sektioner som är definierade i klienten. Resultatet returneras i händelsen *SectionNames*.

GetViewNames([SectionName as String])

Initialiserar överföring av en array med strängar innehållande namnen på de vyer som ingår i aktuell eller specificerad sektion. Resultatet returneras i händelsen *ClientViewNames*. Parametern är valfri. Om den är med kommer vyer som tillhör sektionen *SectionName* att returneras. Om ingen sektion specificeras, kommer vyerna som tillhör den för tillfället valda sektionen i Ethis Client att returneras.

GetVisiblePopupWindows()

Initialiserar överföring av en array med popupfönsternamn för alla popupfönster som för tillfället visas i klienten. Resultatet returneras i händelsen *VisiblePopupWindows*.

LoadCamerasInPlayerByTimestamp(Timestamp as String, CameraNames[] as String)

Laddar inspelad videoför de kameror som anges i parametern *CameraNames* i videospelaren.

Timestamp – En sträng som innehåller önskad tid för videospelaren. Denna ska vara i formatet *YYYY-MM-DD hh:mm:ss.ttt*, t ex *2011-01-12 14:14:53.000*.

CameraNames[] – En array med strängar som innehåller namnen på de önskade kamerorna som ska laddas i videospelaren.

MoveToNextRecording()

Hoppa till nästa videosegment i videospelaren.

MoveToPrevRecording()

Hoppa till föregående videosegment i videospelaren.

PausePlayer()

Pausa videospelaren.

RecordEvent(CameraNames[] as String)

Startar en händelseinspelning för de kameror som anges i parametern *CameraNames*.

CameraNames[] – En array med strängar innehållande namnen på de kameror för vilka händelseinspelning ska startas.

ReloadConfiguration([ForceReload as Boolean])

Instruerar klienten att initiera en omladdning av sin konfiguration. Parametern är valfri. Om *ForceReload* är *sant*, kommer konfigurationen alltid att laddas om. Om parametern utelämnas eller är *falsk*, kommer konfigurationen bara att laddas om utifall att konfigurationsfilen verkligen är uppdaterad sedan konfigurationen sist laddades i klienten. Se även händelsen *ReloadPending*.

SelectCamera(CameraNames[] as String)

Visar en dynamiskt skapad livevy i panelen *Default Live panel* med de kameror som anges i parametern *CameraNames*.

CameraNames[] – En array med strängar innehållande namnen på de kameror som den dynamiska vyn ska visa.

SelectSection(SectionName as String, [ViewName as String])

Väljer en specifik sektion i Ethis Client. Om parametern *ViewName* specificeras kommer denna vy också att väljas.

SectionName – En sträng som innehåller namnet på den sektion som ska väljas.

ViewName – En valfri sträng som innehåller namnet på den vy som ska väljas.

SelectView(ViewName As String, [SectionName as String])

Väljer en specifik vy i aktuell eller specificerad sektion. Parametern *SectionName* är valfri. Om den anges, kommer en vy som ingår i sektionen *SectionName* att väljas. Om ingen sektion anges, kommer den specificerade vyn som tillhör den för tillfället valda sektionen i Ethis Client att väljas.

ViewName – En sträng som innehåller namnet på den vy som ska väljas.

SectionName – En valfri sträng som innehåller namnet på den sektion som ska väljas.

ShowDynamicView(Monitor as Long, CameraNames[] as String)

Visar en dynamiskt skapad vy på monitorn som anges i parametern *Monitor* med de kameror som anges i parametern *CameraNames*.

Monitor – Ett värde av typen *Long* som anger vilken monitor som den dynamiska vyn ska visas på. Värdet *0* innebär att vyn kommer att visas i *Default Live panel*. Värdet *1* kommer att skapa vyn i ett eget fönster på monitor 1 etc.

CameraNames[] – En array med strängar innehållande namnen på de kameror som den dynamiska vyn ska visa. En tom vektor innebär att den dynamiska vyn på vald monitor stängs.

ShowPopupView(PopupWindowName as String, ViewName as String, CameraNames[] as String)

Visar den fördefinierade vy som anges i parametern *ViewName* i det fördefinierade popupfönster som anges med parametern *PopupWindowName*. Fyller dessutom i de kameror som anges i parametern *CameraNames* i odefinierade kameravyer i vyn.

PopupWindowName – Ett värde av typen *String* som anger vilket popupfönster som skall användas.

ViewName – Ett värde av typen *String* som anger vilken vy som skall användas som layout för att visa kamerorna i parametern *CameraNames()*. Om denna parameter är en tomsträng, kommer kamerorna att fördelas i en dynamiskt skapad vy. Om vyn innehåller definierade kameravyer (kameror, klientronderingar, hotspots eller bakgrundsbilder), kommer dessa inte att ersättas med kameror som anges i parametern *CameraNames*. Endast kameravyer av typen *Kamera*, men utan definierad kamera kommer att fyllas i.

CameraNames[] – En array med strängar innehållande namnen på de kameror som den valda eller dynamiska vyn ska visa. En tom vektor innebär att den definierade vyn visas i sin ursprungsform.

Om både parametern *ViewName* och *CameraNames* är tomma, kommer det specificerade popupfönstret att stängas.

StartPlayer()

Startar uppspelning av inspelad video som laddats i videospelaren.

Händelser

CameraNames(Names[] as String)

Denna händelse skickas från fjärrkontrollen efter att kommandot *GetCameraNames* har skickats till kontrollen. Följande parametrar skickas med händelsen:

Names[] – En array med strängar innehållande namnen på de tillgängliga kamerorna i Ethis Client.

DisplayedCameras(Cameras[] as String)

Denna händelse skickas från fjärrkontrollen efter att kommandot *GetDisplayedCameras* har skickats till kontrollen. Händelsen skickas även spontant från fjärrkontrollen, när en förändring sker avseende vilka kameror som visas i klienten. Följande parametrar skickas med händelsen:

Cameras[] – En array med strängar innehållande namnen på de kameror som för tillfället visas i Ethis Client.

Failure(Command as eCmdCodes, Code as Long, Message as String)

Denna händelse skickas från fjärrkontrollen när ett fel har detekterats i en operation som inte är direkt initierad av ett anrop från värdapplikationen. Följande parametrar skickas med händelsen:

Command – Ett heltal som anger vilken operation som gick fel. För mer information, se avsnittet *Uppräkningar* nedan.

Code – Om tillämpligt innehåller denna parameter en felkod.

Message – En sträng som innehåller information om problemet.

PlayerLoaded(Names[] as String, Timestamp as String)

Denna händelse skickas från fjärrkontrollen när videospelaren laddas. Följande parametrar skickas med händelsen:

Names[] – En array med strängar innehållande namnen på de kameror som har laddats i videospelaren.

Timestamp – En sträng med tiden i tidslinjalen när videospelaren laddades.

PlayerState(State as ePlayerState)

Denna händelse skickas från fjärrkontrollen när videospelaren ändrar tillstånd, t ex när videospelaren startar uppspelning eller pausar. Följande parametrar skickas med händelsen:

State – Det finns totalt 3 olika tillstånd som beskrivs nedan i avsnittet *Uppräkningar*. Parametern *State* anger videospelarens nya tillstånd.

PopupWindowNames(Names[] as String)

Denna händelse skickas från fjärrkontrollen efter att kommandot *GetPopupWindowNames* har skickats till kontrollen. Följande parametrar skickas med händelsen:

Names[] – En array med strängar innehållande namnen på de tillgängliga popupfönstren i Ethis Client.

Recording(CameraName as String, Start as Boolean)

Denna händelse skickas från fjärrkontrollen när en kamera startar eller stoppar inspelning. Följande parametrar skickas med händelsen:

CameraName – Namnet på kameran som startar/stoppar inspelning.

Start – Ett digitalt värde som är *sant* när inspelning startar och *falskt* när inspelning stoppar.

ReloadPending(WillReload as Boolean)

Denna händelse skickas från fjärrkontrollen som svar på kommandot *ReloadConfiguration(...)*. Följande parameter skickas med händelsen:

WillReload – Ett digitalt värde som är *sant* när klienten kommer att ladda om sin konfiguration och *falskt* om klienten inte kommer att ladda om sin konfiguration.

SectionNames(Names[] as String)

Denna händelse skickas från fjärrkontrollen efter att kommandot *GetSectionNames* har skickats till kontrollen. Följande parametrar skickas med händelsen:

Names[] – En array med strängar innehållande namnen på de tillgängliga sektionerna i Ethis Client.

Success(Command as eCmdCodes)

Denna händelse skickas från fjärrkontrollen när ett kommando har lyckats. Följande parametrar skickas med händelsen:

Command – Ett heltal som anger vilket kommando som lyckades. För mer information, se *Uppräkningar* nedan

ViewChanged(Section as String, View as String)

Denna händelse skickas från fjärrkontrollen när aktuell vy i Ethis Client har ändrats. Följande parametrar skickas med händelsen:

Section – Namnet på eventuellt ny sektion.

View – Namnet på den nya vyn.

ViewNames(SectionName as String, Names[] as String)

Denna händelse skickas från fjärrkontrollen efter att kommandot *GetViewNames* har skickats till kontrollen. Följande parametrar skickas med händelsen:

SectionName – Namnet på aktuell eller specificerad sektion.

Names[] – En array med strängar som innehåller namnen på de tillgängliga vyerna i sektionen.

VisiblePopupWindows(PopupWindows[] as String)

Denna händelse skickas från fjärrkontrollen efter att kommandot *GetVisiblePopupWindows* har skickats till kontrollen. Händelsen skickas även spontant från fjärrkontrollen, när något popupfönster har öppnats eller stängts i klienten. Följande parametrar skickas med händelsen:

PopupWindows[] – En array med strängar innehållande namnen på de popupfönster som för tillfället visas i Ethisis Client.

Uppräkningar

eCmdCodes innehåller 13 olika värden från 0 till 12 som beskrivs nedan:

eccUnknown – 0. Kommandot okänt.

eccConnect – 1. Kommandot *Connect* (Anslut).

eccDisconnect – 2. Kommandot *Close* (Stäng anslutning).

eccSelectSection – 3. Kommandot *SelectSection* (Välj sektion).

eccSelectView – 4. Kommandot *SelectView* (Välj vy).

eccSelectCamera – 5. Kommandot *SelectCamera* (Välj kamera).

eccRecordEvent – 6. Kommandot *RecordEvent* (Händelseinspelning).

eccLoadCamerasInPlayerByTimestamp – 7. Kommandot *LoadCamerasInPlayerByTimestamp* (Ladda kameror i videospelaren med tidsstämpel).

eccStartPlayer – 8. Kommandot *StartPlayer* (Starta uppspelning i videospelaren).

eccPausePlayer – 9. Kommandot *PausePlayer* (Pausa uppspelning i videospelaren).

eccShowDynamicView – 10. Kommandot *ShowDynamicView* (Visa dynamisk vy).

eccMoveToNextRecording – 11. Kommandot *MoveToNextRecording* (Hoppa till nästa videosegment).

eccMoveToPrevRecording – 12. Kommandot *MoveToPrevRecording* (Hoppa till föregående videosegment).

eccShowPopupView – 13. Kommandot *ShowpopupView* (Visa vy i popupfönster).

ePlayerState innehåller 3 olika värden från 0 till 3 som beskrivs nedan:

epsUnknown – 0. Videospelaren har ett okänt tillstånd.

epsRun – 2. Videospelaren är startad.

epsPause – 3. Videospelaren är pausad.

