

## How to make Your Own Web pages with FlexWATCH.

This technical note to explain how to customize web page of FlexWATCH System.

It is our recommendation that any one who want to customize Web pager of FlexWATCH system should have following experience to properly achieve his goal.

- 1<sup>st</sup>. You have managed your company or own Web site
- 2<sup>nd</sup>. You know HTML syntax.
- 3<sup>rd</sup>. You can upload files from you PC to server using FTP.
- 4<sup>th</sup>. You can connect to the server through TELNET.
- 5<sup>th</sup>. Optional : You can make Web page using HTML with Java applet.
- 6<sup>th</sup>. Optional : You know JavaScript syntax.

If you don't have experiences like above, you would like to recommend you to lean or acquire similar equivalent experience from other source to achieve your task without problem. We do not explain that here, in this technical note, but we explain differences between general UNIX server and FlexWATCH, and attributes and parameters for FlexWATCH software.

## CHAPTER 1. Build a Web site with FlexWATCH

We provide two ways to watch remote site image. One is Java applet, and the other is plug-in software. As you know Java applet is platform independent. So if you make Web pages using Java applet, the Windows, Mac O/S, UNIX, or Linux users can access your pages with any Web browser. Plug-in is MS-Windows dedicated, but it has more functions than Java applet. With plug-in, the user can save remote site images on his local PC hard drive space and replay the saved video clip later. And the user can save the scenes they want according to motion detection, sensor alarm, or time reservation. Our plug-in works properly on MS IE 4.x and Netscape Communicator 4.x either.

### 1. Web page with Java applet

We provide two types of Java applet. One is for single camera view, and the other is for multi camera view. The name of a Java applet for single-view is FWatch.class(case sensitive). And the name for multi-view is FWM.class(case sensitive). Our Java applet automatically resize images and fit it to applet's window area.

Let me explain how you could embed our Java applets in your Web page, and what attributes and parameters are there for controlling the Java applet.

#### 1) Embedding Java applets in Your Web page

As you know, to embed Java applets to your Web page you must use the <APPLET> tag. In the <APPLET> tag, there are some attributes you must know correctly. The <APPLET> tag has a corresponding ending tag, </APPLET>. Between these two tags, you may use additional <PARAM> tags to specify parameters for the Java applet. The HTML code listed below shows a HTML document for embedding one of our Java applet.

```

<HTML>
<HEAD></HEAD>
<BODY>
<APPLET MAYSCRIPT CODEBASE=/ NAME=FWatch CODE=FWatch.class WIDTH=324 HEIGHT=264>
<PARAM NAME="CamNum" VALUE="1">
<PARAM NAME="CamRun" VALUE="FALSE">
<PARAM NAME="Speed" VALUE="0">
<PARAM NAME="Model" VALUE="FlexWATCH300">
</APPLET>
</BODY>
</HTML>

```

#### \* The <APPLET> tag

As you can see from the above HTML, the <APPLET> tag requires six attributes: **MAYSCRIPT**, **CODEBASE**, **NAME**, **CODE**, **WIDTH**, **HEIGHT**:

```

<APPLET MAYSCRIPT
        CODEBASE=/
        NAME=FWatch
        CODE=FWatch.class
        WIDTH=324
        HEIGHT=264>

```

#### **MAYSCRIPT**

- With this attribute, you can reference a method of the Java applet from JavaScript.

#### **CODEBASE**

- This attribute specifies the path where the Java applet is. You can use both relative and absolute path here. And URL also. Through this attribute, you can link a Java applet to a Web page on other Web server. In this case, the CODEBASE attribute in that Web page would be “[http://ip\\_address\\_of\\_flexwatch/FWatch.class](http://ip_address_of_flexwatch/FWatch.class)” or “[http://ip\\_address\\_of\\_flexwatch/FWM.class](http://ip_address_of_flexwatch/FWM.class)”.

#### **NAME**

- Through the NAME, you can reference this Java applet using JavaScript like “document.applets.NAME”.

#### **CODE**

- This attribute specifies the applet to be embedded on the Web page.

#### **WIDTH**

- This attribute specifies the width of the applet’s window in pixels.

