AXIS 7000 Application Integration using HTTP / FTP

Introduction

This document describes how the AXIS 7000 can be integrated with different kinds of applications using two of the most common TCP/IP protocols, FTP and HTTP. There are two main ways to perform the scanning process using the AXIS 7000 together with an application:

- Pull Type Scanning - initiated from an application and performed using the HTTP protocol.
- Push Type Scanning - initiated from the AXIS 7000 control panel and performed by using the FTP protocol

The document also describes how you can use FTP to obtain and modify a list of available Profiles used for scanning. A Profile defines the resolution, color etc of the scanned image.

HTTP and FTP

HTTP (HyperText Transfer Protocol) and FTP (File Transfer Protocol) are network protocols used for exchanging files (text, graphic images, sound, video and other files) on a TCP/IP based network. HTTP is more advanced and rich in functionality than the more basic FTP.

AXIS 7000 Network Document Server

The AXIS 7000 offers the convenience of allowing end users to distribute and store documents digitally. The AXIS 7000 offers many options to do business in a more effective and cost-efficient way. With a wide variety of formats (TIFF, JPEG/JFIF, PDF, PCL) and transport methods (SMTP, FTP, LPD, Raw TCP) to choose from, an AXIS 7000 provides increased value to digital copiers and scanners. Included in its architecture are embedded Web pages. Administrators can use an existing Web browser for configuration and management purposes, thus eliminating the need to load extra software.

Possible Applications

There are numerous examples of how the Meta Information functionality can be used. Many business processes with documents involved can be improved by adding an AXIS 7000 integrated with an application server. Examples of applications are Knowledge Management Systems, Document Management Systems, Imaging Systems, Enterprise Resource Planning Systems (ERP), Databases and Internet Services. Most application servers connected to a network can be integrated with the AXIS 7000.

http://www.axis.com

Axis Communications (USA)
Tel: (978) 614-2000
Fax: (978) 614-2100

Axis Communications (Europe)
Tel: +46 46 270 18 00
Fax: +46 46 13 61 30

Axis Communications (Japan)
Tel: +81 3 3545 8282
Fax: +81 3 3545 8280

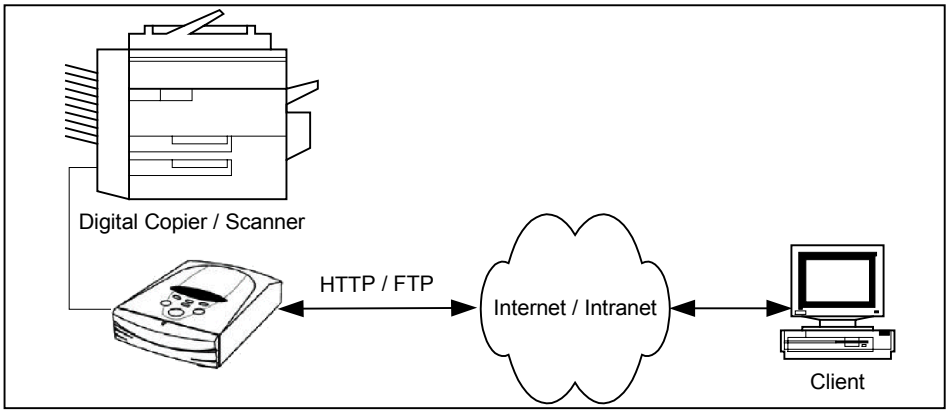
Overview

Figure 1 shows the different components involved when integrating an application running on a client or server together with an AXIS 7000. There are two main ways to perform the scanning process using the AXIS 7000 together with an application:

- Pull Type Scanning - initiated from an application and performed by using the HTTP protocol.
- Push Type Scanning - initiated from the AXIS 7000 control panel and performed by using the FTP protocol

The AXIS 7000 has many different profiles that can be used for different purposes such as scanning text documents, pictures etc. These profiles can be retrieved from the AXIS 7000 and modified using FTP.

Figure 1: Components involved in application integration.



Pull Type Scanning

Pull Type scanning is conducted by sending HTTP commands to the AXIS 7000 over a TCP/IP network. In this section the basic idea is described and then exemplified by showing more specifically how it can be done using the Microsoft Foundation Classes (MFC) in a Windows environment. However, Pull type scanning can be done from almost any environment that has an HTTP client.

Send-to-Web

One of the scanning methods of the AXIS 7000 is Send-to-Web which enables scanning directly into a Web-browser. This is done through the **Scan Document** page of the AXIS 7000 internal web pages by following the steps described below. There are two different approaches. The first is based on starting the scanning directly. The second method reserves the scanner first and then initiates the actual scanning from the AXIS 7000 control panel. The same method as when scanning from a Web-browser will be used when you scan from an application. The application will communicate with the AXIS 7000 using HTTP.

Scan directly:
1. Select a profile from the profile list.
2. Click the **Scan** button.

Reserve:
1. Select a profile from the profile list.
2. Click the **Reserve** button.
3. Initiate the scanning from the AXIS 7000 control panel.