4 Modbus OPC Server	4:1
4.1 Modbus OPC Server	4:1
4.1.1 Översikt	4:1
4.1.2 Installation	4:1
4.1.3 Konfiguration	4:1

4 Modbus OPC Server

4.1 Modbus OPC Server

4.1.1 Översikt

Syftet med Kentima Modbus OPC Server är att kommunicera med system via Modbus protokollet. I detta avsnitt kommer vi att upprätta kommunikation mellan Ethisis Server och en Beckhoff BK9000 enhet, som används för distribuerad I/O.

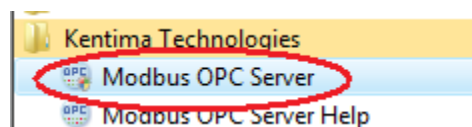
För att få detta att fungera behöver vi minst licensnivå *Extended* för Ethisis Server, eftersom vi kommer att använda OPC Client funktionaliteten i Ethisis Server. Vi behöver också installera *Kentima Modbus OPC Server* på samma dator som Ethisis Server. Då kommer Ethisis Server att använda COM istället för DCOM, som är mycket besvärligare att konfigurera på komplexa säkerhetsinställningar. DCOM är utanför den här manualens täckningsområde.

4.1.2 Installation

Starta installationen av *Kentima Modbus OPC Server* på samma dator som Ethisis Server kör på. Detta installationsprogram kräver en *produktkod*. Se till att du har tillgång till en produktkod för Kentima Modbus OPC Server innan du försöker installera OPC-Servern.

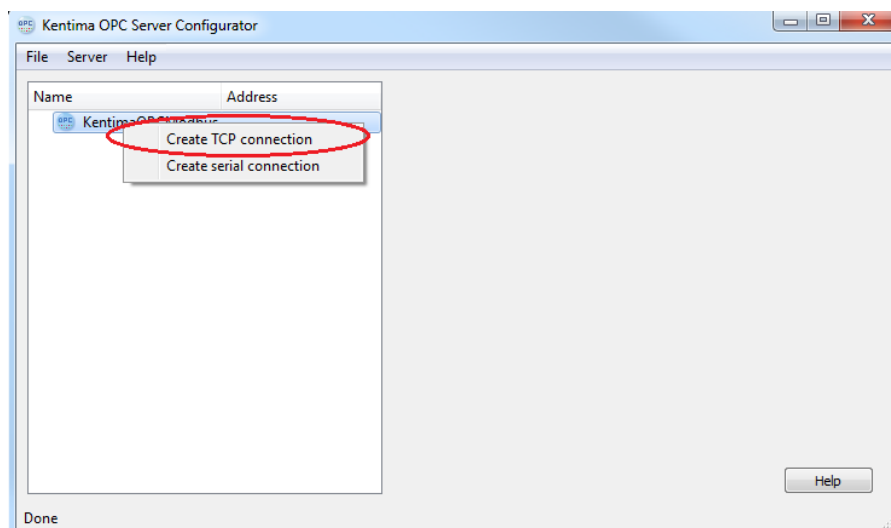
4.1.3 Konfiguration

Först måste vi konfigurera OPC-Servern. **Starta** programmet *Modbus OPC Server* i startmenyn.



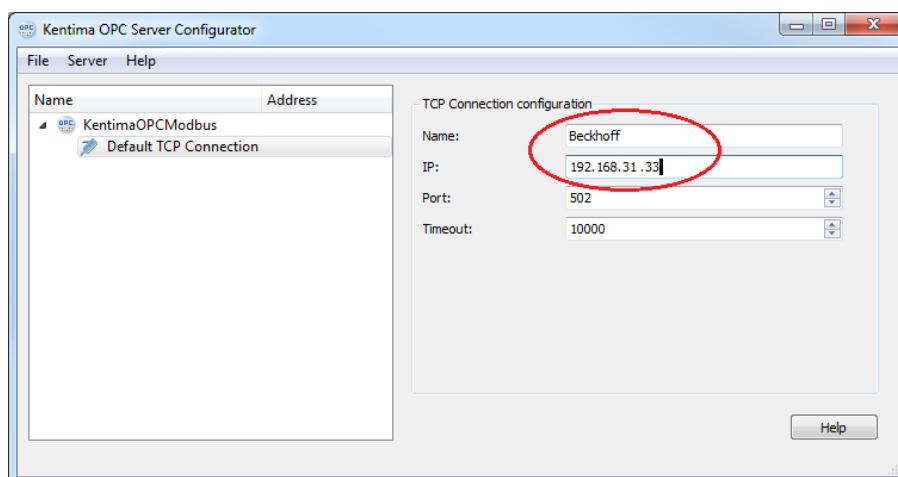
Figur 4.1 Programmet Modbus OPC Server i startmenyn.

Högerklicka på *KentimaOPCModbus* i trädvyn och **välj** *Create TCP connection* (*Skapa TCP-anslutning*) i popupmenyn.



Figur 4.2 Skapa en TCP-anslutning.

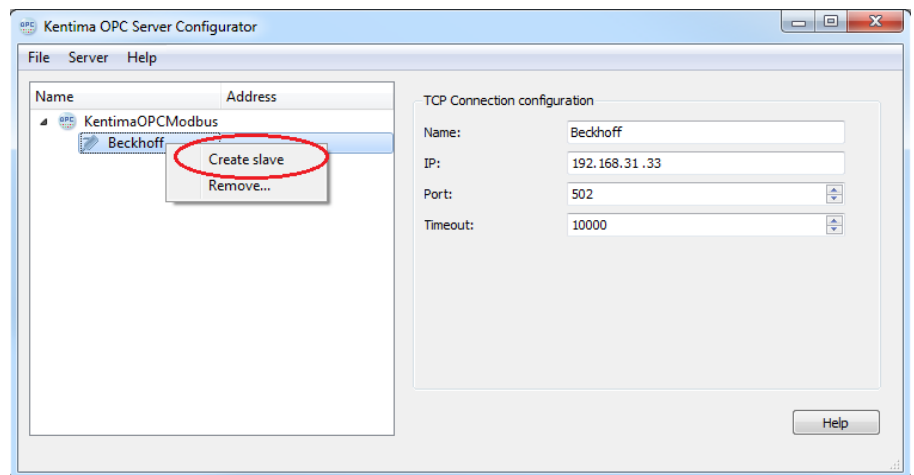
En ny nod kallad *Default TCP Connection* skapas i trädvyn. **Ange IP-adress** till Beckhoff-enheten.



Figur 4.3 Ange IP-adress.

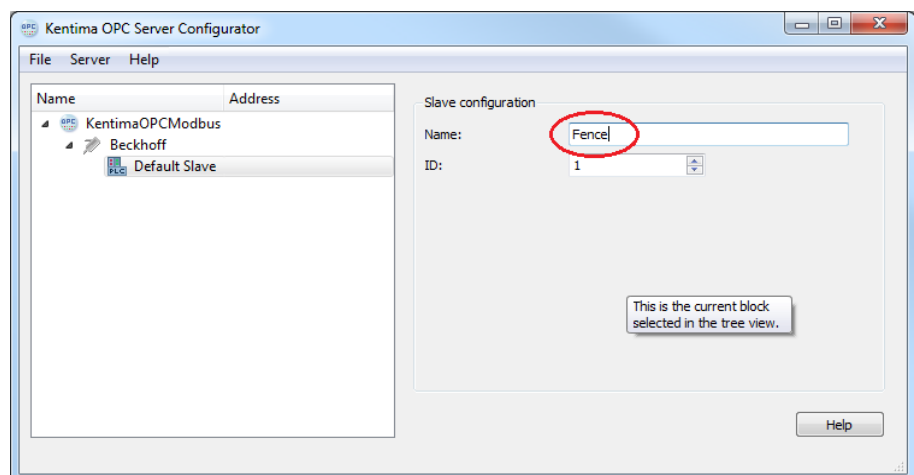
Om du vill kan du ändra namn på TCP-anslutningen, men detta är inte nödvändigt. För tydlighets skull ändrar jag namnet till *Beckhoff*.

Högerklicka på *Default TCP Connection (Beckhoff)* noden i trädvyn och **välj** *Create slave (Skapa slav)* i popupmenyn. Observera att namnet uppdateras i trädvyn när du klickar på noden i trädvyn.



Figur 4.4 Skapa slav.

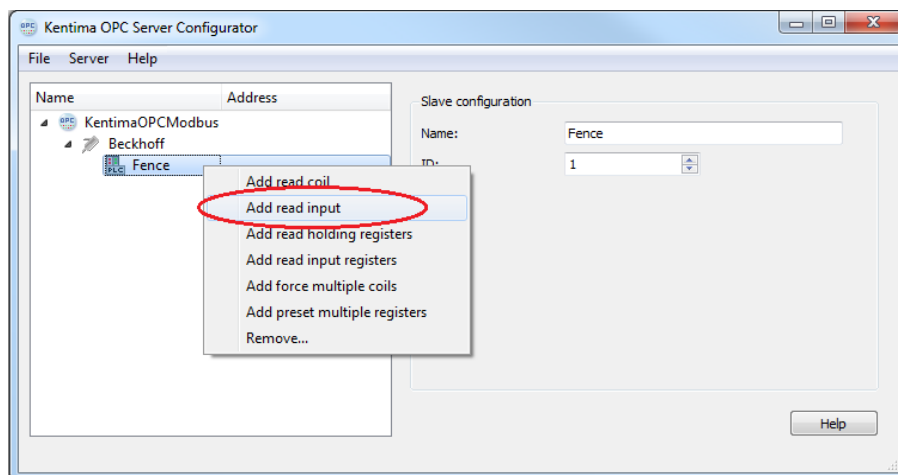
En ny nod dyker upp i trädvyn, *Default Slave*. Du kan ändra namnet för denna nod också. I detta exempel väljer vi namnet *Fence (Stängsel)*, och låtsas att enheten används för någon typ av detektorer till ett stängsel.



Figur 4.5 Slav skapad.