#### **HEIGHT**

- This attribute specifies the height of the applet’s window in pixels.

#### \* The <PARAM> tag

By now you are well aware that Java applets accept parameters from the Web page into which they are embedded. Some parameters are optional, some are not. When parameters are specified by the Web page, they are identified by single <PARAM> tags. Each parameters passed to the Java applet. The <PARAM> tag accepts two attributes of its own: NAME and VALUE:

<PARAM NAME=Java Parameter to change  
VALUE="Java parameter's value">

**NAME**

- A Java applet may accept multiple parameters. To indicate which parameter the Web page is changing, the Web page must specify the parameter's name within the <PARAM> tag.

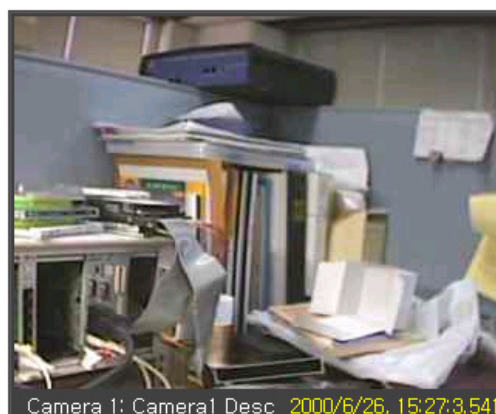
**VALUE**

- Once the parameter to be changed has been identified, the actual value it is to be changed must be passed from the Web page. The value must be enclosed by quotation marks because it is treated as a string by Java.

2) Single-view

This is a sample HTML code with one of our Java applet, FWatch.class for single-view.

```
<HTML>
<HEAD></HEAD>
<BODY>
<APPLET MAYSCRIPT CODEBASE=/ NAME=FWatch CODE=FWatch.class WIDTH=324 HEIGHT=264>
<PARAM NAME="CamNum" VALUE="1">
<PARAM NAME="CamRun" VALUE="FALSE">
<PARAM NAME="Speed" VALUE="0">
<PARAM NAME="Copyright" VALUE="www.flexwatch.com/copyright">
<PARAM NAME="Producer" VALUE="Powered By SEYEON TECH">
<PARAM NAME="Product" VALUE="FlexWATCH - Web Camera Server">
</APPLET>
</BODY>
</HTML>
```

**ATTRIBUTES of <APPLET> tag:****CODE**

- In the case of single-view example, the code would be "FWatch.class". Don't forget it is case sensitive.

**WIDTH**

- FWatch.class has border. It takes 2 pixels per each right & left side. So if you want to show a HxW size image, WIDTH would be W+2+2.

**HEIGHT**

- FWatch.class has border and text area. Border takes 2 pixels per each top & bottom side. And text area takes 20 pixel height. So if you want to show a HxW size image, HEIGHT would be H+20+2+2.

PS) The size of images that FlexWATCH can support has been changed.

**Before kernel version 1.2 (#2k0711)**

W x H  
 192 x 144  
 320 x 240  
 640 x 480

**After kernel version 1.2 (#2k0711)**NTSC mode

W x H  
 176 x 112  
 352 x 240  
 704 x 480

PAL mode

W x H  
 176 x 144  
 352 x 288  
 704 x 576

**PARAMETERS:****CamNum**

- This parameter specifies a camera source. In the case of FlexWATCH 300/500, CamNum is from 1 to 6.

**CamRun**

- This parameter determines the initial state of Java applet. If the value is TRUE, it runs from the beginning. If it is FALSE, at the initial state the Java applet pauses.

**Speed**

- This parameter identifies the delay between images in millisecond scale.

**Copyright**

- The VALUE of this parameter is a key string. Unless you set this parameter exactly same as this sample code, the Java applet will not work. It's case sensitive.

**Producer**

- The VALUE of this parameter is a key string. Unless you set this parameter exactly same as this sample code, the Java applet will not work. It's case sensitive.

**Product**

- The VALUE of this parameter is a key string. Unless you set this parameter exactly same as this sample code, the Java applet will not work. It's case sensitive.

