Allen-Bradley

EtherNet/IP Web Server Module

1756-EWEB, 1768-EWEB

User Manual

Rockwell Automation

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://literature.rockwellautomation.com) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING 	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.
ATTENTION 	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
SHOCK HAZARD 	Labels may be located on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
BURN HAZARD 	Labels may be located on or inside the equipment, for example, a drive or motor, to alert people that surfaces may be at dangerous temperatures.

This document describes how to use the EtherNet/IP Web server module. Revision bars in the margin identify updated information. Changes for this version of the document include the addition of the 1768-EWEB module and related information.

Enhanced Web Server Module User Manual Changes

Topic	Chapter	Page No.
Browser Requirements	Chapter 1 - Getting Started	17
1768-EWEB Installation	Chapter 1 - Getting Started	18
1768-EWEB Flash File Space	Chapter 7 - Access Files in the Web Server Module	83
CIP Connected Messaging Limits	Appendix A - Use the Web Server Module To Connect Over Ethernet	122
CIP Unconnected Messaging Limits	Appendix A - Use the Web Server Module To Connect Over Ethernet	122

Notes:

Chapter 3

Manage Module Settings

Chapter 4

Use Data Views to Access Controller Data

Chapter 5

Send Email

Chapter 6

Manage User Accounts and Access Levels

Chapter 7

Access Files in the Web Server Module

Chapter 8

Create Custom Web Pages

Chapter 9

Monitor Diagnostics

Appendix A

Use the Web Server Module To Connect Over Ethernet

Appendix B

Socket Interface

Index

About This Publication

Use this manual as a reference when installing, using, and troubleshooting your EtherNet/IP Web Server Module.

This manual explains the use of the following EtherNet/IP Web Server modules:

- 1756-EWEB
- 1768-EWEB

Who Should Use This Publication

This manual is intended for anyone who accesses, configures, or manages the web pages EWEB module.

Conventions

Text that is	Identifies
Bold	A value that you must enter exactly as shown
Italic	A variable that you replace with your own text or value
`courier`	Example programming code, shown in a monospace font so you can identify each character and space
enclosed in brackets	A keyboard key

Getting Started

Introduction

This chapter describes procedures for getting started with your EtherNet/IP Web Server module. It includes information about the module and quick start procedures.

About the Enhanced Web Server Module

Both the 1756-EWEB and 1768-EWEB modules, known as Enhanced Web Server modules, provide access to information from the control system using a web browser.

Using an EWEB module, you can monitor and modify control system data remotely using XML web pages.

Enhanced Web Server Module Applications

The following features and applications are available with your Enhanced Web Server Module.

- Remote access to controller data using a standard web browser

  Use a standard web browser to monitor live controller data in two ways: use data views that you create in the web server module, or custom-develop your own web pages.

  For example, create a custom web page for managers to monitor production processes directly from their desks. Use data views or custom web pages for OEMs to remotely monitor controller data and reduce support costs.

- Deliver data initiated by the control system

  System data and information can be sent via email when initiated by a controller in the system. The controller uses a message instruction to initiate an email. Use the email to notify a maintenance person or an engineer of an alarm or alert so that corrective actions can be done in a timely fashion.

  The system can also send system status or production reports. The module supports all email clients, such as email applications and text pagers.

- Share system data with external applications

  The module stores data in its data views in XML files. This generic XML data presentation allows external applications to easily access and manipulate system data.

  XML support is also platform and operating system neutral, so you can share data between different applications. For example, design a database application to obtain controller data from the web server module to streamline the data acquisition process.

  In addition, the module supports an open-socket interface that lets a Logix controller communicate with Ethernet devices that do not support the EtherNet/IP application protocol, such as bar code scanners or RFID readers.

Features of EtherNet/IP Web Server Module in a Control System

The module provides the following features and services in the control system.

- Bridging and routing of messages

  Like other EtherNet/IP modules, you can route messages, upload/download programs, and flash upgrade modules using the web server module as part of the communication path to access the target device.

- Data access (read and write) to controllers

  Access the XML pages in the Enhanced Web Server module to view and modify data that resides in a controller that is in the same chassis as the EWEB module.

- Custom web pages

  Create custom web pages that are tailored to your application. Use ASP functions to populate your web pages with live controller data.