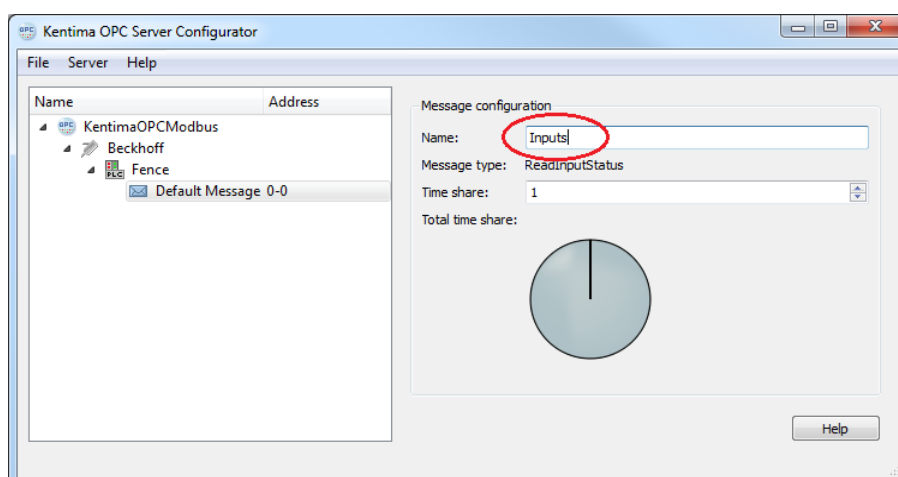
I detta exempel har vi ett digitalt ingångskort (KL1104) anslutet till Beckhoff-enheten. Det finns flera olika modeller av ingångskort med olika antal ingångar. Just denna typ har 4 digitala ingångar.

Nästa steg är att konfigurera de digitala ingångarna i OPC-Servern. All kommunikation sker via meddelanden. Så, allra först skapar vi ett ingångsmeddelande. **Högerklicka** på noden *Default Slave (Fence)* och **välj Add read input (Lägg till läs ingångar)** i popupmenyn.



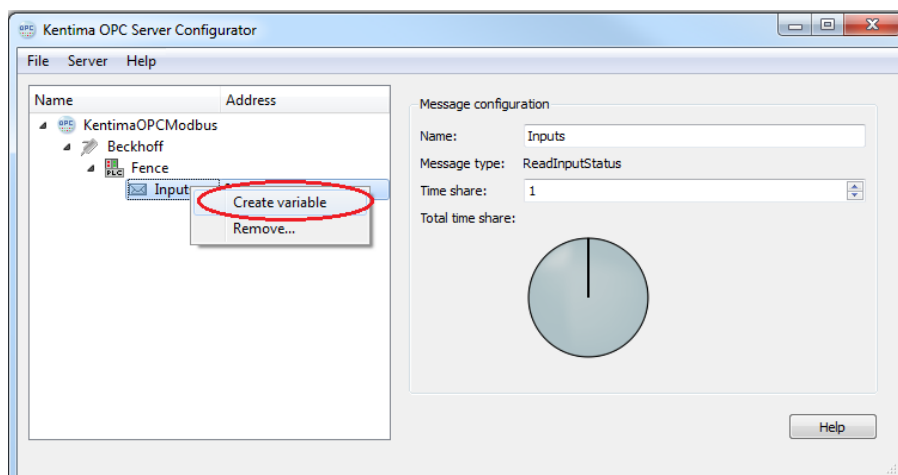
Figur 4.6 Lägg till digitalt ingångsmeddelande.

Ytterligare en nod dyker upp, *Default Message 0-0*. Ändra namnet på meddelandet till något meningsfullt, såsom *Inputs* (*Ingångar*).



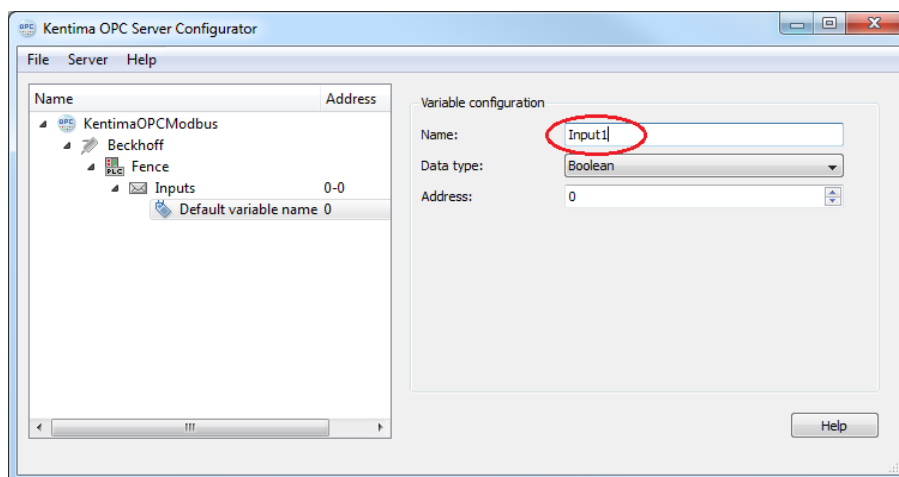
Figur 4.7 Meddelande för att läsa ingångar tillagt.

Nu är det dags för ingångarna. **Högerklicka** på den nya noden *Default Message 0-0* (*Inputs 0-0*) och **välj** *Create variable* (*Skapa variabel*) i popupmenyn.



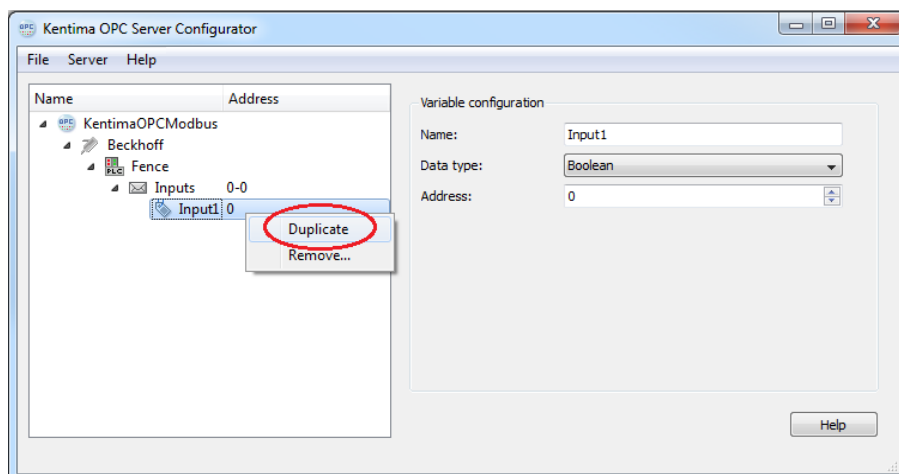
Figur 4.8 Skapa variabel.

En ny nod dyker upp. Ändra namnet till något vettigare, såsom *Input1* (*Ingång1*). Adressen är som standard *0*. Detta är enhetens första adress.



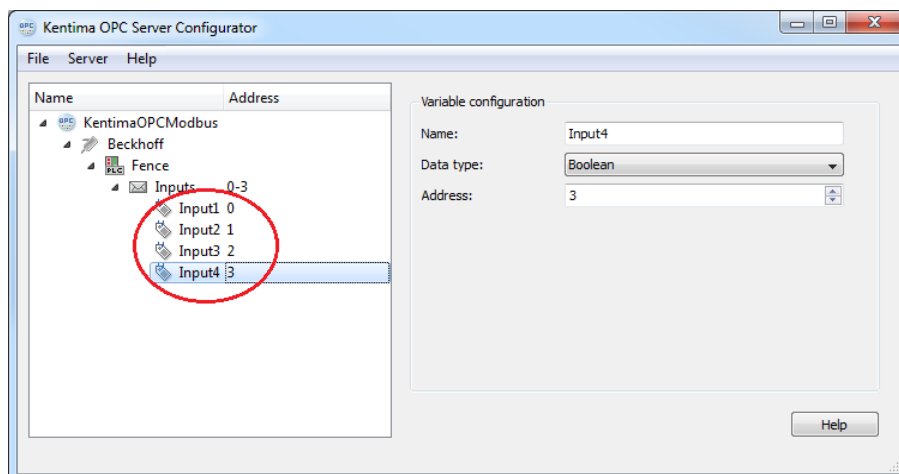
Figur 4.9 Ändra namn på den nya variabeln.

Som nämndes tidigare, vi har 4 digitala ingångar. Du kan använda menyalternativet *Duplicate* (*Duplicera*) för att lägga till ytterligare 3 digitala ingångar till konfigurationen. Nu genereras automatiskt nästa adress och namnet genereras automatiskt i samma stil för följande ingångar.



Figur 4.10 Duplicera variabler.

Efter att ha gjort detta 3 gånger, har vi 4 digitala ingångar.

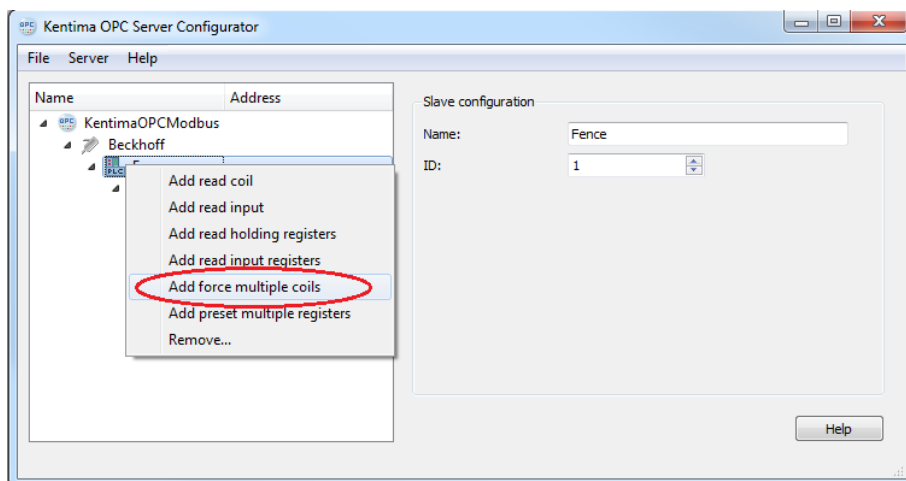


Figur 4.11 4 digitala ingångar skapade.

Vi är nu klara med konfigurationen av våra 4 digitala ingångar. Låt oss nu konfigurera 4 digitala utgångar också. Precis som för ingångskortet finns flera olika modeller av utgångskort att välja mellan. I detta exempel har vi ett kort med 4 digitala utgångar, kallat *KL2134*.