Application Example

To initiate a scanning job from the application a **POST** request is sent to the AXIS 7000. It contains the name of the page, the name of the selected profile and the name of the selected function.
Follow the steps below to scan an image to your application using your AXIS 7000.

1. Send a HTTP request with the following headers:

```
POST /user/scan/scan.shtml HTTP/1.0
Content-type: application/x-www-form-urlencoded
Content-length: 40

page=scan&profileName=Color+low&scanNow=
```

The last row contains the parameters to the AXIS 7000.

page=scan	- necessary for the AXIS 7000 to process the request.
profileName=Color+low	- tells the AXIS 7000 which profile to use.
scanNow	- immediate scan. No value is needed.

A space is not a valid character in an HTTP parameter therefore it is replaced by "+" in the profileName (e.g. "Color low" becomes "Color+low").
If *scanNow* is replaced with *reserve* then the AXIS 7000 and the scanner are reserved, and the user is required to press the **Send** button on the AXIS 7000 control panel. This is the same as clicking the **Reserve** button on the **Scan Document** page.
The AXIS 7000 will respond with either status **302 Moved temporarily** or **200 OK**. If you receive the **200 OK** response then *an error has occurred*; e.g. the scanner could be busy. The reason for using **200 OK** is that the **302 Moved temporarily** message includes a way to send a reference to the image which the **200 OK** do not. The **200 OK** response

contains a complete **Scan Document** page, including an error message. The **302 Moved Temporarily** response means that the scanning has been initiated.

2. The **302 Moved Temporarily** response includes the URL to the scanned image. The syntax is /image.xxx where xxx is one of pdf, jpg, pcl or tif. The table below lists the image name and MIME types for the different image types.

Image Type	Image Name	MIME Type
PDF	image.pdf	Application/pdf
TIFF	image.tif	Image/tiff
JPEG	image.jpg	Image/jpeg
PCL	image.pcl	Image/pcl

```
HTTP/1.0 302 Moved Temporarily
Content-Length: 169
Content-Type: text/html
Location: /image.jpg

<head><title>Moved Temporarily</title></head>
<body><h2>Moved Temporarily!</h2>
        <p>The requested resource has been temporarily moved
        to a new location.
        </p>
</body>
```

Send a GET request with the URL to the AXIS 7000 to retrieve the image data. The request would look something like this:

```
GET /image.jpg HTTP/1.0
Accept: image/jpeg, */*
```

The AXIS 7000 will now respond by sending you a **200 OK** response and the image data.

Error handling

When a request is received the following is verified by the AXIS 7000:

- A scanner is connected and supported.
- The scanner or the AXIS 7000 is not busy.
- The selected profile exists.
- Users are allowed to do Send-to-Web.

When all of these conditions are met the application will receive an HTTP response stating that the document has been temporarily moved, along with the URL, to the image itself. However, if some of the conditions are not met, e.g. the scanner is busy, the response will contain the complete **Scan Document** page, including an error message at the top.

Scanning Using MFC

Microsoft Foundation Classes (MFC) offers good support for Internet connections, especially when it comes to HTTP and FTP. As described above, scanning is performed by sending a **POST** request using HTTP. If the operation is successful, the image is returned via a redirect response. The steps below describe how a scanning job is performed using MFC. If you need more information about the classes described, see the MFC documentation. The following code sample does not perform any error handling, as a real application would do. There is a complete example application for Microsoft Visual C++ available at http://www.axis.com/products/axis_7000/ in the Developer section.

1. Create a CInternetSession object:

```
CInternetSession aSession("ScanSDK");
```

The parameter is a string identifying the application. An application needs only one instance of this class.

2. Using this object the applications can now retrieve an HTTP connection:

```
CHttpConnection* anHttpConnection =
            aSession.GetHttpConnection("171.16.4.140",
                                       (INTERNET_PORT) 80,
                                       NULL, NULL);
```

3. Using this connection, the application can now open a request. By default, redirected responses are handled automatically:

```
CHttpFile* anHttpFile =
                  anHttpConnection->OpenRequest(0,
                  "/user/scan/scan.shtml");
```

The first parameter defines that a POST request shall be opened, the second is the request target.

4. Create a query string containing the parameters:

```
CString aQueryString =
"\n\npage=scan&profileName=Color+low&scanNow=";
```

5. Add an HTTP header which informs the HTTP server that the request contains encoded data:

```
anHttpFile->AddRequestHeaders("Content-type:
application/x-www-form-urlencoded");
```

6. Send the request:

```
anHttpFile->SendRequest(NULL, 0, (void*)(const
char*)aQueryString, aQueryString.GetLength());
```

The method will not return until a response is received from the server or after a time out.

7. If the request is successful, image data will be returned. If not, an HTML page containing error information will be returned. One way of testing what was returned is to look at the Content-type. Its value can be retrieved by the following call:

```
CString aContType;
anHttpFile->QueryInfo(HTTP_QUERY_CONTENT_TYPE,
aContType);
```

If it was successful, aContType will have the value "image/jpeg" (if JPEG is the chosen format. See table above), otherwise "text/html".