PS)

As you can see from the above image, there is a text area at the lower part of Java applet. It is shown that a brief description about camera source, date, and time. And time value is updated whenever the image is refreshed. The brief description about a camera comes from FlexWATCH. You can set this value through CONSOLE or TELNET using CONFIG command. Refer the user's manual.

PS)

If you need a Java applet without any border and text area, you may use "FWB.class" instead of "FWatch.class". "FWB.class" is a single-view applet and has same parameters as "FWatch.class". Because it does not have any border and text area, WIDTH and HEIGHT are exactly same as the image size that you want to display.

PS)

If you use Java applets from Web page version 1.1, you would better set parameters which is camera description and server model like below list. If you use Java applet from Web page version 1.2 or later, but use kernel before version 1.1, you would better set these parameters either.

```
<HTML>
<HEAD></HEAD>
<BODY>
<APPLET MAYSCRIPT CODEBASE=/ NAME=FWatch CODE=FWatch.class WIDTH=324 HEIGHT=264>
<PARAM NAME="CamNum" VALUE="1">
<PARAM NAME="CamRun" VALUE="FALSE">
<PARAM NAME="Speed" VALUE="0">
<PARAM NAME="Copyright" VALUE="www.flexwatch.com/copyright">
<PARAM NAME="Producer" VALUE="Powered By SEYEON TECH">
<PARAM NAME="Product" VALUE="FlexWATCH - Web Camera Server">

<!-- If you use Java applets from Web page version 1.1, parameters below must be used. -->
<PARAM NAME="CamName1" VALUE="Cam1 Description">
<PARAM NAME="CamName2" VALUE="Cam2 Description">
<PARAM NAME="CamName3" VALUE="Cam3 Description">
<PARAM NAME="CamName4" VALUE="Cam4 Description">
<PARAM NAME="CamName5" VALUE="Cam5 Description">
<PARAM NAME="CamName6" VALUE="Cam6 Description">
<!-- Model - FlexWATCH300, FlexWATCH500, FlexWATCHCAM100, FlexWATCHCAM50 -->
<PARAM NAME="model" VALUE="FlexWATCH300">

</APPLET>
</BODY>
</HTML>
```

### 3) Multi-view

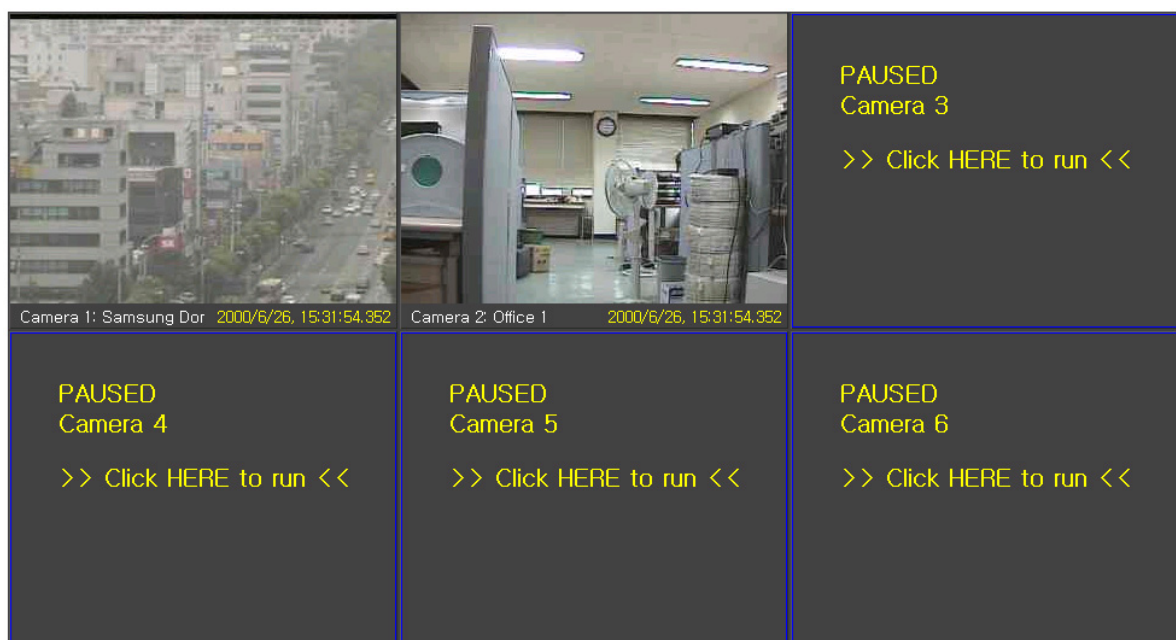
This is a sample HTML code with one of our Java applet, FWM.class for multi-view.

```
<HTML>
```

```

<HEAD></HEAD>
<BODY>
<APPLET MAYSCRIPT CODEBASE=/ NAME=FWM CODE=FWM.class WIDTH=972 HEIGHT=528>
<PARAM NAME="CamRun1" VALUE="FALSE">
<PARAM NAME="CamRun2" VALUE="FALSE">
<PARAM NAME="CamRun3" VALUE="FALSE">
<PARAM NAME="CamRun4" VALUE="FALSE">
<PARAM NAME="CamRun5" VALUE="FALSE">
<PARAM NAME="CamRun6" VALUE="FALSE">
<PARAM NAME="Speed" VALUE="0">
<PARAM NAME="Copyright" VALUE="www.flexwatch.com/copyright">
<PARAM NAME="Producer" VALUE="Powered By SEYEON TECH">
<PARAM NAME="Product" VALUE="FlexWATCH - Web Camera Server">
</APPLET>
</BODY>
</HTML>