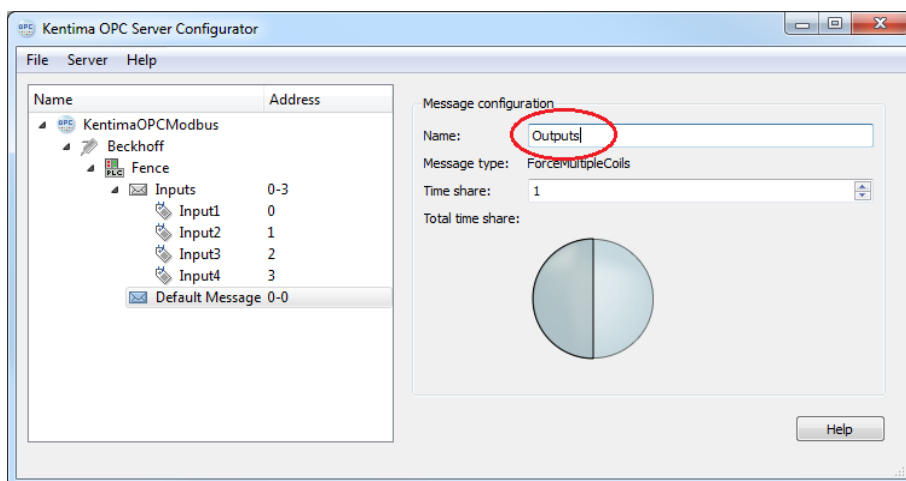
Eftersom vissa delar av enheten redan är konfigurerade, såsom IP-adress, kan vi gå direkt på utgångsmeddelandet.

Högerklicka på noden *Fence* i trädvyn och **välj** menyalternativet *Add force multiple coils*.



Figur 4.12 Lägg till ett utgångsmeddelande.

Ett nytt meddelande skapas.

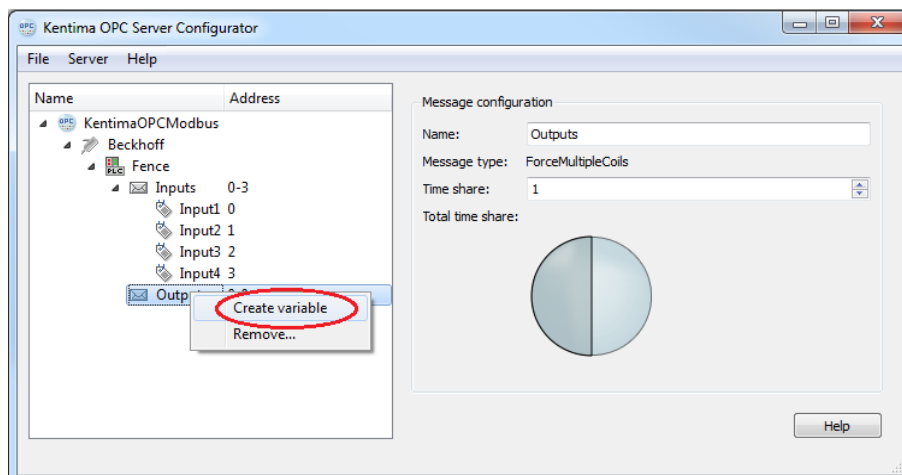


Figur 4.13 Utgångsmeddelande skapat.

Ändra namnet på meddelandet till exempelvis *Outputs* (Utgångar). Ändringen får ingen effekt förrän du klickar i trädvyn.

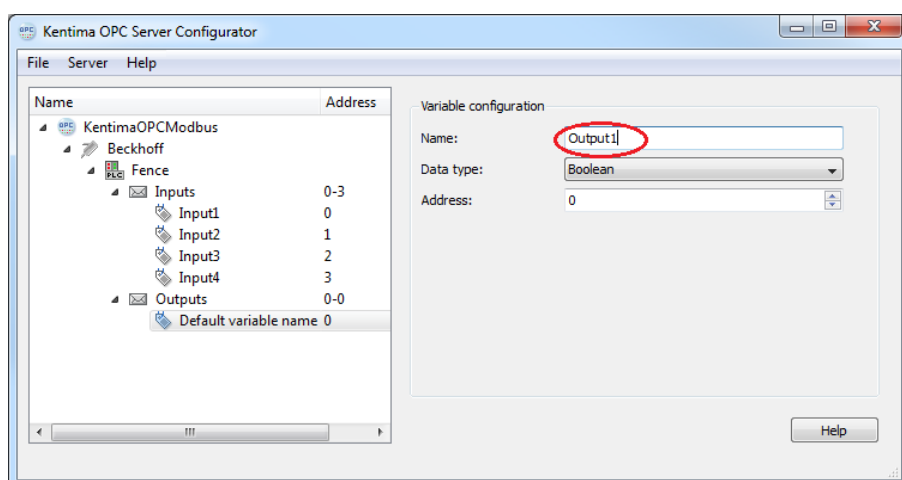
Nästa steg är att skapa utgångssignalerna.

Högerklicka på noden *Outputs* och **välj** *Create variable* (Skapa variabel).



Figur 4.14 Skapa utgångsvariabel.

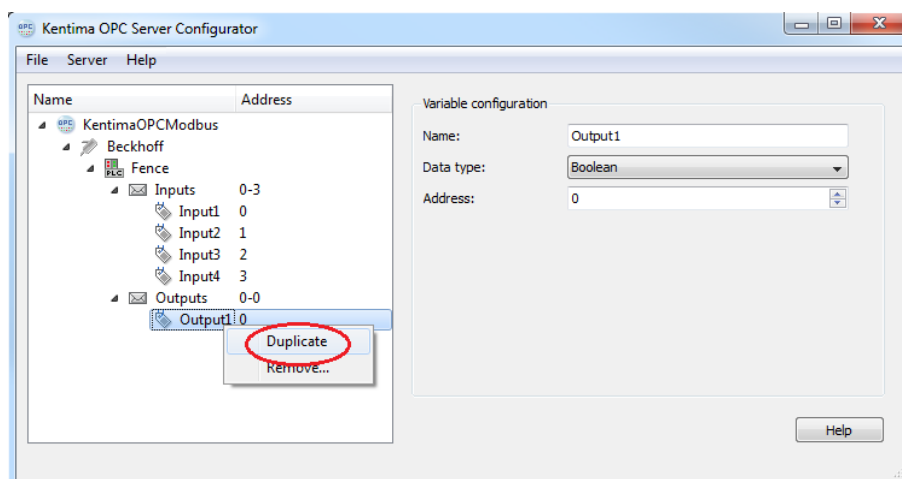
En ny variabel skapas. **Välj den** i trädvyn och **ändra namn** på den till *Output1* (*Utgång1*).



Figur 4.15 Ny utgångsvariabel skapad.

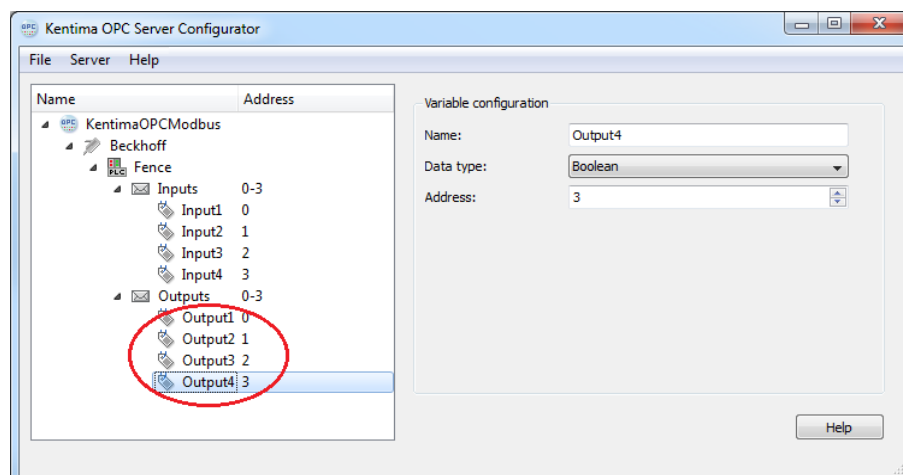
Observera! I detta exempel använder vi oss av en *Beckhoff*-enhet. Den första adressen för utgångar är *0*. Om du använder en *ADAM* enhet, startar den första utgångsadressen på *16*.

Duplicera utgångsvariablerna 3 gånger.



Figur 4.16 Duplicera utgångsvariablerna 3 gånger.

När du är klar, ska konfigurationsdialogen se ut så här:

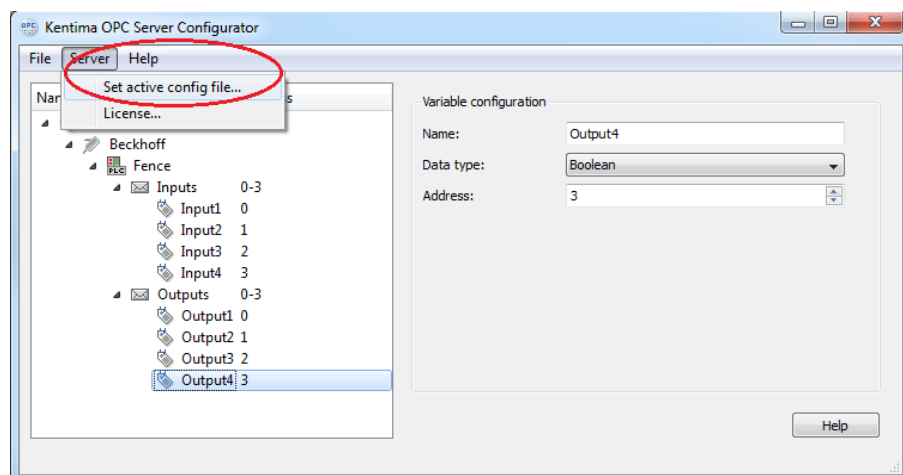


Figur 4.17 Alla meddelanden och variabler är skapade.

OK, nu har vi konfigurerat OPC-Servern för TCP-kommunikation med en enhet med en viss IP-adress. Vi har skapat ett meddelande för 4 digitala ingångar och ett meddelande för 4 digitala utgångar som ska kommuniceras av OPC-Servern.

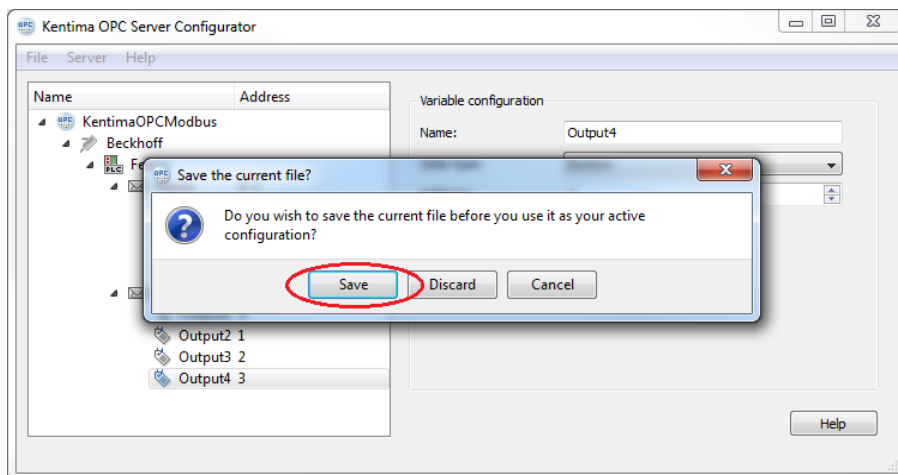
Nästa steg är att spara konfigurationen och tala om för OPC-Servern att börja använda den nya konfigurationen.

