

---

# MOBOTIX MxServer: System Overview

MOBOTIX AG  
Software Department

August 6, 2008

## Abstract

The MOBOTIX MxServer has been developed primarily to integrate analog video cameras (PAL/NTSC/SECAM) and non-MOBOTIX IP cameras into a computer network as if they were MOBOTIX *digital* cameras. A user working with MOBOTIX MxControlCenter will be able to connect and view the analog cameras in the same way as digital MOBOTIX cameras. IP cameras can also be integrated into the complete MOBOTIX system. In this case, MxServer will store their M-JPEG streams to a file server and MxControlCenter can directly display their video streams.

MxServer offers these features:

- MxServer uses an optional FalconQuattro video grabber card to digitize analog video inputs to M-JPEG streams.
- MxServer can permanently save the digitized video streams on the hard disk of a file server to allow off-line video analysis.
- MxServer can control one or more pan/tilt heads, such as the BOSCH AutoDome or the Siemens SpeedDome; more systems are being added.
- MxServer can retrieve digital video streams from IP cameras or DVRs from third-party manufacturers and save them on a file server in the MOBOTIX specific folder and file structure.

---

---

### **Document Information**

---

Document name: MxServer.tex  
Document version: 10590 (tsc)  
Document date: 2008-08-06  
State: Final  
Author: Giorgio Dal Molin  
MxServer version: 1.2.1  
Confidentiality: Public

---

## **Contact Information**

MOBOTIX AG  
Security Vision Systems  
Luxemburger Str. 6  
67657 Kaiserslautern, GERMANY

eMail: [sales@mobotix.com](mailto:sales@mobotix.com)  
Web: <http://www.mobotix.com>  
phone: +49 (631) 3033-103  
fax: +49 (631) 3033-190

## Contents

<b>1</b>	<b>Description of the System</b>	<b>4</b>
<b>2</b>	<b>System Components</b>	<b>4</b>
2.1	Analog Video Cameras . . . . .	4
2.2	IDS FalconQuattro Video Grabber Card . . . . .	4
2.3	MxServer Computer . . . . .	6
2.4	File Server . . . . .	7
2.5	Pan/Tilt Heads . . . . .	7
2.6	The MxServer Package . . . . .	8
2.6.1	MxServer_XX_Pn.exe . . . . .	8
2.6.2	MxServer Configuration . . . . .	10
2.7	HTTP Control API . . . . .	10
2.7.1	Example 1: . . . . .	13
2.7.2	Example 2: . . . . .	13
2.7.3	Example 3: . . . . .	14
<b>3</b>	<b>Restrictions/Problems</b>	<b>15</b>
3.1	Frame Clipping or Scaling . . . . .	15
3.2	Firewall Problems . . . . .	16
3.3	Frame Rate Restrictions . . . . .	16
<b>4</b>	<b>System Requirements</b>	<b>16</b>
<b>5</b>	<b>Troubleshooting</b>	<b>17</b>

## 1 Description of the System

Figure 1 shows the structure of the MxServer system.

The analog cameras are connected directly to an IDS FalconQuattro video grabber card installed in a computer that is acting as a video server. The FalconQuattro card has four analog video input channels. Its main task is to convert the received analog video frames into equivalent *digital* bitmaps and to transfer them to the server's RAM. The FalconQuattro card is controlled by MxServer that transforms the bitmaps provided by the card into JPEG-encoded images and sends them either to the video clients (i.e. MxControlCenter) connected to MxServer through the network or to the hard disk of the file server.

The computer running MxServer can handle as many clients at the same time as the operating system permits, provided that the network bandwidth is sufficient to service all clients.

It is possible to install more than one FalconQuattro card in the same server to have more than four analog video inputs available. It should be noted, however, that adding another FalconQuattro card will reduce the maximum frame rate for each channel.

MxServer also supports certain third-party IP cameras/VDRs as additional video channels:

- Axis
- Sony
- Toshiba
- Merit Li-Lin
- Asip

The video streams from these network devices can only be saved on the disk of the file server and can be used for offline video analysis.

## 2 System Components

### 2.1 Analog Video Cameras

Each camera must have its own power supply and its video cable has to be connected to one of the input ports of the FalconQuattro card in the file server. The grabber card supports PAL, NTSC and SECAM video cameras.