```

**ATTRIBUTES:****CODE**

- In the case of multi-view example, the code would be “FWM.class”. Don’t forget it is case sensitive.

**WIDTH**

- FWM.class has border. It takes 2 pixels per each right & left side. So if you want to show a 320x240 size image, WIDTH would be  $972 = (320+2+2) \times 3$ .

**HEIGHT**

- FWM.class has border and text area. Border takes 2 pixel per each top & bottom side. And textarea takes 20 pixel height. So if you want to show a 320x240 size image, HEIGHT would be  $528 = (240+20+2+2) \times 2$ .

**PARAMETER:****CamRunX**

- This parameter determine the initial state of Java applet. If the value is TRUE, it runs from the beginning. If it is FALSE, at the initial state the Java applet would be stopped. X means camera source.

**Speed**

- This parameter identify the delay between images in millisecond scale.

**Copyright**

- The VALUE of this parameter is a key string. Unless you set this parameter exactly same as this sample code, the Java applet will not work. It's case sensitive.

**Producer**

- The VALUE of this parameter is a key string. Unless you set this parameter exactly same as this sample code, the Java applet will not work. It's case sensitive.

**Product**

- The VALUE of this parameter is a key string. Unless you set this parameter exactly same as this sample code, the Java applet will not work. It's case sensitive.

PS)

As you can see from the above image, there is a text area at the lower part of Java applet. It is shown that a brief description about camera source, date, and time. And time value is updated whenever the image is refreshed. The brief description about a camera comes from FlexWATCH. You can set this value through CONSOLE or TELNET using CONFIG command. Refer the user's manual.

PS)

