



POWERLOGIC® Ethernet Gateway EGX400

Reference Guide

Retain for future use.

CONTENTS

CONTENTS	1
INTRODUCTION	2
SAFETY PRECAUTIONS	2
APPLICATIONS FOR THE EGX400	2
Using the EGX400 as a Web Server	2
Using the EGX400 as an FTP Server	3
OVERVIEW OF CREATING CUSTOM WEB PAGES FOR THE EGX400	3
Hardware, Software, and Logistics Requirements	3
Components of Custom HTML Pages	3
Static Components	3
Dynamic Components	3
Accessing Devices	4
EXAMPLE 1—CREATING A NEW HTML CUSTOM PAGE	5
HTML Source Code for Example 1	5
EXAMPLE 2—MODIFYING AN EXISTING HTML CUSTOM PAGE ...	10
Solution	10
HTML Modification	11
JavaScripting Source Code Modification	11
USING TEMPLATES	12
STORING FILES ONBOARD THE EGX400	13
DELETING FILES FROM THE EGX400	15

INTRODUCTION

This reference manual specifically covers the additional features of the POWERLOGIC Ethernet Gateway EGX400 (Figure 1) as compared to the EGX200. For details on installation and operation of both the EGX200 and EGX400, refer to instruction bulletin 63230-314-200.



Figure 1: POWERLOGIC Ethernet Gateway EGX400

SAFETY PRECAUTIONS

⚠ DANGER

HAZARD OF ELECTRIC SHOCK, BURN, OR EXPLOSION

- This equipment must be installed and serviced only by qualified personnel.
- Turn off all power supplying this equipment and the equipment in which it is installed before working on this equipment.
- Always use a properly rated voltage sensing device to confirm that power is off.

Failure to observe these instructions will result in death or serious injury.

APPLICATIONS FOR THE EGX400

The EGX400 functions primarily as an Ethernet gateway to allow Ethernet access to Modbus/Jbus and POWERLOGIC serial devices. Additionally, the EGX400 functions as a web server.

Using the EGX400 as a Web Server

The EGX400 provides web server functionality that supports the HTTP protocol. You can configure it remotely and it lets you view real time data with a web browser. In addition, the EGX400 has nonvolatile memory for storing files, graphics, manuals, and web pages.

The EGX400 supports the following Ethernet protocols:

- Modbus/TCP server via port 502 to serve compatible Modbus/TCP masters such as the System Manager™ Software (SMS) system
- HTTP server via port 80 to serve on-board web pages
- FTP server via port 21 to download files

Using the EGX400 as an FTP Server

The EGX400 supports FTP (file transfer protocol) making it possible to download the following types of files to be displayed via the web server:

- GIF and JPEG graphics files
- PDFs
- Java applets
- ActiveX controls

NOTE: The file types with an asterisk () will be parsed by the EGX400 as described in the following sections.*

- HTM/HTML*
- XML*
- XSL*
- XSD*
- DTD*
- Txt files*

OVERVIEW OF CREATING CUSTOM WEB PAGES FOR THE EGX400

To create custom web pages for the EGX400, you should have the following:

- A general understanding of the POWERLOGIC Power Monitoring and Control System
- A general understanding of the Internet and the World Wide Web (WWW)
- Basic skills with text editor software
- Working knowledge of HyperText Markup Language (HTML) and JavaScripting.

Hardware, Software, and Logistics Requirements

Before proceeding, ensure that you have the following:

- The EGX400 installed and assigned an IP address
- Access to the EGX via a LAN connection
- Web page editor or text editor software

Components of Custom HTML Pages

In general, each EGX custom page has two components and each one is developed using different tools.

Static Components

This component includes the page layout, static text, color schemes, lines, and tables. This part of the custom page is usually created with a web page editor and customized by adding or modifying HTML tags.

Many books are available about HTML and the web that can give you a basic understanding of HTML tags and rules. We recommend acquiring such reference materials to aid in your custom web page creation.

Because the static portion of the web page is dependant on the user, it is left to the web designer to decide how to write the HTML code. Therefore, it is not addressed in this manual.

Dynamic Components

This component includes special delimiters that tell the EGX to dynamically get register information from attached devices and display it in the HTML page.