### 2.2 IDS FalconQuattro Video Grabber Card

The FalconQuattro card must be properly installed on the server computer. The card has to be inserted into a free PCI slot on the motherboard and the software

2.2 *IDS FalconQuattro Video Grabber Card*2 SYSTEM COMPONENTS

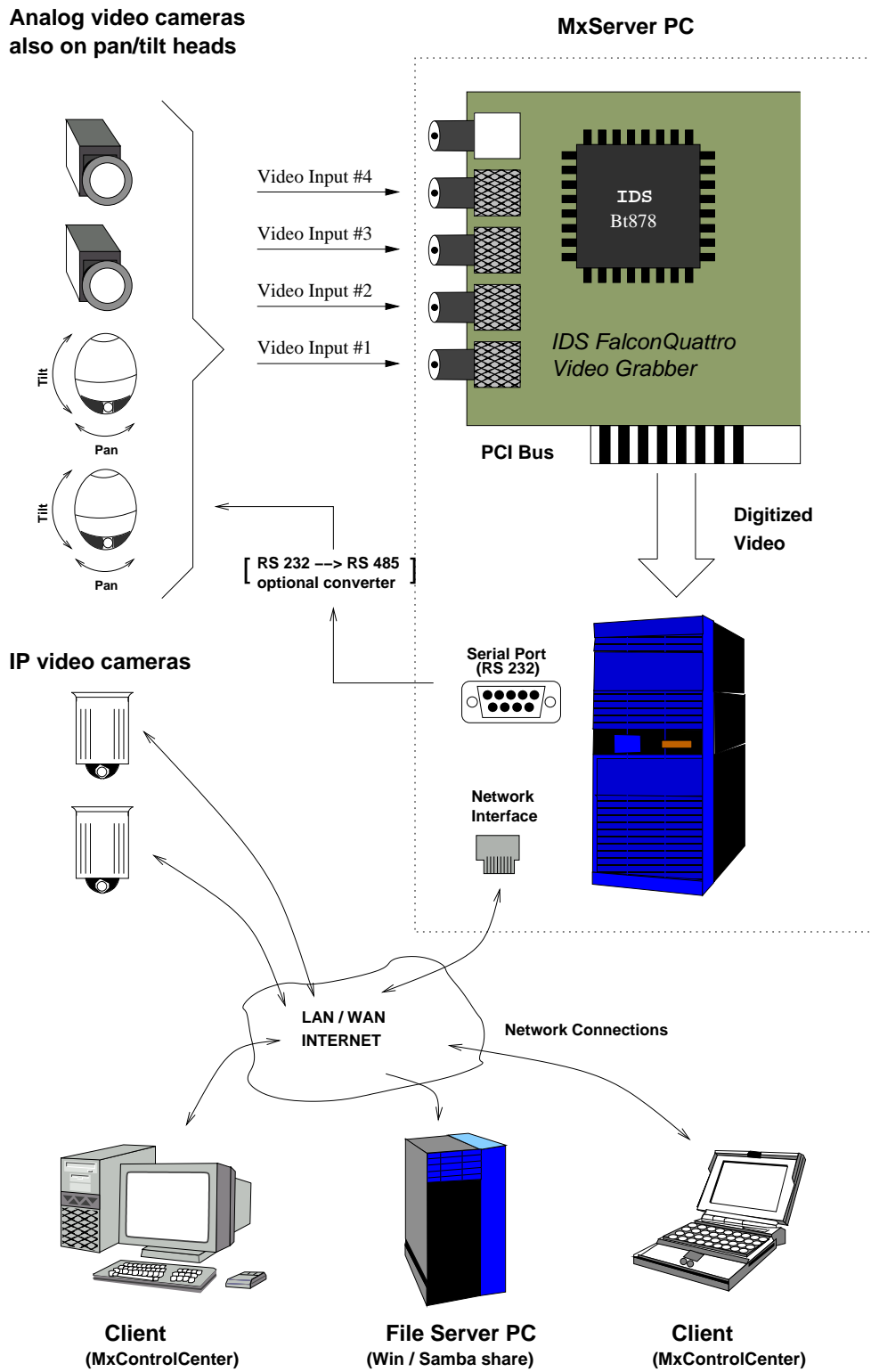


Figure 1: System Overview using MxServer

driver has to be installed from the IDS CD ROM (the installation directory also contains a link to the latest driver installation file).