Välj menyn *Server->Set active config file...* (Server->Sätt aktiv konfigurationsfil...)



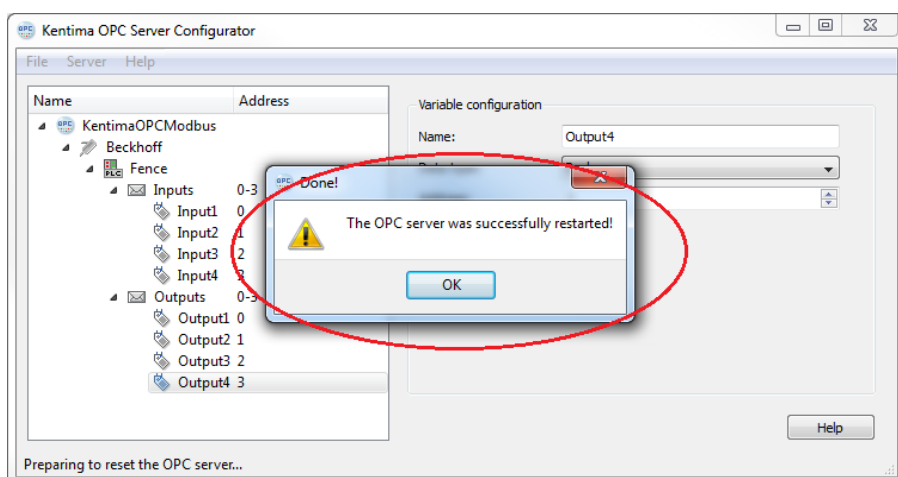
Figur 4.18 Aktivera konfigurationen.

Klicka *Save (Spara)* i dialogen som visas.



Figur 4.19 Spara den nya konfigurationen.

Ett meddelande visas som informerar om att OPC-servern har startats om. Den startas om för att läsa den nya konfigurationen.

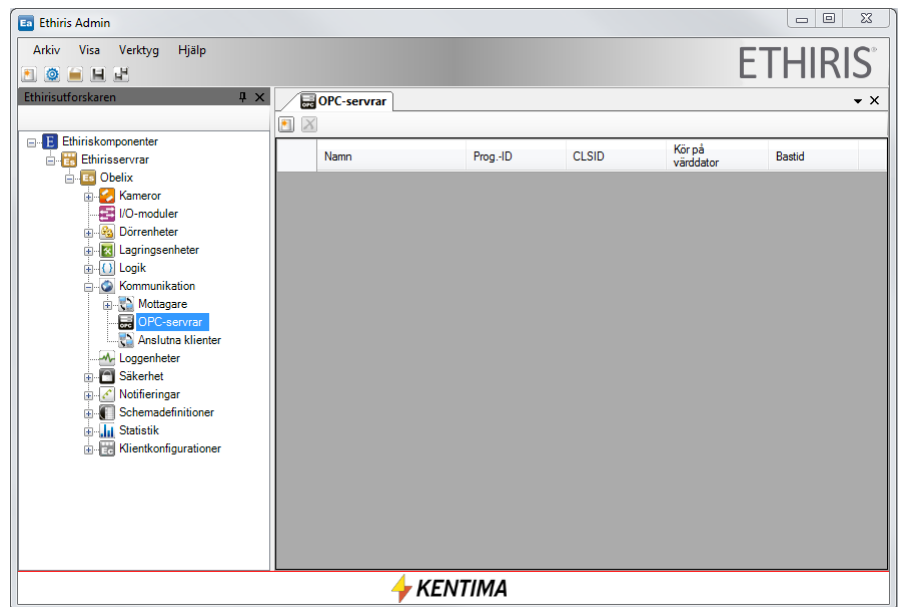


Figur 4.20 OPC-Server omstartad.

Nu är vi klara med konfigurationen av OPC-Servern.

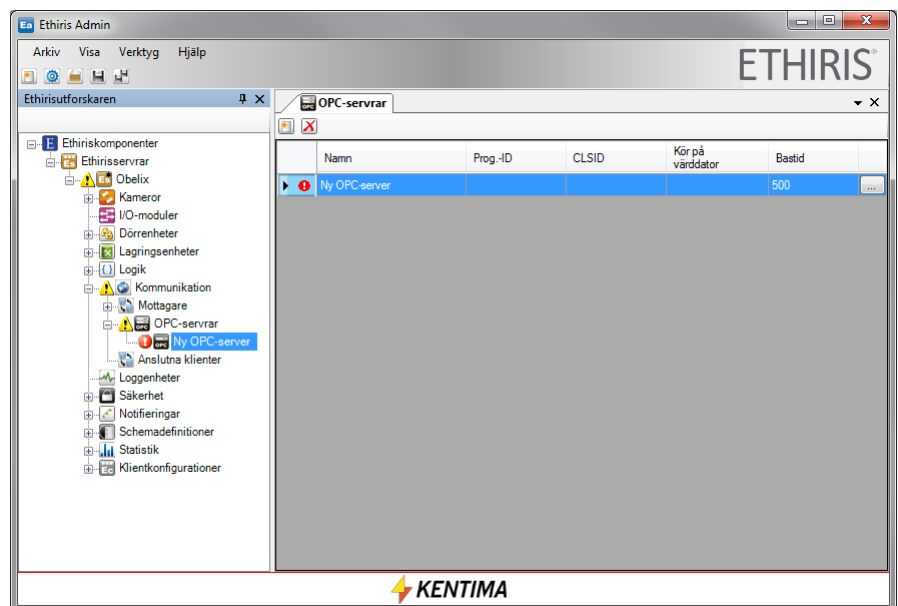
Nästa steg är att konfigurera Ethisis för att kommunicera med OPC-Servern.

Starta Ethisis Admin och **dubbelklicka** på noden *OPC-serverrar* i trädvyn.



Figur 4.21 Panelen OPC Servers i Ethisis Admin.

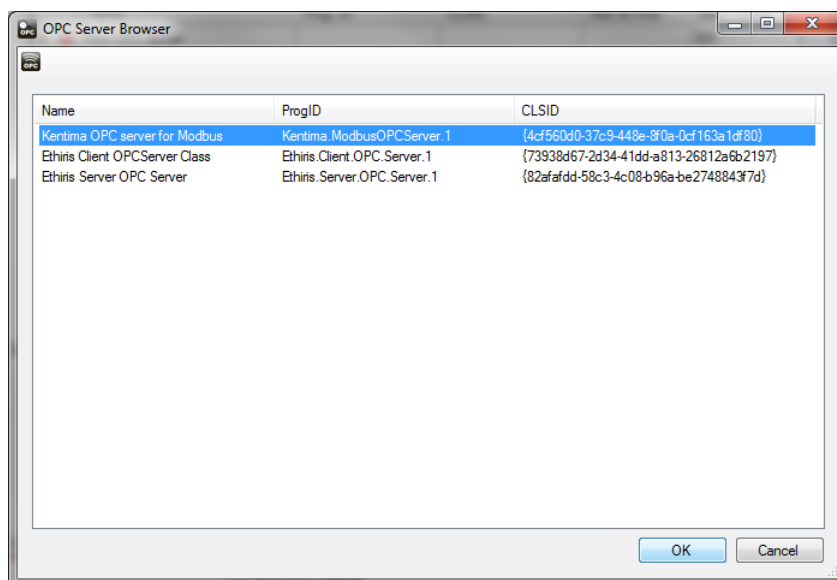
Skapa en ny OPC-Server genom att klicka på knappen *New OPC Server (Ny OPC-Server)*.



Figur 4.22 Ny OPC-Server skapad.

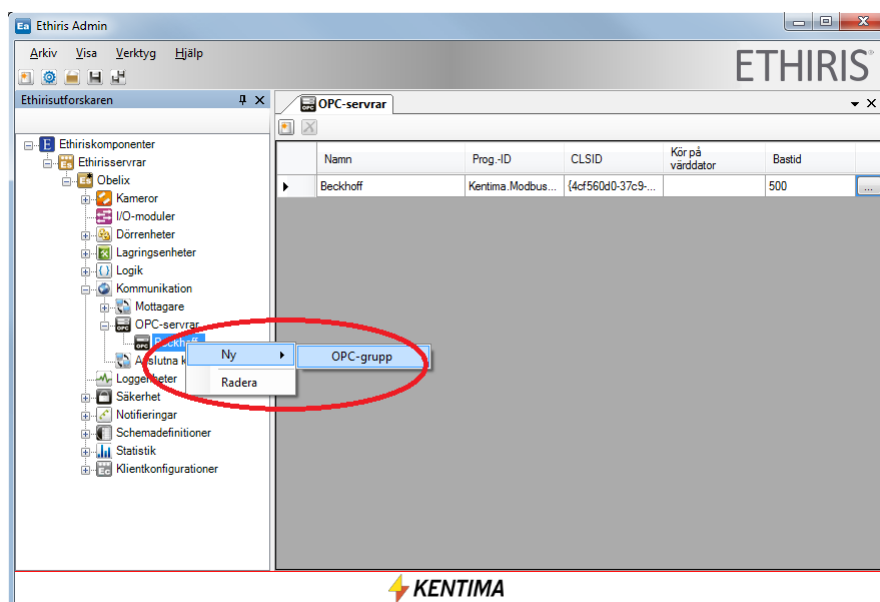
Ändra namnet till något annat, t ex *Beckhoff*.

Klicka på Bläddringsknappen längst till höger för att välja en OPC-Server. I bläddringsdialogen, **välj Kentima OPC server for Modbus**.



Figur 4.23 Välj Kentima OPC server for Modbus

Högerklicka på den *nya noden* i trädvyn och **välj** *New->OPC Group (Ny-OPC-grupp)* i popupmenyn.



Figur 4.24 Skapa en ny OPC-grupp.

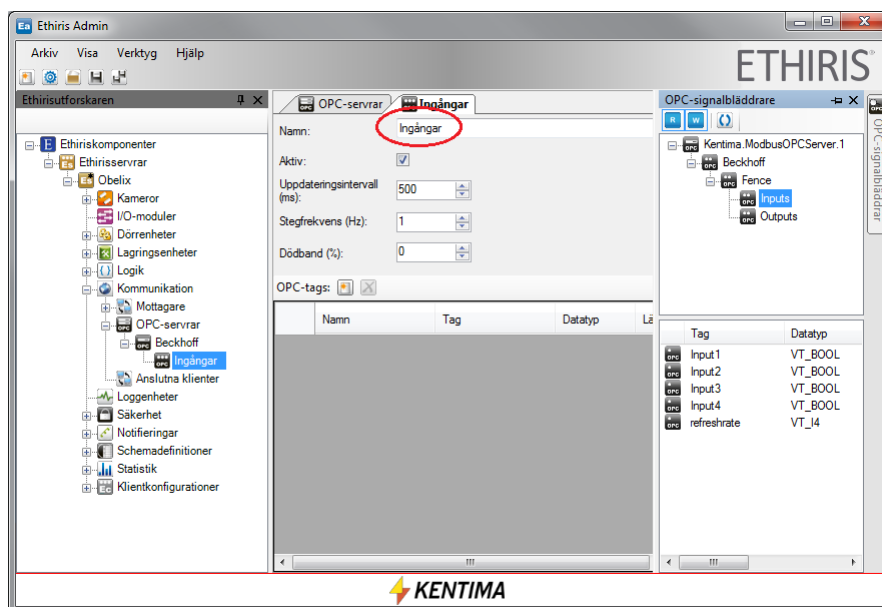
En ny nod skapas i trädvyn.