Accessing Devices

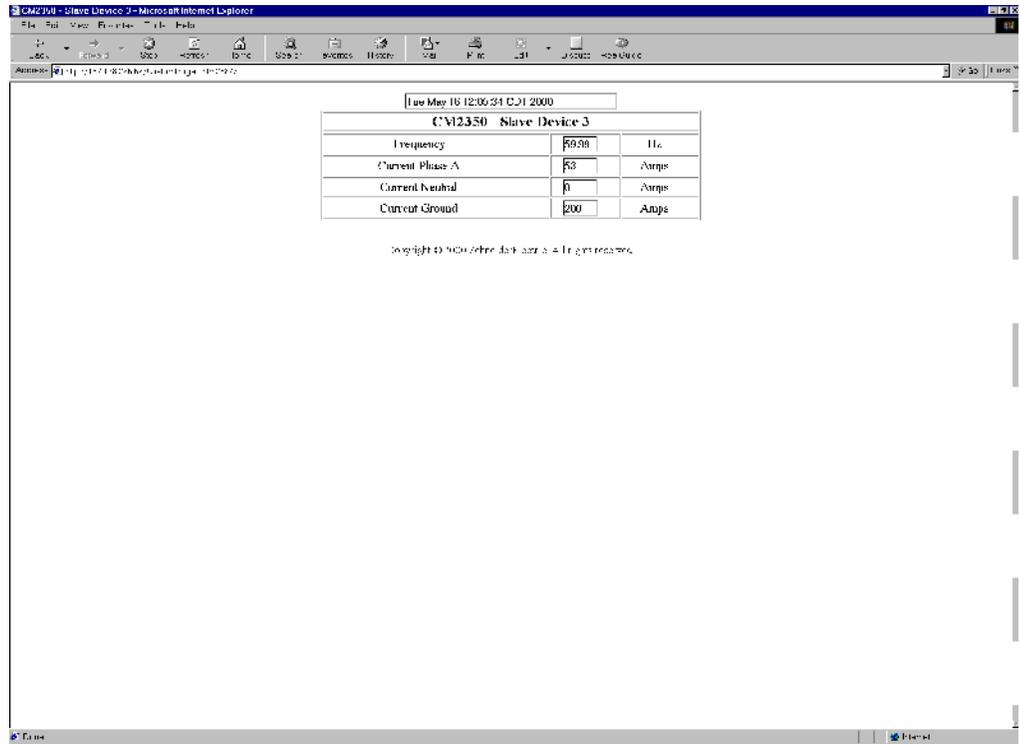
In general, the custom pages will be written in HTML with a special delimiter that tells the EGX400 to dynamically get register information from a device. The delimiter at the beginning of a string is (PL__) and the delimiter at the end is (__PL). This tells the EGX400 to parse this string and dynamically fill it with register data. Table 1 lists the supported POWERLOGIC tags.

Table 1: POWERLOGIC Tags and Usage

Function Name	Function Code	POWERLOGIC TAG
SyMax Block Read - Registers	SyMax Function Code 0	<DeviceID>^<StartingRegisterAddress>[<NumberOfRegisters>] example tag = PL__1^1003[5]__PL example of data returned = 85,86,84,25,56
SyMax Scattered Read – Registers	SyMax Function Code 4	<DeviceID>^<RegisterAddress1>,<RegisterAddress2>,etc example tag = PL__1^1003,1004,1005,1006,1007__PL example of data returned = 85,86,84,25,56
Modbus Block Read- Coil Status	Modbus Function Code 1	<DeviceID>^C<StartingCoilAddress>[<NumberOfCoils>] example tag = PL__1^C1003[5]__PL example of data returned = 1,0,0,1,1
Modbus Block Read – Input Status	Modbus Function Code 2	<DeviceID>^D<StartingInputAddress>[<NumberOfInputs>] example tag = PL__1^D1003[5]__PL example of data returned = 1,0,0,1,1
Modbus Block Read – Holding Registers	Modbus Function Code 3	<DeviceID>^H<StartingRegisterAddress>[<NumberOfRegisters>] example tag = PL__1^H1003[5]__PL example of data returned = 85,86,84,25,56
Modbus Block Read – Input Registers	Modbus Function Code 4	<DeviceID>^I<StartingRegisterAddress>[<NumberOfRegisters>] example tag = PL__1^I1003[5]__PL example of data returned = 85,86,84,25,56
Modbus Scattered Read – Holding Registers	Modbus Function Code 100	<DeviceID>^S<RegisterAddress1>,<RegisterAddress2>,etc example tag = PL__1^S1003,1004,1005,1006,1007__PL example of data returned = 85,86,84,25,56

EXAMPLE 1—CREATING A NEW HTML CUSTOM PAGE

Example 1 is an exercise in creating the following new HTML page for downloading to the EGX400.