The following software is required:

- The video and audio drivers located at `<CDROM>\Drivers\<Windows version>`. For Windows 2000 and XP, these would be:

```
<CDROM>\Drivers\Win2000_XP\falcon.sys
```

and

```
<CDROM>\Drivers\Win2000_XP\falcaud.sys .
```

- The IDS utility applications: `DirectDrawInfo`, `IdsID`, `IdsInfo`.

The IDS driver CD ROM has an installation wizard that automatically starts after the CD ROM has been inserted. Select the “User defined” installation option and click on the “Next” button to proceed. Make sure that at least the “Tools” and the “Drivers” checkboxes are selected and proceed to install the files. A system reboot will be required at the end of this procedure.

If the installation wizard fails and Windows does *not* recognize the grabber card, you should try installing the drivers manually. It has to be repeated four times for the video driver and four times for the audio driver (once for each input channel of the card). In case Windows complains that the drivers are not “Microsoft-certified”, proceed anyway.

Once the drivers have been properly installed and the operating system recognizes the grabber card, you will need to assign a unique ID to each of the four FalconQuattro input channels. Start the `IdsID` software utility you just installed and set a unique ID (from 1 to 254) to each of the input channels of the grabber card. It does not matter which ID number you assign to each channel as long as they are all different. In most cases the best choice is to simply assign the IDs 1, 2, 3, 4 to the four input channel of the first card and 5, 6, 7, 8 for a second FalconQuattro card and so on ...

It is possible to install more than one FalconQuattro on the same server, as long as there are free PCI slots on the motherboard. Installing more than one card in one computer, however, (i.e. more than four input channels) will result in a reduced frame rate for each channel (mainly due to restrictions of the PCI bus). If you are planning larger installations, you should use more computers with one FalconQuattro card each to get optimum performance.

## 2.3 MxServer Computer

The MxServer computer hosts the video grabber card(s) and runs the MxServer application. The computer should be at least a Pentium 4 class computer with a minimum of 512 MB RAM. No monitor or keyboard is required as long as the

software is installed and MxServer is started automatically after the system has booted (using a Scheduled Task, for example).

In most cases, the server is part of a LAN so that the clients can reach it from the Intranet/Internet. The computer should run Microsoft Windows 2000/XP Professional or Windows 2000 Server (or higher), if more than ten simultaneous network connections to the server are required.

## 2.4 File Server

As an additional feature, MxServer can save the video streams from the analog/IP channels to the hard disk of a file server (see figure 1). This computer can be the MxServer computer itself, acting also as file server or it can be a different one, reachable from the MxServer computer via the network.

The file server has to provide an accessible share where MxServer can save the video streams. Each channel's video stream can be saved to a different file server.

The video streams saved on the file server can be accessed and reviewed any time using MxControlCenter.

## 2.5 Pan/Tilt Heads

MxServer can also control one or more optional pan/tilt heads, to let the user change the direction to which the video camera points.

MxServer currently supports two different pan/tilt systems:

- BOSCH AutoDome
- Siemens SpeedDome

These pan/tilt systems are connected to the server via the serial port (COMx).

The BOSCH AutoDome system supports both the RS232 and the RS485 serial standards. Up to 254 AutoDomes can be connected to the same serial port, provided that each one has a different ID, which can be set using a set of switches on the AutoDome itself.

The Siemens SpeedDome system supports only the RS485 serial standard and will generally require an additional RS232-to-RS485 signal converter. Up to 100 SpeedDomes can be connected to the same serial port, provided that each one has a different ID, which can be set using a set of switches on the SpeedDome itself.

The maximum cable length for an RS232 serial port cable should not exceed 15~20 m. In case a longer cable is needed, an RS232-to-RS485 converter should be installed. Make sure that you do not exceed the maximum cable lengths as specified by the manufacturer of the RS232-to-RS485 converter.

## 2.6 The MxServer Package

The MOBOTIX MxServer software package is the core software component of the system. Download the correct file for your hardware platform:

- The P3 release has been optimized for Pentium 3 with SSEI and AMD machines.
- The P4 release is optimized for Pentium 4 with SSEII.

Download the appropriate file for your hardware, then start the installation wizard by executing the `MxServer-<version>_Px.msi` Windows installer file, then follow the instructions.