If you use Java applets from Web page version 1.1, you would better set parameters which is camera description and server model like below list. If you use Java applet from Web page version 1.2 or later, but use kernel before version 1.1, you would better set these parameters either.

```
<HTML>
<HEAD></HEAD>
<BODY>
<APPLET MAYSCRIPT CODEBASE=/ NAME=FWM CODE=FWM.class WIDTH=972 HEIGHT=528>
<PARAM NAME="CamRun1" VALUE="FALSE">
<PARAM NAME="CamRun2" VALUE="FALSE">
<PARAM NAME="CamRun3" VALUE="FALSE">
<PARAM NAME="CamRun4" VALUE="FALSE">
<PARAM NAME="CamRun5" VALUE="FALSE">
<PARAM NAME="CamRun6" VALUE="FALSE">
<PARAM NAME="Speed" VALUE="0">
<PARAM NAME="Copyright" VALUE="www.flexwatch.com/copyright">
<PARAM NAME="Producer" VALUE="Powered By SEYEON TECH">
<PARAM NAME="Product" VALUE="FlexWATCH - Web Camera Server">

<!-- If you use Java applets from Web page version 1.1, parameters below must be used. -->
<PARAM NAME="screenCount" VALUE="6">
<PARAM NAME="CamName1" VALUE="Camera1 Description">
<PARAM NAME="CamName2" VALUE="Camera2 Description">
<PARAM NAME="CamName3" VALUE="Camera3 Description">
<PARAM NAME="CamName4" VALUE="Camera4 Description">
<PARAM NAME="CamName5" VALUE="Camera5 Description">
<PARAM NAME="CamName6" VALUE="Camera6 Description">
<!-- If you use Java applets from Web page version 1.1, parameters below must be used. -->
<PARAM NAME="model" VALUE="FlexWATCH300">

</APPLET>
</BODY>
</HTML>
```

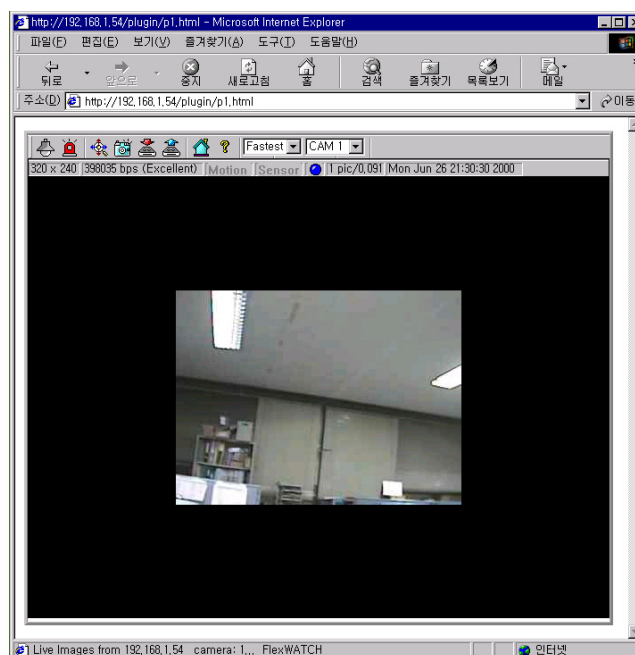
## 2. Web page with Plug-in

Let me explain how you could embed our plug-in in your Web page, and what attributes are there for controlling the plug-in.

## 1) Embedding Plug-in in Your Web page

As you know, to embed plug-in to your Web page you must use the <EMBED> tag, and the user who may access your page have installed our plug-in. In the <EMBED> tag, there are some attributes you must know correctly. The <EMBED> tag has a corresponding ending tag, </EMBED>. The HTML code listed below shows a HTML document for embedding our plug-in.

```
<HTML>
<HEAD></HEAD>
<BODY>
<EMBED SRC="flex.fdv" WIDTH="652" HEIGHT="542"
      CamNum="1" CamMode="1" PLUGINSOURCE="http://www.flexwatch.com">
</EMBED>
</BODY>
</HTML>
```



### \* The <EMBED> tag

As you can see from the above HTML, the <EMBED> tag requires six attributes: SRC, WIDTH, HEIGHT, CamNum, CamMode, PLUGINSOURCE:

```
<EMBED SRC="flex.fdv"
      WIDTH="652"
      HEIGHT="542"
      CamNum = "1"
      CamMode="1"
      PLUGINSOURCE="http://www.flexwatch.com">
```

### ATTRIBUTES:

#### SRC

- This attribute specifies a MIME type file that we have defined, “text/x-flexdv”. This MIME type have a corresponding extension “.fdv”. It’s not a plug-in itself, but it’s just a trigger for loading our plug-in. The user who already install our plug-in may see the plug-in is loaded in their Web browser, when they access this Web page.



**WIDTH**

- This attribute specifies the width of the plug-in's window in pixels.

**HEIGHT**

- This attribute specifies the height of the plug-in's window in pixels.