HTML Source Code for Example 1

The following is the HTML source code for the Example 1 web page. See Table 2 on page 10 for descriptions of source code.

NOTE: Bold type indicates the HTML and JavaScript that are key points to consider when making a custom page.

Line No. HTML Syntax

1. <html>
2. <head>
3. <META HTTP-EQUIV="refresh" CONTENT="5">
4. <title>**CM2350 - Slave Device 3**</title>
5. </head>
6. <body>
7. <form name="view_form">
8. <p align="center">
9. <input type = "text" name = "time_spot" size = "40">
- 10.<table border="1" width="600">
- 11.<tr>
- 12.<td width="600"><p align="center">
- 13.**CM2350 - Slave Device 3**</p></td></tr>

```
14.</td>
15.</tr>
16.</table>
17.<table border="1" width="600">
18.<tr>
19.<td width="300">
20.<p align="center">Frequency</p>
21.</td>
22.<td align="center" width="90"><p align="center">
23.<input type="text" size="5" name="frequency"></p>
24.<td width="100">
25.<p align="center">Hz</p>
26.</td>
27.</tr>
28.<tr>
29.<td width="300">
30.<p align="center">Current Phase A</p>
31.</td>
32.<td align="center" width="90"><p align="center">
33.<input type="text" size="5" name="currentphasea"></p>
34.<td width="100">
35.<p align="center">Amps</p>
36.</td>
37.</tr>
38.<tr>
39.<td width="300">
40.<p align="center">Current Neutral</p>
41.</td>
42.<td align="center" width="90"><p align="center">
43.<input type="text" size="5" name="currentneutral"></p>
44.<td width="100">
45.<p align="center">Amps</p>
46.</td>
47.</tr>
48.<tr>
49.<td width="300">
50.<p align="center">Current Ground</p>
51.</td>
52.<td align="center" width="90"><p align="center">
53.<input type="text" size="5" name="currentground"></p>
```

```
54.<td width="100">
55.<p align="center">Amps</p>
56.</td>
57.</tr>
58.</table>
59.<br><HR SIZE="1" width="66%"><CENTER><font
    face="Times Roman" size="2">Copyright © 2000
    SchneiderElectric. All rights reserved.</font></
    CENTER>
60.</form>
61.<script language="JavaScript">
62.function ShowFreq()
63.{
64.Registers =
    [PL__3^2020,2021,2022,2025,1001,1003,1006,1007__PL];
65.ScaleFactorA = Registers[0];
66.ScaleFactorB = Registers[1];
67.ScaleFactorC = Registers[2];
68.ScaleFactorF = Registers[3];
69.Frequency = Registers[4];
70.CurrentPhaseA = Registers[5];
71.CurrentNeutral = Registers[6];
72.CurrentGround = Registers[7];
73.ScaleFactorAMultiplier = 0;
74.ScaleFactorBMultiplier = 0;
75.ScaleFactorCMultiplier = 0;
76.ScaleFactorFMultiplier = 0;
77.TheTime = new Date();
78.switch (ScaleFactorA)
79.{
80.case -2:
81.ScaleFactorAMultiplier = 0.01;
82.break;
83.case -1:
84.ScaleFactorAMultiplier = 0.1;
85.break;
86.case 1:
87.ScaleFactorAMultiplier = 10;
88.break;
89.default:
90.ScaleFactorAMultiplier = 1;
91.break;
```

```
92.}
93.switch (ScaleFactorB)
94.{
95.case -2:
96.ScaleFactorBMultiplier = 0.01;
97.break;
98.case -1:
99.ScaleFactorBMultiplier = 0.1;
100.break;
101.case 1:
102.ScaleFactorBMultiplier = 10;
103.break;
104.default:
105.ScaleFactorBMultiplier = 1;
106.break;
107.}
108.switch (ScaleFactorC)
109.{
110.case -2:
111.ScaleFactorCMultiplier = 0.01;
112.break;
113.case -1:
114.ScaleFactorCMultiplier = 0.1;
115.break;
116.case 1:
117.ScaleFactorCMultiplier = 10;
118.break;
119.default:
120.ScaleFactorCMultiplier = 1;
121.break;
122.}
123.switch (ScaleFactorF)
124.{
125.case -1:
126.ScaleFactorFMultiplier = 0.1;
127.break;
128.default:
129.ScaleFactorFMultiplier = 0.01;
130.break;
131.}
```

```
132.Frequency *= ScaleFactorFMultiplier;
133.CurrentPhaseA *= ScaleFactorAMultiplier;
134.if (CurrentNeutral == -32768)
135.CurrentNeutral = "N/A";
136.else
137.CurrentNeutral *= ScaleFactorBMultiplier;
138.if (CurrentGround == -32768)
139.CurrentGround = "N/A";
140.else
141.CurrentGround *= ScaleFactorCMultiplier;
142.document.view_form.frequency.value = Frequency;
143.document.view_form.currentphasea.value =
    CurrentPhaseA;
144.document.view_form.currentneutral.value =
    CurrentNeutral;
145.document.view_form.currentground.value =
    CurrentGround;
146.document.view_form.time_spot.value = TheTime;
147.}
148.ShowFreq();
149.</script>
150.</body>
151.</html>
```