The default installation directory `c:\<programs folder>\MOBOTIX\MxServer` should be a reasonable choice for most computers.

The MxServer package installs the following files:

- `MxServer_XX_Pn.exe` is the executable (where "XX" represents the language and "n" is either 3 or 4),
- `readme_en.txt` is a text file with a quick overview on the system,
- `MxServer.ico` is the official MxServer program icon,
- `mx_manual_mxserver_en.pdf` is this file.

### 2.6.1 MxServer\_XX\_Pn.exe

This is the application that implements the MxServer functionality. It runs on the MxServer computer. On one hand it listens for incoming connection requests from the clients on the network and accepts them, on the other hand it retrieves and encodes the images from the input video channels of the FalconQuattro card(s) and then sends the encoded images to the clients.

MxServer should be started at boot time as a Windows *Scheduled Task* in a production environment; doing so will properly start MxServer without any user interaction.

The program takes two *optional* command line parameters as shown in the following example call:

```
> MxServer_XX_Pn 0.0.0.0 80
```

("XX" represents the language and "n" is either 3 or 4)

The first parameter is the IP address on which the *configuration* socket should listen for incoming connections, the second is the first port used to access the browser-based configuration page of MxServer (see section 2.6.2). If no arguments are provided, the application will search the Windows registry for saved values from the last session. If no values are found, the default values will be used: 0.0.0.0 (any IP address configured on the local host) as IP address and 80 as port.

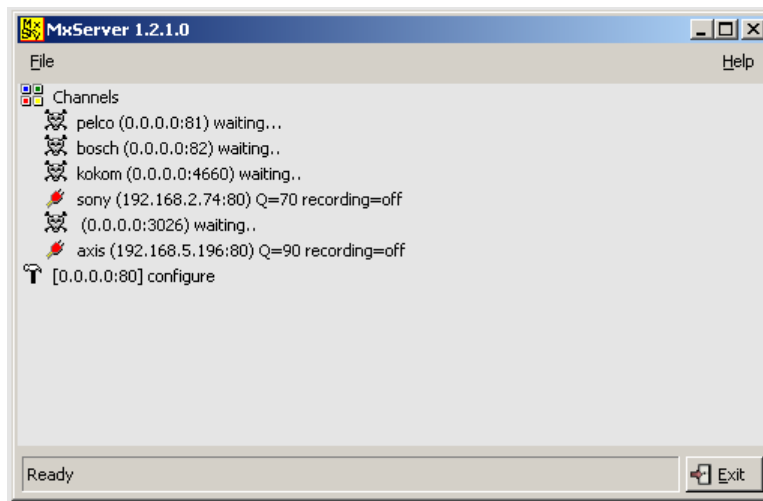


Figure 2: MxServer window

If you would like to set a specific IP address for MxServer to listen on (eg if the computer has two network interface cards), you can set this address as in the following example:

```
> MxServer_XX_Pn 192.168.23.125 80
```

**Caution!** If the IP port is already used by another service, e.g. a web server running on port 80), you should choose a different port for the MxServer or stop the other service and restart MxServer.

Once started, MxServer opens a listening socket for its integrated browser-based configuration page and scans the system to activate the video channels that are available (four for each configured FalconQuattro card). For every channel that has been found, a listening server is opened on the configured IP address:port for that channel (see section 2.6.2).

The first time MxServer runs, the Falcon analog channels are not configured yet and MxServer will use default values for the IP address and port:

- MxServer browser-based configuration page → port 80,
- Falcon analog channels → port <free port from the system>

By reconfiguring MxServer (see section 2.6.2), it is possible to assign a specific IP:port address to each channel.

### 2.6.2 MxServer Configuration

To configure MxServer, any web browser (even text-based web browsers such as `lynx`) will work. The MxServer application window (see figure 2) displays a little hammer icon followed by the `IP address:port`. This address and the port correspond to the command line parameters given when starting MxServer (eg `192.168.23.125 80` as in the example in section 2.6.1).

If the MxServer application window shows the IP address and port behind the hammer icon as `0.0.0.0 80`, the computer will listen to any of its IP addresses. In this case, you will have to obtain the IP address of this computer using the command shell and the `ipconfig /all` command.