**CamNum**

- This attribute specifies a camera source. In the case of FlexWATCH 300/500, CamNum is from 1 to 6.

**CamMode**

- This is a reserved attribute.

**PLUGINSOURCE**

- If an user accesses this Web page who does not install our plug-in yet, the Web browser make a dialog box to inform the user that he does not have proper plug-in for this Web page and he can get the plug-in at that URL.

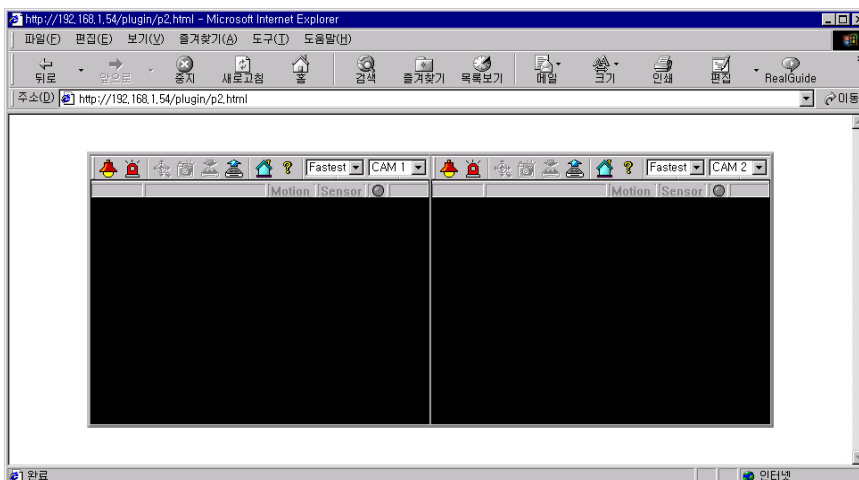
You can make a multi-view plug-in Web page by using <EMBED> tag in multiple. And you can align them with <TABLE> tag.

```
<HTML>
<HEAD></HEAD>
<BODY>
<TABLE WIDTH='100%' HEIGHT='100%' CELLPADDING=0 CELLSPACING=0 BORDER=0 BGCOLOR='#FFFFFF'>
<TR>
<TD align=center valign='middle'>

<TABLE border="3" cellspacing="0" cellpadding="0">
<tr>
<td align="center">
<EMBED SRC="flex.fdv" height="300" camMode="1" width="380" camNum = "1" PLUGINSOURCE="http://www.flexwatch.com">
</EMBED></td>
<td align="center">
<EMBED SRC="flex.fdv" height="300" camMode="1" width="380" camNum = "2" PLUGINSOURCE="http://www.flexwatch.com">
</EMBED></td>
</tr>
</TABLE>

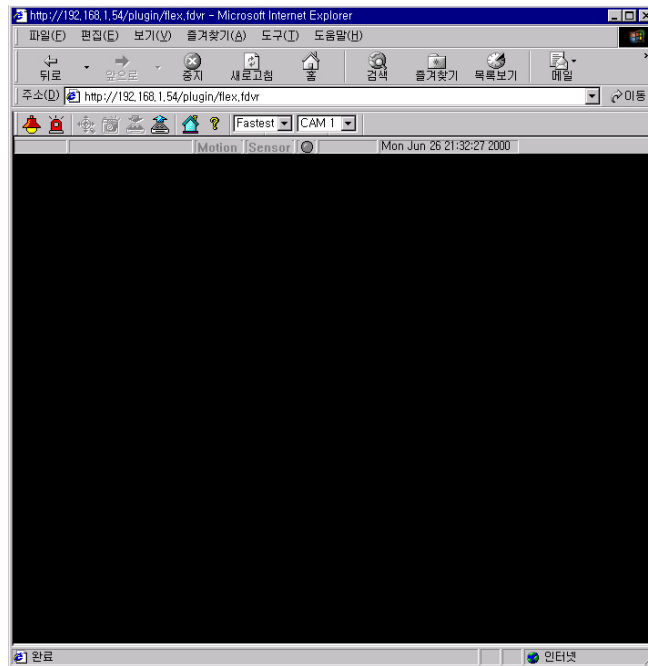
</TD>
</TR>
</TABLE>

</BODY>
</HTML>
```



You can load the plug-in in the full Web browser client area like below.