Table 2 provides descriptions of HTML source code for Example 1.

Table 2: Description of HTML Source Code for Example 1

HTML Code Line No.	Description
HTML Source for the Static Elements	
3	HTML tag to set up page refresh cycle in seconds.
4	HTML tag to define page title label. This title appears on the browser title bar and is used in the main links page of the EGX400.
13	HTML syntax to write the title of the table "CM2350 - Slave Device 3".
20	HTML syntax to write "Frequency" table cell text label.
23	HTML syntax for the input control to be filled by dynamic data.
25	HTML syntax to write "Hz".
30, 33, 35	HTML syntax for displaying Current Phase A.
40, 43, 45	HTML syntax for displaying Current Neutral.
50, 53, 55	HTML syntax for displaying Current Ground.
JavaScripting Code for the Dynamic elements	
64	This line contains the following: "PL" delimiters at the beginning and end to signify to the EGX400 to parse this string and dynamically fill it with register data. 3^ to signify the device address on the daisy chain. 2020,2021, ...,1007 a list of register numbers, which contain necessary CM2350 data.
65	Register #2020 of the CM2350 has the Scale Factor A value.
66	Register #2021 of the CM2350 has the Scale Factor B value.
67	Register #2022 of the CM2350 has the Scale Factor C value.
68	Register #2025 of the CM2350 has the Scale Factor F value.
69	Register #1001 of the CM2350 has the Frequency value.
70	Register #1003 of the CM2350 has the Current Phase A value.
71	Register #1006 of the CM2350 has the CurrentNeutral value.
72	Register #1007 of the CM2350 has the Current Ground value.
142-145	JavaScript statements to print variable values into their field.

Once you have created the HTML page, you must download this page into the EGX400 using FTP. For details on how to use FTP, see "Storing Files Onboard The EGX400" on page 13.

EXAMPLE 2—MODIFYING AN EXISTING HTML CUSTOM PAGE

Example 2 is an exercise in modifying the Example 1 custom page to pull data from a CM4000 rather than a CM2000. For Example 2, the CM4000 has a device address of 1 on the EGX400 daisy chain.

Solution

In this exercise, we will keep the web page layout and colors exactly the same. However, the real-time data tags must be changed to get information from the CM4000 rather than the CM2000. To do that, open the HTML source file using any web page or text editor and make the following changes.

HTML Modification

Replace the title line:

```
4. <title>CM2350 - Slave Device 3</title>  
with:
```

```
4. <title>CM4000 - Slave Device 1</title>
```

Also, replace the title of the table:

```
13. CM2350 - Slave Device 3</b></font></p>  
with:
```

```
13. CM4000 - Slave Device 1</b></font></p>
```

JavaScripting Source Code Modification

To receive the dynamic data from the CM4000 rather than the CM2000, change the CM2000 device address and register numbers to the CM4000 device address and its corresponding register numbers. Make the following changes:

Replace:

```
64. Registers =  
[PL__3^2020,2021,2022,2025,1001,1003,1006,1007__PL];  
with:
```

```
64. Registers =  
[PL__1^3209,3210,3211,1180,1000,1003,1005__PL];
```

NOTE: The CM4000 register list is one register less than the CM2000. The frequency parameter in the CM2000 is stored in register number 1001 and must be formatted by a scaling factor that is saved in register 2025. While the CM4000 frequency reading also requires formatting, a fixed scaling factor of 0.01 is always applied and does not require a register. Therefore, make the following modification as well:

Replace the following line of code:

```
76. ScaleFactorFMultiplier = 0;  
with:
```

```
76. ScaleFactorFMultiplier = 0.01;
```

Also, you need to delete lines 123 through 131.