If you are directly logged in the computer where the MxServer runs just click on the menu command `File→Configure` on the MxServer GUI. If you are logged in somewhere else on the net then manually open a web browser and enter the URL of the computer running MxServer and the corresponding port:

```
<MxServer IP address>:<port>
```

The browser displays an authentication request window. Enter the default login user name `admin` and the password `meinsm`. MxServer should respond with an HTML page showing its current configuration. There is an `http` link available for each channel found/configured on the system, a set of links to create and configure new dynamic IP channels and a link to the “System Settings” section containing settings like admin login and password settings, path and maximum size of the MxServer log file and the `IP address:port` of the browser-based configuration page. To update the MxServer configuration, click on the desired link, modify the desired parameters and click on the `Send` button at the end of the HTML page. MxServer replies with a short report, which also includes any problems encountered during the update process.

The configuration parameters are saved in the Windows registry so that they will be available between different MxServer sessions.

## 2.7 HTTP Control API

MxServer can be configured manually using the browser-based configuration page as described in section 2.6.2, but it can also be controlled remotely using the URL/CGI `/submitConfig?` with a set of proper arguments. Sending the proper arguments along with `/submitConfig?` it is possible (for example) to switch the video recording on or off for a particular channel or to change the channel name without manually opening the `/configure` configuration page. Using the `/submitConfig?` it is also possible to change the parameters of different channels *simultaneously*.

A control client must connect to the server using a TCP socket and send either a `GET` or a `POST` HTTP request as shown in the following templates (make sure that you replace the text between the `<...>` brackets by valid HTTP code):

POST control request example:

```
POST /submitConfig HTTP/1.1
<http headers>
Host: <client host name>
Authorization: Basic <usr:pwd>
Content-Length: <post data length>
```

```
<post data: var1=val1&var2=val2&...>
```

GET control request example:

```
GET /submitConfig?var1=val1&var2=val2&... HTTP/1.1
<http headers>
Host: <client host name>
Authorization: Basic <usr:pwd>
```

Note that the GET version is simpler, but will not work if the parameter list is *very* long. The POST version will work with parameter lists of any length.

Tables 2, 3, 4 and 5 describe which *var* are recognized by MxServer and what the *val* values mean.

MxServer supports several video channel types, currently analog and IP channels. Some parameters are common to all channel types, others are type-specific; the parameters with the *sys\_* prefix of table 5 are system-wide parameters.

The *<prefix>* present in tables 2, 3 and 4 is used to identify *exactly one* channel on the system. It has the following syntax:

```
<ch type ID>_<ch index>
```

The *<ch type ID>* value identifies a channel type as specified in table 1. The *<ch index>* is the index of the channel within its type.

<i>Channel type</i>	<i>ch type ID</i>
Analog (IDS Falcon)	0
Axis IP	1
Merit Li-Lin IP	2
Asip IP	3
Toshiba IP	4
Sony IP	5

Table 1: Channel type IDs

<i>var</i>	<i>description</i>	<i>Units</i>
<code>&lt;prefix&gt;_CnlIPAddr</code>	IP address channel server socket	
<code>&lt;prefix&gt;_CnlIPPort</code>	port of the channel server socket	
<code>&lt;prefix&gt;_CnlVideoStandard</code>	video standard	[PAL NTSC SECAM]
<code>&lt;prefix&gt;_ImageQuality</code>	image quality index	[10%-90%]
<code>&lt;prefix&gt;_ImageSize</code>	image size	[vga vga2 cif]
<code>&lt;prefix&gt;_ImageClipScale</code>	clip or scale the frame	[clip scale]
<code>&lt;prefix&gt;=&lt;ch type ID&gt;_&lt;ch index&gt;</code> (see table 1)		

Table 2: HTTP parameters for the analog channels

<i>var</i>	<i>description</i>	<i>Units</i>
<code>&lt;prefix&gt;_Remove</code>	removes the IP channel from the system	
<code>&lt;prefix&gt;_CamIPAddr</code>	ip address of the video network device	
<code>&lt;prefix&gt;_CamIPPort</code>	port of the video network device	
<code>&lt;prefix&gt;_userName</code>	user name for the video network device	
<code>&lt;prefix&gt;_userPwd</code>	password for the video network device	
<code>&lt;prefix&gt;_ImageQuality</code>	image quality index	[10%-90%]
<code>&lt;prefix&gt;_ImageSize</code>	image size	[vga cif]
<code>&lt;prefix&gt;=&lt;ch type ID&gt;_&lt;ch index&gt;</code> (see table 1)		