Dubbelklicka på den *nya noden* för att öppna tillhörande panel.

Ändra namnet på gruppen till något meningsfullt, t ex *Ingångar*.

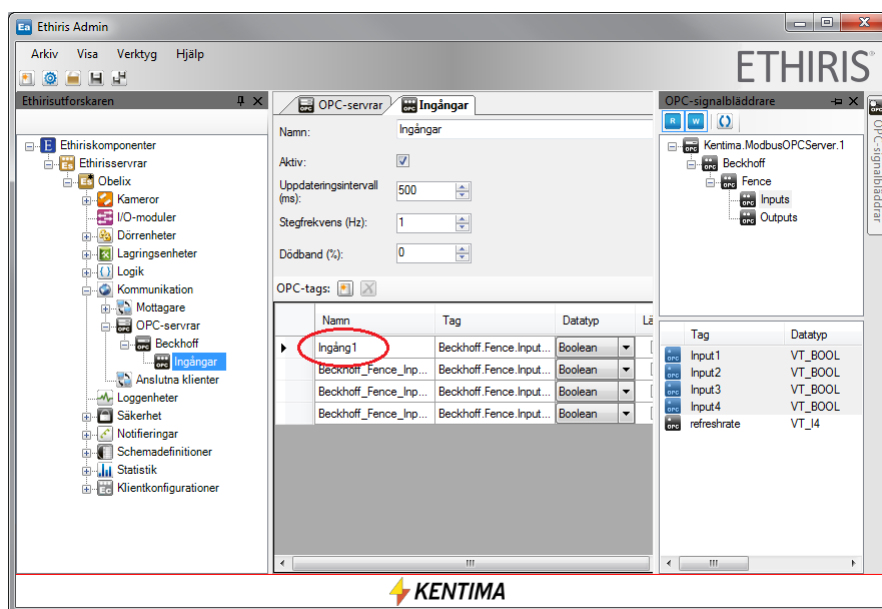
Till höger i panelen finns ett verktygsfönster med en *OPC signalbläddrare* som är dockat. Du kan bläddra i OPC-Servern genom att expandera trädvyn i den övre vyn.

Välj noden *Inputs* i den övre vyn. Nu är de 4 ingångarna synliga i den nedre vyn.



Figur 4.25 Byt namn på OPC-Gruppen.

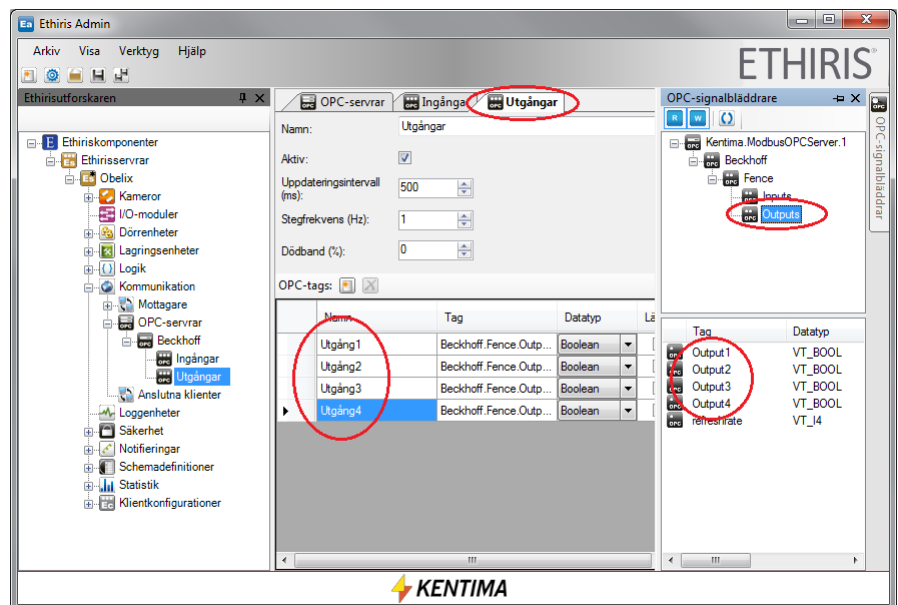
Välj de 4 ingångarna i den nedre vyn och dra-och-släpp dem i listan OPC-Tags till vänster.



Figur 4.26 OPC taggar tillagda.

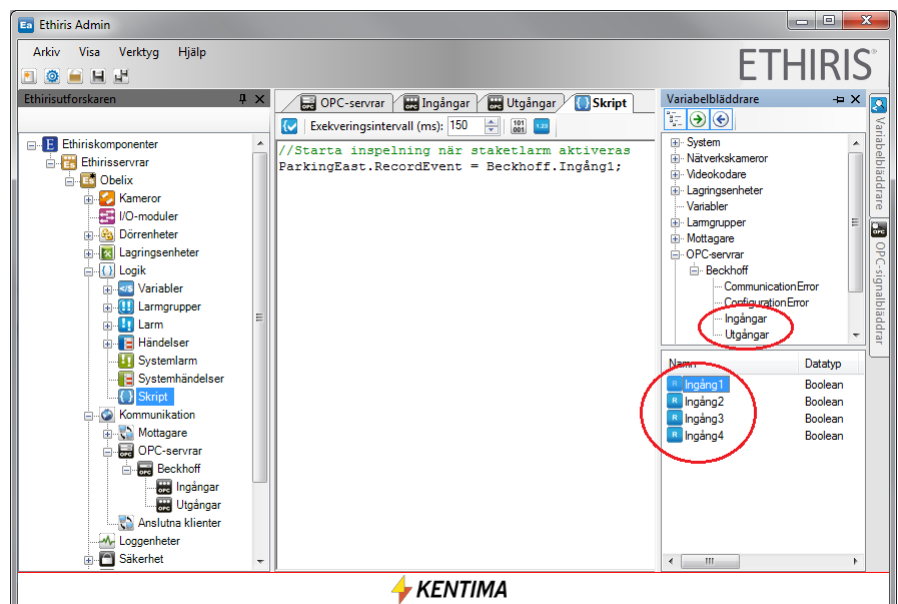
När taggarna läggs till listan, kan du ändra namn på dem till ett något kortare namn.

Om du vill, kan du skapa en annan grupp för utgångarna och lägga till de 4 utgångarna till den gruppen. Ett alternativ är att lägga till alla 8 signalerna (4 ingångar och 4 utgångar) till samma OPC-Grupp.



Figur 4.27 OPC taggar för de 4 utgångarna är också tillagda.

Nu är de nya signalerna tillgängliga i Ethis Servers datalager via *Variabelbläddraren* eller via Ethis Client för presentation i livevyerna.



Figur 4.28 Nya signalerna är tillgängliga i Variabelbläddraren.

5 Mottagare

5:1

5.1 Mottagare	5:1
5.1.1 Översikt	5:1
5.1.2 Mottagare allmänt.....	5:2
5.2 Mottagartyper	5:3
5.2.1 TCP-inkommande	5:3
5.2.2 TCP-utgående	5:4

5 Mottagare

5.1 Mottagare

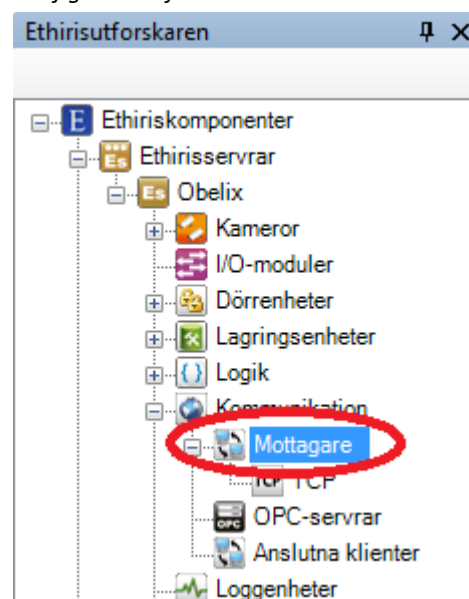
5.1.1 Översikt

Mottagare används i första hand för att lyssna på meddelanden från extern utrustning som t ex kameror. Men, lite motsägelsefullt, kan mottagare även användas för att i viss mån skicka information till extern utrustning så man kan uppnå tvåvägskommunikation. Huvudsyftet är dock att ta emot information, därav namnet *Mottagare*.

Licenskrav

För att samtliga huvudtyper av mottagare skall fungera krävs minst licensnivå *Extended* för Ethis Server.

Det finns en mottagare med begränsad funktionalitet som är tillgänglig för alla licensnivåer. Denna mottagare skapas automatiskt av systemet och går inte att radera. Namnet på denna mottagare är *TCP*. Se vidare manualen *Admin – Konfiguration för Ethis* för mer information om denna mottagare.



Figur 5.1 Noden Mottagare i Ethisutforskarens trädvy.

Mottagartyper

Det finns fyra huvudtyper av mottagare:

- TCP inkommande
- TCP utgående
- HTTP inkommande
- HTTP utgående

Vilken typ man ska använda beror på vilket sätt den externa utrustningen kan kommunicera.

5.1.2 Mottagare allmänt

Mottagare utgör en relativt komplex del av Ethisis där uppsidan är att de kan användas i många sammanhang för att kommunicera med extern utrustning.

Vilket protokoll som ska användas anges med *Typ*, där man kan välja på *TCP* eller *HTTP*.

Via *Typ* anger man också vilken av Ethisis Server och den externa utrustningen som ska ta initiativ till kommunikation.

Inkommande anger att det är den externa utrustningen som ska ta initiativ till kommunikationen. I detta fall lyssnar Ethisis Server på den port som specificerats och väntar på att den externa utrustningen ska börja skicka meddelanden. Tänk på att den angivna porten måste vara öppen i en eventuell brandvägg.

Utgående anger att det är Ethisis Server som ska ta initiativ till kommunikation. Då kommer Ethisis Server att aktivt försöka ansluta till den externa utrustningen med angiven IP-adress och port.

När väl kommunikationen är igång tar Ethisis Server emot text från den externa utrustningen (oavsett om typen är inkommande eller utgående). Det hela går nu ut på att aktivera *triggers* genom att *matcha* tillhörande texter i det textflöde som tas emot av Ethisis Server.

En *trigger* består av tre delar; *Namn*, *Funktion* och *Matcha* (text att matcha).