Once complete, save the file. Then, download this page into the EGX400 using FTP. For more details on how to use FTP, see “Storing Files Onboard The EGX400” on page 13.

USING TEMPLATES

The CD-ROM that accompanies the EGX400 has some sample HTML pages. You can view the file source code to better understand how they are constructed and experiment with them by changing the source code to see what effect that has on a page. You can also use these pages as templates and modify the source code to create your own custom pages.

STORING FILES ONBOARD THE EGX400

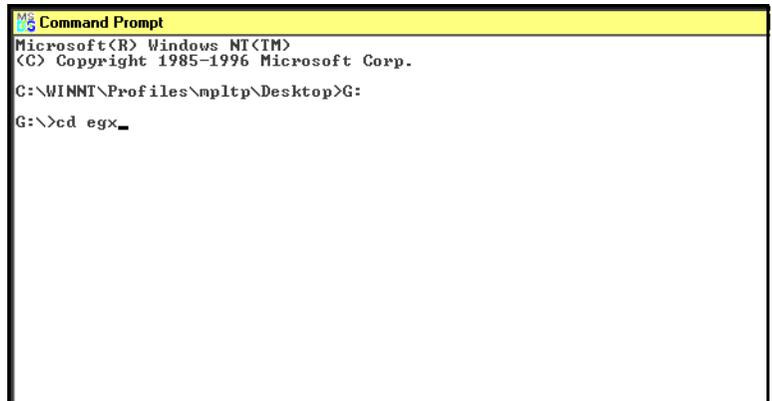
The EGX400 has internal memory for storing the custom web pages:

Use File Transfer Protocol (FTP) to download these files. Follow these steps:

1. Create a folder on your computer hard drive to store files you want to download into the EGX400 via FTP. In this example, the folder **egx** is located on the **G:** drive.
2. Access DOS on your computer by selecting Start>Program>Command Prompt.

NOTE: This path is typical for Windows NT. The path may differ for other platforms.

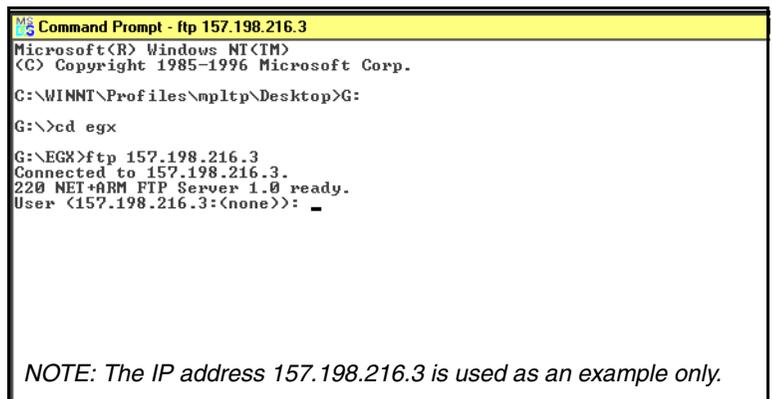
3. Type the drive you want to access (in this example **G:**) and press Enter.
4. Change to your new directory. Type: **cd egx** and press Enter. See Figure 2.



```
MS-DOS Command Prompt
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
C:\WINNT\Profiles\mplt\p\Desktop>G:
G:\>cd egx_
```

Figure 2: Identifying the folder where the files are stored

5. Type: **ftp** and the **IP address** assigned to the EGX400 and press Enter. You should receive the message “Connected to [IP address]”, indicating you are now in an FTP session (Figure 3).



```
MS-DOS Command Prompt - ftp 157.198.216.3
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
C:\WINNT\Profiles\mplt\p\Desktop>G:
G:\>cd egx
G:\EGX>ftp 157.198.216.3
Connected to 157.198.216.3.
220 NET+ARM FTP Server 1.0 ready.
User (157.198.216.3:(none)): _
```

NOTE: The IP address 157.198.216.3 is used as an example only.