Table 3: HTTP parameters for the IP channels

<i>var</i>	<i>description</i>	<i>Units</i>
<code>&lt;prefix&gt;_CnlName</code>	channel text label	
<code>&lt;prefix&gt;_CnlEventRecActive</code>	(de)activate the event recording	[on off]
<code>&lt;prefix&gt;_CnlEventRootPath</code>	event recording root path	
<code>&lt;prefix&gt;_CnlEventDiskQuota</code>	disk space for event recording	[Mb]
<code>&lt;prefix&gt;_CnlEventAgeDays</code>	max event age before it is deleted	[days]
<code>&lt;prefix&gt;_CnlEventAgeHours</code>	max event age before it is deleted	[hours]
<code>&lt;prefix&gt;_CnlEventFramePeriod</code>	period between two recorded frames	[ms]
<code>&lt;prefix&gt;_RotorType</code>	pan/tilt type	[speeddome  autodome]
<code>&lt;prefix&gt;_RotorComPort</code>	serial port connected to the pan/tilt head	[COM1,...]
<code>&lt;prefix&gt;_RotorBoudRate</code>	serial port speed	[2400 4800...]
<code>&lt;prefix&gt;_RotorAddress</code>	numerical address of the dome	[0 1 2 ...]
<code>&lt;prefix&gt;=&lt;ch type ID&gt;_&lt;ch index&gt;</code> (see table 1)		

Table 4: HTTP parameters common to all channels

<i>var</i>	<i>description</i>	<i>Units</i>
sys_ConfIPAddr	IP address of the browser-based configuration page	
sys_ConfIPPort	IP address of the browser-based configuration page	
sys_logFilePath	log file path	
sys_logFileSize	log file size	[Mb]
sys_login_<admin name>	if present removes the <admin name> admin	
sys_newLogin	add/update the corresponding admin	
sys_newPasswd1	password for the new/updated admin	
sys_newPasswd2	confirm the sys_newPasswd1	

Table 5: System-wide HTTP parameters

### 2.7.1 Example 1:

This example shows how to activate the recording for the first Falcon channel (<prefix>=0\_1) with a frame rate of 5 fps ( $\rightarrow 1000/5 = 200ms$  frame period). It is assumed that the other recording parameters (root path, event max age, ...) have already been set up properly:

POST version:

```
POST /submitConfig HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46bWVpbmNt
Content-Length: 52
```

```
0_1_CnlEventRecActive=on&0_1_CnlEventFramePeriod=200
```

GET version (must be all on one line):

```
GET /submitConfig?0_1_CnlEventRecActive=on&
0_1_CnlEventFramePeriod=200 HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46bWVpbmNt
```

It is not allowed to insert spaces directly in a GET or POST parameter list, for example to specify a channel name like “Camera side west”. Make sure that you encode the the complete URL according to RFC 1738 before sending it.

### 2.7.2 Example 2:

Change the image parameters for the third analog channel:

POST version:

```
POST /submitConfig HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46bWVpbnNt
Content-Length: 91
```

```
0_3_CnlName=Landscape_nord&0_3_ImageQuality=70
&0_3_ImageClipScale=scale&0_3_ImageSize=vga
```

GET version (must be all on one line):

```
GET /submitConfig?0_3_CnlName=Landscape_nord&0_3_ImageQuality=70&
0_3_ImageClipScale=scale&0_3_ImageSize=vga HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46bWVpbnNt
```

### 2.7.3 Example 3:

Create a new Axis IP channel.

The IP channels are somewhat special because they can be dynamically created and removed from the system.