Varje trigger finns representerad i Ethisis Servers datalager i form av en variabel av typen *Boolean*. En boolean-variabel kan anta två olika värden; *sant* eller *falskt*. I det här sammanhanget innebär detta att triggeren antingen är triggad eller så är den inte triggad.

Varje mottagare kan ha en lista med flera triggers.

Man kan säga att hela processen består av två huvuddelar. Den första delen handlar om att mottagaren tar emot data och den andra delen handlar om att skriptmotorn aktiverar triggerarna.

Mottagningen av data ser lite olika ut beroende på om man har angett *SOT* (*Start Of Text*) och/eller *EOT* (*End Of Text*) eller inte.

Vi börjar med att anta att varken *SOT* eller *EOT* har angetts. Så fort mottagaren har tagit emot ny data kommer den att gå igenom sina triggers och se om någon av triggerarnas *Match*-sträng finns i den mottagna datan. Om ingen matchning finns fortsätter mottagaren att ta emot data och fyller på sin interna buffert med all data som har kommit från den externa utrustningen.

Varje gång ny data kommer till mottagaren går den igenom sina triggers i den ordning som triggerarna är definierade i listan. När en matchning finns kommer all data som har tagits emot så långt att kopieras till en variabel som heter *TriggerString*. Variabeln skapas automatiskt för varje mottagare och är tillgänglig via skript. Triggern kommer att läggas i en kö som skriptmotorn betar av. Flera triggers kan på det här sättet matcha på samma gång för en mottagare. Då kommer de att läggas i tur och ordning i den kö som skriptmotorn betar av. Om någon av triggerarna triggar kommer den interna bufferten av data i mottagaren att tömmas och hela processen börjar om från början igen.

När det finns triggers i kö så kommer skriptmotorn att beta av dessa en i taget. Den trigger som ligger först i kön kommer att vara aktiv under ett skriptvarv. Det innebär att tillhörande variabel är *true* under ett varv i skriptet. Därefter blir variabeln *false* igen och triggern tas bort från kön. Tillhörande *TriggerString* kommer också att tömmas. Därefter plockar skriptmotorn nästa trigger i kön och ettställer dess variabel under ett skriptvarv, och så vidare.

Om vi nu antar att man har angett en text för *SOT*. Nu kommer mottagaren att kasta all data som tas emot fram till att den text som angetts för *SOT* dyker upp i textflödet. Då börjar mottagaren att samla på sig inkommande data och kolla av sina triggers på samma sätt som beskrivits ovan. När vi får en trigging kommer de triggar som triggar att läggas i kö till skriptmotorn, *TriggerString*-variabeln kommer att innehålla den data som har tagits emot från och med *SOT* och därefter kommer den interna bufferten att rensas och mottagaren börjar spana efter en ny *SOT* i inkommande textflöde.

Om vi anger *EOT* kommer inte genomgång av triggers ske förrän texten som angetts för *EOT* dyker upp i den mottagna datan. Har vi ingen matchning då kommer all mottagen data fram till och med *EOT* att kastas och processen börjar om från början igen. Om vi har en trigging kommer de triggers som matchar att läggas i kön till skriptmotorn, *TriggerString* uppdateras och den interna bufferten fram till och med *EOT* töms. Observera att det potentiellt kan finnas flera *EOT* i samma inläsning av data. Då kommer dessa att behandlas en i taget. Först kommer den del av datan fram till och med den första *EOT* att behandlas. Triggers kollas och läggs eventuellt i kö. Därefter kastas den delen av datan och nästa del fram till och med den andra *EOT* behandlas, och så vidare.

Slutligen, om både *SOT* och *EOT* är angivna kommer mottagaren att börja buffra data när den har tagit emot *SOT* och därefter samlar den på sig data ända till ett *EOT* dyker upp. Först då kommer mottagaren att gå igenom sin lista av triggers. För övrigt fungerar det på samma sätt som beskrivits ovan.

På följande sidor beskriver vi exempel på hur man kan utnyttja dessa mottagare i Ethisis. Vi förutsätter här att du känner till hur man hanterar mottagarna i Admin. För mer information om hur man skapar och hanterar mottagare hänvisas till manualen *Admin – Konfiguration för Ethisis*.

5.2 Mottagartyper

5.2.1 TCP-inkommande

Exempel på denna typ av mottagare finns beskriven i manualen *Admin – Konfiguration för Ethisis*.

5.2.2 TCP-utgående

Denna mottagare fungerar på så sätt att Ethisis Server tar initiativ till kommunikationen med den externa enheten. Ethisis förväntar sig sedan att den externa enheten spontant skickar data.

För att detta skall fungera måste den externa enheten lyssna på en specifik TCP-port och eventuella brandväggar i den externa enheten måste tillåta inkommande anslutningar på den aktuella porten.

I detta exempel använder vi en UDP-kamera som kommer att skicka notifieringar till Ethisis Server när den detekterar rörelse.

Konfigurera kameran

Det första som måste göras är att konfigurera kameran så att den lyssnar på en port.

Surfa till kamerans webbgränssnitt, logga in och klicka sedan på *Setup*.

IPE3500M

VIEW | SETUP

Basic Configuration
Video & Audio
Event Configuration
DI
DO
Video
Motion Detection
VCA
Network
Health
System Options
IO Configuration
Maintenance
VCA
Motion Detect
Activation
About

Publisher - Motion Detection

Subscriber - DO
All Zone DO #1

Subscriber - Email
 Post notification message Attach a [snapshot](#)

Subscriber - Multicast
 Post notification message

Subscriber - TCP Post notification message

Subscriber - HTTP
 HTTP Server # 1 HTTP Server # 2
 HTTP Server # 3 HTTP Server # 4

Subscriber - FTP
 Post notification message

Apply Reset

v1.09.12

Figur 5.2 Webbgränssnitt i UDP-kameran

Navigera till *Event Configuration* och sedan *Motion Detection* i kamerans menyer. Klicka sedan på *TCP* för att konfigurera kameran till att skicka *TCP Events*. Följande sida visas.

TCP Setting

Configuration

Enable TCP Event

Name : Tcp Event

Listen Port : 1234 (1 ~ 65535)

OK Cancel

v1.07.03

Figur 5.3 Inställning för TCP Event i kameran

Klicka i *EnableTCP Event* och ange den port kameran skall lyssna efter anslutningar på. I detta fall skall kameran lyssna på port 1234.

Klicka OK och klicka sedan i checkrutan *Post notification message* under *Subscriber - TCP*. Avsluta med att klicka *Apply* för att spara ändringarna i kameran.

Subscriber - TCP

Post notification message

Subscriber - HTTP

HTTP Server # 1 HTTP Server # 2

HTTP Server # 3 HTTP Server # 4

Subscriber - FTP

Post notification message

Apply **Reset**

Figur 5.4 Konfiguration av notifiering.

Nu är det dags att konfigurera själva rörelsedetektionen i kameran.

Navigera till *Motion Detect* och sedan *Zones and Rules* i kamerans webbgränssnitt.

Motion Detect

Use	Sensitivity	ObjectSize	Location (x1;y1...x2;y2)	Sensitivity	ObjectSize	Apply
<input checked="" type="checkbox"/> Zone 1	31	32	24-41,9955-41,9955-9957,24-9957	←	←	→ apply
<input type="checkbox"/> Zone 2	32	32	2699-200,4998-200,4998-4999,2699-4999	←	←	→ apply
<input type="checkbox"/> Zone 3	32	32	5198-200,7497-200,7497-4999,5198-4999	←	←	→ apply
<input type="checkbox"/> Zone 4	32	32	7697-200,9996-200,9996-4999,7697-4999	←	←	→ apply
<input type="checkbox"/> Zone 5	32	32	200-5199,2499-5199,2499-9998,200-9998	←	←	→ apply
<input type="checkbox"/> Zone 6	32	32	2699-5199,4998-5199,4998-9998,2699-9998	←	←	→ apply
<input type="checkbox"/> Zone 7	32	32	5198-5199,7497-5199,7497-9998,5198-9998	←	←	→ apply
<input type="checkbox"/> Zone 8	32	32	7697-5199,9996-5199,9996-9998,7697-9998	←	←	→ apply

Figur 5.5 Konfiguration av Motion Detection i kameran

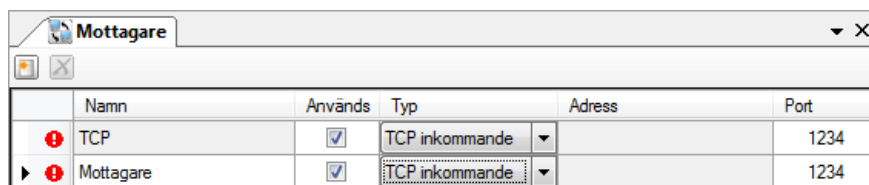
I detta fall använder vi bara en zon (Z1) i kameran och den täcker hela kamerabilden (den gröna ytan). I diagrammet till höger ser man mängden rörelse den senaste tiden. Rött innebär att mängden rörelse varit över inställd trignivå. Kameran kommer i det fallet att skicka en notifiering till Ethirisservern.

Konfigurationen i kameran är nu klar, vi går vidare till att konfigurera mottagaren i Ethiris Admin.

Konfigurera mottagare i Ethiris Admin

Dubbelklicka på noden *Mottagare* under *Kommunikation* i trädvyn.

Klicka sedan på *Ny mottagare*. Du ser då något som liknar bilden i Figur 5.6.

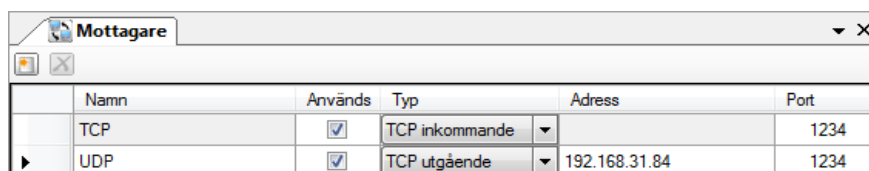


	Namn	Används	Typ	Adress	Port
❗	TCP	<input checked="" type="checkbox"/>	TCP inkommande		1234
▶	Mottagare	<input checked="" type="checkbox"/>	TCP inkommande		1234

Figur 5.6 Panelen Mottagare.

Ändra namnet på mottagaren till UDP och typen till *TCP utgående* eftersom det är Ethisis Server som skall ta initiativet till kommunikationen. När typen ändras till *TCP utgående* får du även möjlighet att ange *Adress* till den externa enheten. I detta fall anger vi där IP-adressen till UDP-kameran, 192.168.31.84. *Port* skall stå kvar på 1234 eftersom det var den porten vi konfigurerade i kameran, se Figur 5.3.

När du är färdig ser det ut något i stil med detta.