8. Finally we are ready to receive the data:

```
UINT nBytesRead = anHttpFile->Read(buffer, MAXBUF - 1);
while( nBytesRead > 0)
{
 //Save data somewhere
.
.
.
//Read next chunk
nBytesRead = anHttpFile->Read(buffer,MAXBUF -1);
}
```

Push Type Scanning

We refer to scanning initiated from the control panel of your AXIS 7000 as Push Type scanning. The images are *pushed out* to a directory on an FTP server.
Your application can be configured to scan the directory on the FTP server. When new images arrive, the application can, depending on information sent with the image, take the appropriate actions.
For each image file written to the directory, an Information File will also be written, unless you choose *Include Image Information: None* when defining the destination. The Information File is in raw text format (ASCII) and contains data about the image.
Information included in the Information File is

- Name and size of the image file.
- Internet address of your AXIS 7000.
- Currently connected scanner.
- The name of the destination (the name appearing in the destination list).
- Profile used.
- Paper size.
- Number of scanned pages.

If you selected *Include Image Information: Complete*, then additional information, such as resolutions and file format, is also included.
Your application can use values in the Information File to trigger actions.

Example of Information File: The following example contains a pre-defined Profile Text. Here, *Image Information Included* is set to *Complete*. If it was set to *Basic,* the Image Information part would not be present.

```
; AXIS 7000 Parameter List, V2.20 May 29 2000

[General]
File name            = image.tif
File size            = 71413
Host address         = 172.19.4.22
Scanner              = FCPA ScanPartner 600C
Destination          = FTP Server
Profile              = Text low
Paper size           = Letter
Number of pages      = 2

[Image Information]
Width                = 1696
Height               = 2200
Resolution (DPI)     = 200
Bits per pixel       = 1
Data type            = Black & white
Paper orientation    = Portrait
Format               = CCITT G.4
Double-sided         = Off
Intensity            = 50
Contrast             = 50

; End of Parameter List
```

The Information File is *always* written *after* the image file so when your application processes the Information File the image file itself is guaranteed to be there already.

Example: A document distributing system may check the file name to see whom the image is targeted for. The file name may have the syntax *name_lastNameXXX.tif*, where XXX is the sequence number optionally appended by your AXIS 7000. Your application can now decide the destination of the image.

Example: A fax server application can use the image name to obtain the fax number, and the file format to decide how to convert the document to fax data.

Please see the AXIS 7000 User's Manual for further information about how to create file destinations.

Accessing the List of Profiles

The profile list is stored in the config.ini file, along with all other parameters. It can be obtained using FTP (please see the *AXIS 7000 Users Manual*). There is a Profiles group, which states the number of available profiles, along with their names. In addition, there is one parameter group for each profile.
To obtain all available profiles you just read the file from your AXIS 7000 and find the group **[Profiles]**. It looks like this:

```
[Profiles]
Clear List First      = no; yes/no
Profiles              = 2
Profile0              = Profile-Text
Profile1              = Profile-Color low
```

The first parameter defines whether to remove all existing profiles before storing the new ones or not. The second parameter defines the number of available profiles, two in this case. The following two lines contain the names of the parameter groups defining the two profiles.
This is a parameter group for a profile:

```
[Profile-Text]
Description           = Text
X-Resolution          = 300
Y-Resolution          = 300
Data type             = Black & white
Paper orientation     = Portrait
Image compression     = CCITT G.4
File format           = TIFF
Double-sided          = Off
Document mode         = Multi page
Intensity             = 50
Contrast              = 50
```

The first parameter is the name of the profile followed by the resolutions, the data type etc.
To add a profile you have to create a parameter group for the profile. You also have to increase the *Profiles* parameter in the *[Profiles]* parameter group and add a parameter in that group defining the name of the new profile.
To remove profiles the *Clear List First* can be used. A config.ini file like the one in the example below will clear all currently stored profiles.

```
[Profiles]
Clear List First      = yes                    ; yes/no
```

Note: No validation of the correctness of the defined profiles will occur when writing the modified config.ini file to your AXIS 7000. Scanning using profiles with incorrect values may result in a corrupt image.

References

http://www.w3.org/Protocols/
http://www.ietf.org/
RFC-1945, "Hypertext Transfer Protocol -- HTTP/1.0"

Axis Communications

Axis is a world leader in the rapidly growing network-attached peripherals industry. Headquartered in Sweden, Axis designs and manufactures network-attached print servers, network document servers, CD/DVD servers, camera servers and storage devices based on its ThinServer™ Technology.

All Axis products leverage Axis' own ThinServer Technology, which allows any peripheral device to be directly attached to the network without a file server or PC. These devices are displayed on any desktop without changing client software and can be accessed and used by multiple clients using virtually any desktop and network operating system. At the core of the ThinServer Technology is embedded software consisting of self-contained, "thin" versions of popular operating systems, a Web server for consistent, network-wide management, and an optimized 32-bit RISC chip complete with device I/O and network controllers for high-speed data transport.