POST version:

```
POST /submitConfig HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46bWVpbnNt
Content-Length: 187
```

```
newCnl=1&1_9999_CamIPAddr=192.168.1.100&1_9999_userName=root&
1_9999_userPwd=topsecret&1_9999_CnlName=Axis_212&1_9999_ImageQuality=70&
1_9999_CnlEventRootPath=/srv/data&1_9999_ImageSize=vga
```

GET version (must be all on one line):

```
GET
/submitConfig?newCnl=1&1_9999_CamIPAddr=192.168.1.100&1_9999_userName=root&
1_9999_userPwd=topsecret&1_9999_CnlName=Axis_212&1_9999_ImageQuality=70&
1_9999_CnlEventRootPath=/srv/data&1_9999_ImageSize=vga HTTP/1.1
Host: localhost
Authorization: Basic YWRtaW46bWVpbnNt
```

Note the special parameter `newCnl=1`: it is needed to communicate to the MxServer that it should create a *new* channel and that this new channel should be of type Axis (ch type ID 1, see Table 1 for the ID of the other types).

Note also the special prefix of the variables: `1_9999_`. The first 1 refers to the Axis channel type, as shown in the Table 1; the second number, 9999 is a special index reserved for the new channels.

### 3 Restrictions/Problems

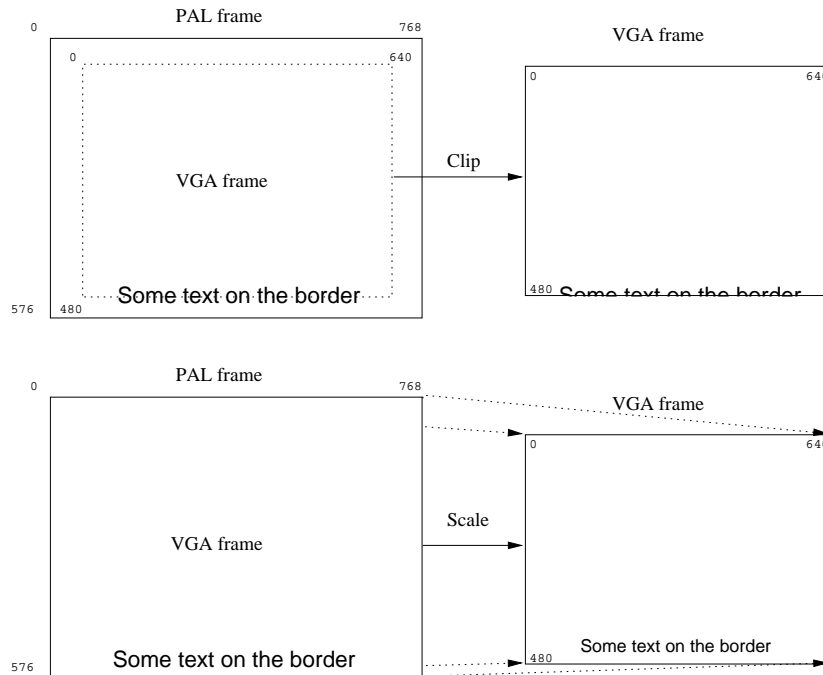


Figure 3: Frame Adaptation Methods

#### 3.1 Frame Clipping or Scaling

The analog video inputs have a fixed image size that depends on the video format, such as PAL video with 768x576 pixels. The user can, however, configure a channel to produce smaller frame sizes like VGA (640x480) or CIF (320x240), to reduce the required network bandwidth. MxServer provides two methods for adjusting the frame size (see figure 3):

1. *Clipping*: the borders of the larger PAL video frames are clipped to fit the selected size.
2. *Scaling*: the larger PAL video frames are scaled down to the selected size using interpolation.

Scaling has the advantage that the whole image area of the original (larger) frame is shown in the reduced user image. The downside is that image quality is also reduced by the interpolation.

Clipping on the other hand produces a frame with the same image quality of the original but without a part of it, namely the borders.

## 3.2 Firewall Problems

Problems can arise if a firewall is active on the server PC. MxServer tries to open a listening server socket for each video channel found on the system, as described in section 2.6.1. The firewall detects the new ports and will pop up a warning message asking you to make a choice: block the program, let the program use the ports only once or let the program always use these ports. When first reading such a message, you should select "Always allow..."

Note that this message will reappear if you change the port for a channel. This may lead to a blocked channel if you change the port remotely without instructing the Windows firewall to allow using this port.

## 3.3 Frame Rate Restrictions

The analog cameras produce video streams with either 25 (PAL) or 30 (NTSC) frames per second (*fps*). With four input channels this would mean a total of  $4 \times 25 = 100$  fps for PAL ( $4 \times 30 = 120$  fps for NTSC). Due to hardware restrictions of the PCI bus, however, it's not possible to digitize and encode all these frames with the full frame rate!