[Click](flex.fdv)



## CHAPTER 2. Updating Web page on FlexWATCH

Simply say, FlexWATCH is a Web server. As long as other UNIX server has TELNET & FTP, FlexWATCH

has those also. If you have made your own Web pages, you just upload that files using FTP to FlexWATCH. That's all you have to do. But FlexWATCH use flash memory instead of hard disk drive, so there are some limitation in the number of files and total file size. And you must execute a command after uploading your Web page on the FlexWATCH that write that pages into flash memory.

To do this job, you may use CONSOLE or TELNET, and FTP. If you don't have any idea to use CONSOLE, TELNET, and FTP, please refer the user's manual.

Let me explain the differences.

PS)

When you try to log-in FlexWATCH through FTP, you must use the ID, "root". You can use this ID when you enter the Web administration tool, and TELNET also.

## Differences between general UNIX server & FlexWATCH

1<sup>st</sup>. FlexWATCH provide UNIX like command, EXCEPT EDITOR, such as vi. Refer Techical Manual.

2<sup>nd</sup>. You must put your Web pages under '/flexwatch' directory.

3<sup>rd</sup>. You must not remove system directories.

/flexwatch/admin  
/flexwatch/cgi-bin  
/flexwatch/conf  
/flexwatch/logs  
/flexwatch/passwd  
/flexwatch/script

4<sup>th</sup>. You can upload files up to 300. (This number include the number of directories.)

5<sup>th</sup>. There is about 768Kbyte space for Web pages in FlexWATCH. Make sure the size of your contents.

6<sup>th</sup>. The file name length must not exceed 12 characters. The file name must not begin with digit.

7<sup>th</sup>. After you upload Web pages on the FlexWATCH, you must excecute "chghtml" command.

- Because "chghtml" is the command to write Web pages into flash memory, you should be careful when you use this command.
- Do not reset during updating. MUST!!
- You would better disconnect FlexWATCH from network during updating.
- You can use this command in CONSOLE or TELNET.
- It is recommended to backup up the Web pages on FlexWATCH before you erase it.
- If you do not execute this command after update files on FlexWATCH and do reset, you may find old files in FlexWATCH.
- The image below shows the process of chghtml command.

```

Serial_9600 - 하이퍼터미널
파일(F) 편집(E) 보기(V) 전화(C) 전송(T) 도움말(H)

$ chghtml
Not Supported PTZ Receiver Type !

chg: Command ambiguous
$

$ chghtml

[ ] Do you want to update html documents ? (y/n)y

Html documents are being updated.....!

System will be rebooted !
FE Code ffffffff
System Initialize...
Reinitialize ...
FLEXWATCH 500 Ver 1.1 (Build:2K0530), S/N : 2000B0196
$

$

Not Supported PTZ Receiver Type !
-

```

연결 0:00:45 VT100 38400 8-N-1 SCROLL CAPS NUM 갈무리 만항 인쇄

### Example work process

- 1<sup>st</sup>. Download the Web page in the FlexWATCH.
- 2<sup>nd</sup>. Remove old Web page. (through CONSOLE or TELNET)
 

```
$> cd /
```

```
$> rm -r flexwatch
```
- 3<sup>rd</sup>. Make flexwatch directory.
 

```
$> cd /
```

```
$> mkdir flexwatch
```
- 4<sup>th</sup>. Upload directories below under '/flexwatch' directory among files that are downloaded at the first time.
 

```
/admin
```

```
/cgi-bin
```

```
/conf
```

```
/logs
```

```
/passwd
```

```
/script
```
- 5<sup>th</sup>. Upload your own made pages on FlexWATCH. (NOTICE! You must upload pages under '/flexwatch' directory)
- 6<sup>th</sup>. Execute 'chghtml' command. (through CONSOLE or TELNET)
 

```
$> chghtml
```
- 7<sup>th</sup>. After one minute, FlexWATCH will be rebooted automatically.