Figure 3: FTP connection to the IP address

- At the “User [IP address:(none)]:” prompt, press Enter.
The Password prompt displays (Figure 4).

```
MS Command Prompt - ftp 157.198.216.3
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\Profiles\npltp\Desktop>G:

G:\>cd egx

G:\EGX>ftp 157.198.216.3
Connected to 157.198.216.3.
220 NET*ARM FTP Server 1.0 ready.
User (157.198.216.3:(none)):
331 User (none) OK, send password.
Password:
```

Figure 4: Password prompt

- At the Password prompt, enter the administrator password (**admin** is the default password until the administrator changes it).
- At the ftp prompt, type **send [filename]** and press Enter to initiate the ftp transfer. (In this example, we entered “send Example.htm”). When the download is complete (Figure 5), the “ftp” prompt displays again.

NOTE: Limit filenames to eight characters plus a three-character extension.

```
MS Command Prompt - ftp 157.198.216.3
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\WINNT\Profiles\npltp\Desktop>G:

G:\>cd egx

G:\EGX>ftp 157.198.216.3
Connected to 157.198.216.3.
220 NET*ARM FTP Server 1.0 ready.
User (157.198.216.3:(none)):
331 User (none) OK, send password.
Password:
230 Password OK.
ftp> send Example.htm
200 PORT command OK.
150 About to open data connection.
226 Transfer complete
895 bytes sent in 0.00 seconds (895000.00 Kbytes/sec)
ftp> -
```

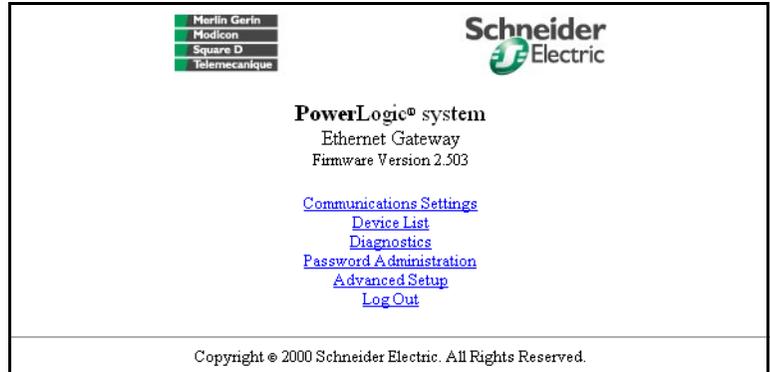
Figure 5: Download complete

- If you have another custom web page to download, type **send [filename]** and press Enter. If you are finished downloading files, type **quit** and press Enter to exit the FTP session.

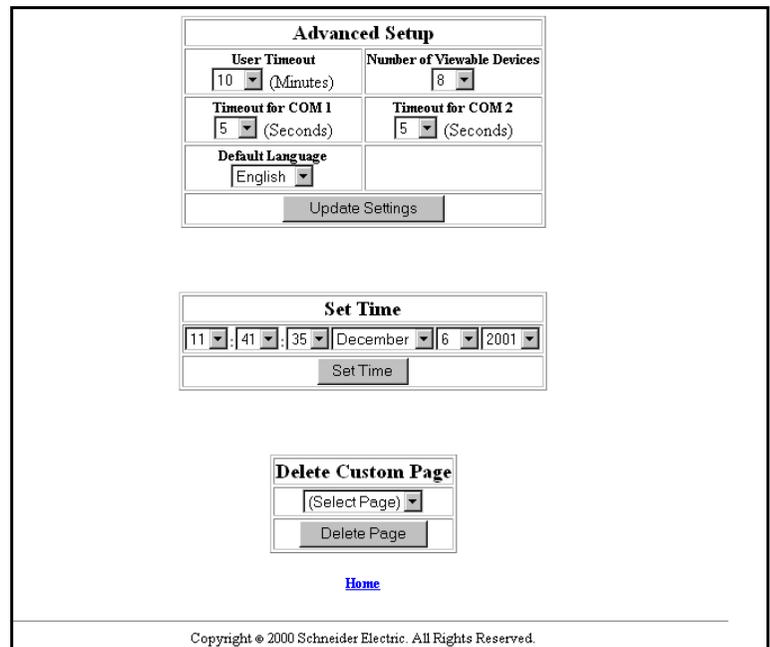
DELETING FILES FROM THE EGX400

To delete a file, follow these steps:

1. Log into the EGX400 with the “admin” password.
The EGX Home page displays.



2. Click Advanced Setup.
The Advanced Setup screen displays.



3. Select the file to delete from the “Delete Custom Page” pull down list and then click the Delete Page button.

Electrical equipment should be serviced only by qualified maintenance personnel. No responsibility is assumed by Square D for any consequences arising out of the use of this material.