Table 6 shows what can be typically obtained from a Pentium 4-class computer with Hyper Threading.

<i>Analog Channels</i>	<i>fps per channel</i>
1 – 2	25 (30 for NTSC)
3	15 ~ 20
4	10 ~ 15

Table 6: Typical frame rates with a P4 Hyper Threading CPU

## 4 System Requirements

The following list contains a summary of what is required to run a MOBOTIX MxServer system:

- A Pentium 4-class computer with at least 512 MB RAM, a network interface and MS Windows 2000/XP Professional or MS Windows server (2000 or higher) operating system.
- The MOBOTIX MxServer software package installed and running on the server PC.
- Optional: to connect analog cameras you need one or more IDS Falcon-Quattro PCI video grabber cards (product number AA.0010.1.00420.00)

properly installed and configured on the server PC. The IDS FalconQuattro cards require a software driver for the Falcon cards family. You can download the installation file `Falcon_<vers.>.exe` from the same page where the MxServer is or directly from [www.ids-imaging.de](http://www.ids-imaging.de).

If no Falcon card is installed the MxServer will still run but will only support IP cameras.

- Optional: one or more supported pan/tilt heads connected to one of the server's COM ports.

## 5 Troubleshooting

The MOBOTIX MxServer creates a log file named `MxServerLog.log` in the same directory as the MxServer executable and with a maximum size of 5MB.

It is a normal text file and contains a log line for each *event* happened during the past and current session of the MxServer.

A typical content of the log file could be the following:

```
Mon Feb 4 09:31:29 2008 (Info) : main : system : MxServer 1.1.0.1 started.
Mon Feb 4 09:31:29 2008 (Warning) : run : System : Unable to bind the socket descriptor.
Mon Feb 4 09:31:49 2008 (Error) : init : Rec. cli. on cnl. Channel 1 9999 : The Recording root directory path is missing.
Mon Feb 4 09:32:23 2008 (Error) : init : Rec. cli. on cnl. Channel 1 0 : The Recording root directory path is missing.
Mon Feb 4 09:32:23 2008 (Error) : init : Rec. cli. on cnl. Channel 1 9999 : The Recording root directory path is missing.
Mon Feb 4 09:32:23 2008 (Error) : init : Rec. cli. on cnl. Channel 1 0 : The Recording root directory path is missing.
Mon Feb 4 09:32:23 2008 (Error) : init : Rec. cli. on cnl. axis : The Recording root directory path is missing.
Mon Feb 4 09:32:43 2008 (Error) : init : Rec. cli. on cnl. Channel 4 9999 : The Recording root directory path is missing.
Mon Feb 4 09:32:45 2008 (Error) : init : Rec. cli. on cnl. Channel 2 9999 : The Recording root directory path is missing.
Mon Feb 4 09:33:51 2008 (Info) : run : Rec. cli. on cnl. axis : Stopped recording thread.
Mon Feb 4 09:33:52 2008 (Info) : main : system : MxServer 1.1.0.1 terminated.
Mon Feb 4 10:02:51 2008 (Info) : main : system : MxServer 1.1.0.1 started.
Mon Feb 4 10:02:52 2008 (Error) : init : Rec. cli. on cnl. Channel 1 0 : The Recording root directory path is missing.
Mon Feb 4 10:02:52 2008 (Error) : init : Rec. cli. on cnl. axis : The Recording root directory path is missing.
Mon Feb 4 10:02:52 2008 (Warning) : run : System : Unable to bind the socket descriptor.
Mon Feb 4 10:26:03 2008 (Info) : main : system : MxServer 1.1.0.1 started.
Mon Feb 4 10:26:03 2008 (Warning) : run : System : Unable to bind the socket descriptor.
Mon Feb 4 10:26:03 2008 (Error) : init : Rec. cli. on cnl. Channel 1 0 : The Recording root directory path is missing.
...
```

The log file can be opened and reviewed with any normal text viewer/editor and, if the MxServer is running on a host `<srv_hostname>`, it is also available under the url:

`http://<srv_hostname>:<port>/logs`

from a web browser. Opening the url with a browser it is possible to read a log file also on a remote machine.

The log file is also useful when sending bug reports and asking for support.