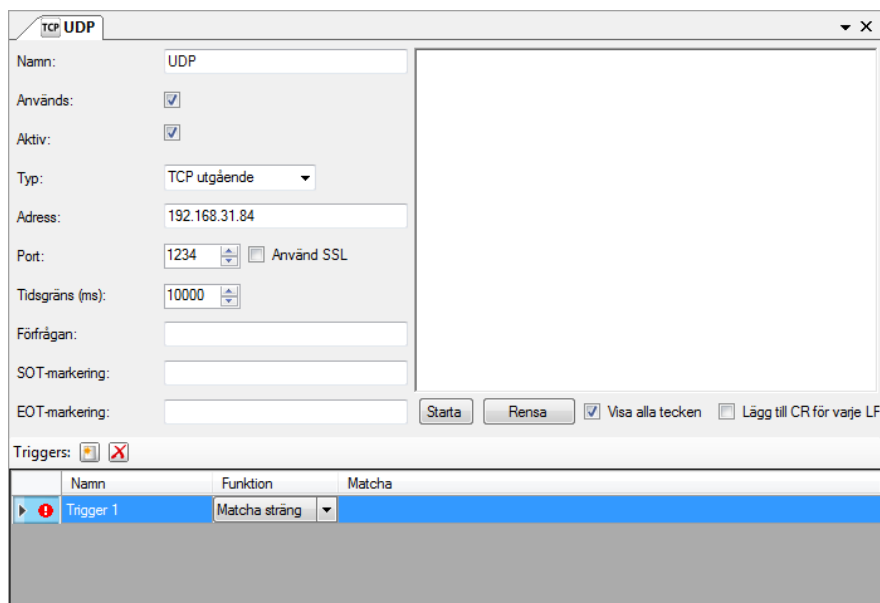


	Namn	Används	Typ	Adress	Port
	TCP	<input checked="" type="checkbox"/>	TCP inkommande		1234
▶	UDP	<input checked="" type="checkbox"/>	TCP utgående	192.168.31.84	1234

Figur 5.7 Panelen Mottagare med typ ändrad för den nya mottagaren.

Observera felmarkeringen som visas i Figur 5.6. Detta beror på att två mottagare är konfigurerade som *inkommande* anslutning mot samma port. Detta kommer inte att fungera. När typen för den nya mottagaren sedan ändras till *utgående* försvinner felmarkeringen igen.

Dubbeltklicka nu på den nya mottagaren i trädvyn till vänster. Mottagarens panel visas då.



Namn: UDP

Används:

Aktiv:

Typ: TCP utgående

Adress: 192.168.31.84

Port: 1234 Använd SSL

Tidsgräns (ms): 10000

Föfrågan:

SOT-markering:

EOT-markering:

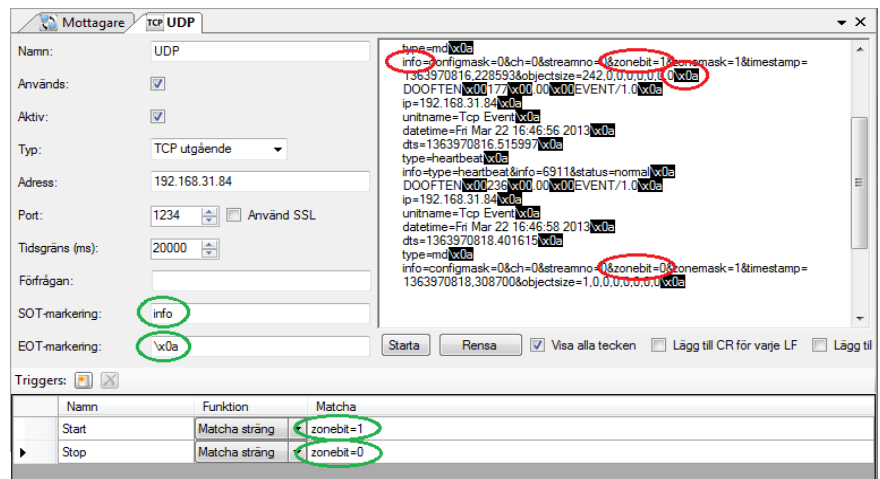
Starta Rensa Visa alla tecken Lägg till CR för varje LF

Triggers:

	Namn	Funktion	Matcha
▶	Trigger 1	Matcha sträng	

Figur 5.8 Panelen UDP

Nu är det dags att göra den slutliga konfigurationen av mottagaren. En hjälp är att starta monitorering av trafiken mellan kameran och Ethisis Server. I trafiken finns de nyckelord vi skall identifiera för att få mottagaren att jobba optimalt. Titta noga i textflödet från kameran och identifierade texter vi skall använda som SOT, EOT, föra att starta inspelning och för att sluta inspelning.



Figur 5.9 Panelen UDP med identifierade nyckeltexter

I textflödet från kameran har de aktuella nyckeltexterna identifierats och markerats med röda ellipser. Nyckeltexterna är sedan införda i respektive fält i mottagarens konfiguration.

Texten "info" inleder varje rad text från kameran där sedan nyckelorden för start respektive stopp av rörelse finns. Därför anger vi "info" som SOT markering för mottagaren.

På motsvarande sätt identifierar vi tecknet '\x0a' som sista tecken i varje rad med information från kameran där nyckelorden för start respektive stopp av rörelse sedan finns. Vi anger alltså '\x0a' som EOT markering för mottagaren.

En liten detalj, som kan vara nog så viktig är inställningen för *Tidsgräns (ms)*. Kameran skickar automatiskt en heartbeat-notifiering ungefär var 15e sekund. Detta för att säkerställa att kommunikationen är uppe och fungerar med kameran. För att inte Ethiris skall larma om kommunikationsfel med kameran, måste vi därför ställa tidsgränsen på 20s. Det fungerar så att om Ethiris inte får några data via den här mottagaren inom inställd tidsgräns, kommer Ethiris att aktivera larmet *Kommunikationsfel med mottagare*.

Nu är det dags att identifiera nyckelorden för de triggers vi skall använda för att starta och stoppa inspelning i Ethiris.

I vårt fall har vi bara en zon definierad i kameran, varför vi bara behöver använda texten 'zonebit=1' som trigger för att rörelse har startat. På motsvarande sätt använder vi 'zonebit=0' som trigger för att rörelse har slutat i kameran.

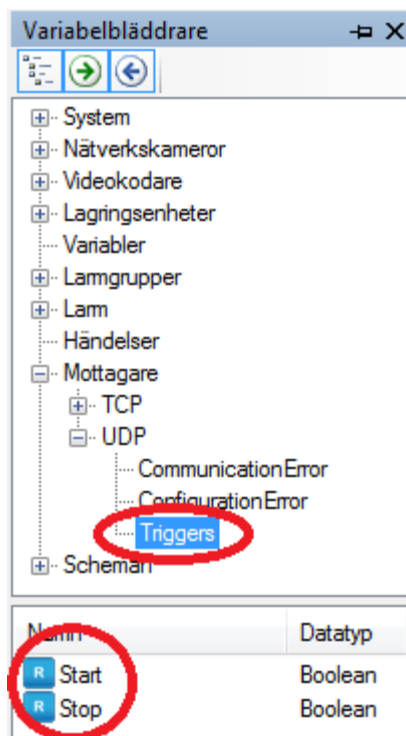
Vi skapar 2 triggers med dessa nyckelord att matcha.

Skript i Ethiris Admin

Sista steget är att knäpa ihop den rad skriptkod som koppar ihop mottagaren med kameran och startar/stoppas inspelning.

Kameran vi vill spela in från heter i vårt exempel UDPCam.

De två triggerarna dyker upp som variabler av typen *Boolean* i variabelbläddraren i skriptpanelen.



Figur 5.10 De nya triggervariablerna är tillgängliga i variabelbläddraren.

Med hjälp av dessa kan vi skriva ett litet skript för att starta inspelning på UDP-kameran när det rör sig framför den.

Med följande skript löser vi detta:

```
UDPCam.RecordEvent = (UDPCam.RecordEvent || UDP.Start) && !UDP.Stop;
```

Eftersom triggervariablerna endast är *true* under ett skriptvarv behöver vi denna konstruktion där *RecordEvent*-variabeln håller sig själv ända tills vi får stoppsignalen i form av att *UDP.Stop* blir *true*.

|| betyder "eller".

! betyder "inte".

6 Förklaring av termer

ActiveX

ActiveX är ett ramverk för att skapa återanvändningsbara mjukvarukomponenter i ett programmeringsspråksberoende sätt. Mjukvaruapplikationer kan då vara sammansatta av en eller flera av dessa komponenter för att uppnå önskad funktionalitet.

COM

Component Object Model. En binär kommunikationsstandard från Microsoft som användas för att kommunicera mellan olika programkomponenter. COM utgör grunden i OPC.

IP adress

Varje enhet i ett datornätverk har en unik IP-adress som består av 4 siffergrupper där varje grupp kan vara mellan 0-255, t ex 192.168.30.29.

Licensnyckel

En kodnyckel som erhålls från Kentima efter fullgjord betalning och registreringskod har skickats in. Licensnyckeln används för att "låsa upp" Ethisis via licensdialogen.

Nätverkskamera

Typ av kamera som kan kopplas in direkt i ett nätverk. Kameraenheten har en egen unik IP-adress.

OPC

OLE for Process Control. En kommunikationsstandard som har tagits fram inom automationsindustrin för kommunikation mellan olika system.

Produktkod

En kodnyckel som följer med varje Ethisislicens. Denna ska matas in vid installation av Ethisis. Ethisis server behöver en produktkod vid installation.

Registreringskod

Den kodnyckel som unikt identifierar en programinstans. Kodnyckeln genereras vid klick på knappen *Registrera* i licensdialogen. Denna kodnyckel ska skickas till Kentima AB vid upplåsning av Ethisis.

TCP/IP

Transmission Control Protocol/Internet Protocol.

Syfte 1:1

7 Index

A

ActiveX 2:1

E

Exempel

ActiveX 2:3

Fjärrstyrning 3:2

F

Fjärrstyrning 3:1

G

Gränssnitt

ActiveX 2:7

Fjärrstyrning 3:9

I

Installation

ActiveX 2:2

Fjärrstyrning 3:1

Modbus OPC Server 4:1

K

Konfiguration

Modbus OPC Server 4:1

M

Modbus OPC Server 4:1

Mottagare 5:1

Licenskrav 5:1

TCP-inkommande 5:3

TCP-utgående 5:4

Typer 5:2

S

Skalbarhet 1:2

KENTIMA PRODUCT LINES

SECURITY

VIDEO MANAGEMENT SOFTWARE

NETWORK VIDEO RECORDER

PSIM SOFTWARE

AUTOMATION

HMI/SCADA SOFTWARE

INDUSTRIAL COMPUTERS

OPERATOR PANELS

Postadress:

BOX 174
SE-245 22 Staffanstorp
Sweden

Tel: +46 (0)46-25 30 40

E-mail: info@kentima.com

Besöksadress:

Kastanjevägen 2
245 44 Staffanstorp
Sweden

Fax: +46 (0)46-25 03 10

www.kentima.com