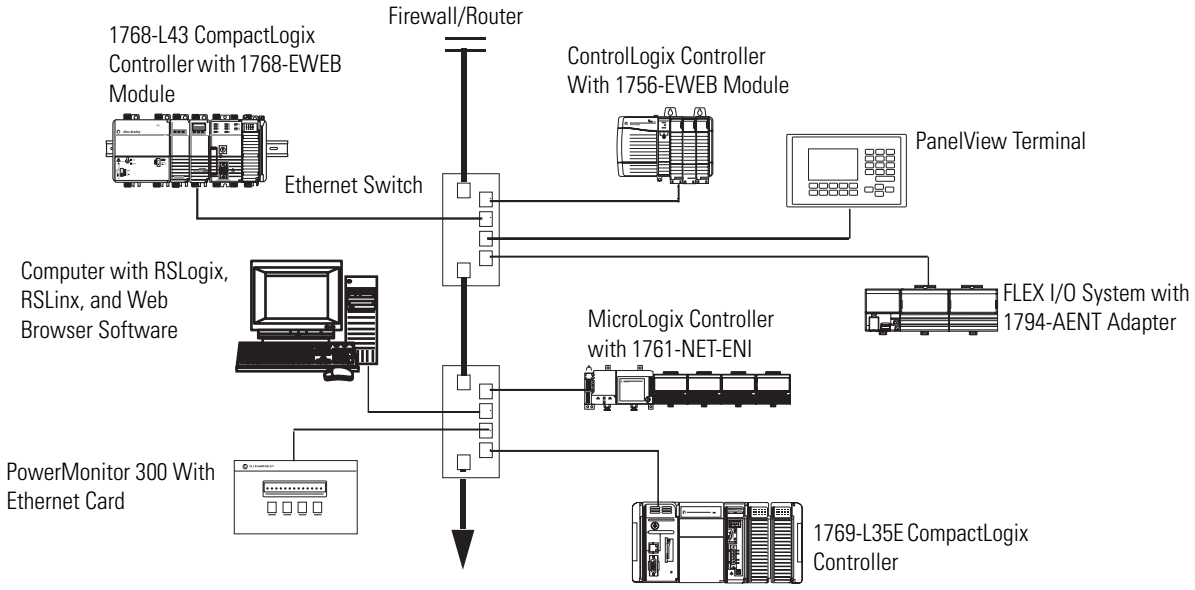
- Email capability

  You can initiate email messages from the embedded email composer in the module. You can also use the module to send an email initiated by a Logix controller via a MSG instruction.

- Open-socket interface

  You can open TCP or UDP communication links to other standard Ethernet devices via the module.

The following diagram shows how EWEB modules might fit in your control system on an EtherNet/IP network.

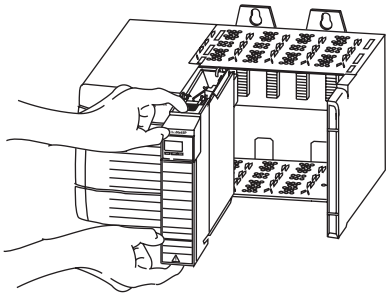


If both the ControlLogix and CompactLogix chassis in this sample system contain an EWEB module, you could access either module to monitor and modify data in the controllers using a computer with standard Web browser software.

1756-EWEB Installation

To install a ControlLogix Enhanced Web Server Module (1756-EWEB), follow these steps.

1. Align the module with a slot in the 1756 chassis.

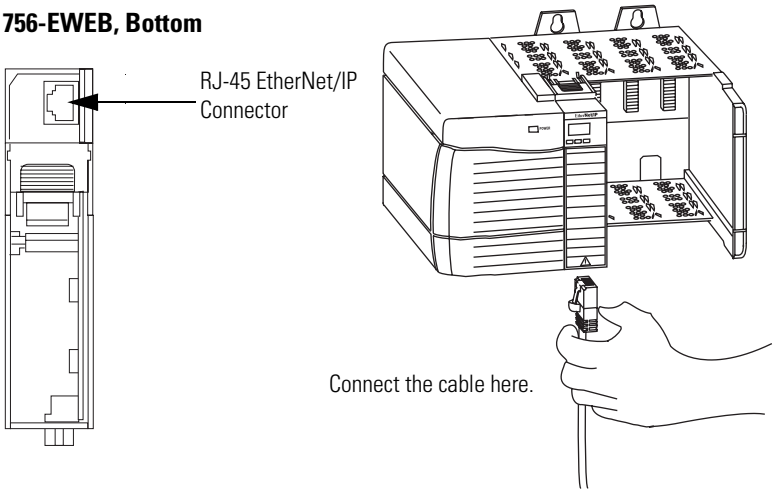


2. Slide the module back into the chassis until it snaps into place.

3. Connect the module to the network.

   The RJ-45 connector is on the bottom, front of the module.

1756-EWEB, Bottom



4. Obtain an IP address.

   For more information, see chapter 2.

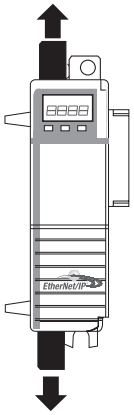
   By default, the web server module is DHCP enabled. If you connect the web server module to a network that has a DHCP server, that server will assign a dynamic IP address to the web server module and the four-digit display on the front of the web server module will display each of the four numbers of the IP address.

   If your network does not have a DHCP server, use one of the methods described in chapter 2 to assign an IP address to the web server module.

1768-EWEB Installation

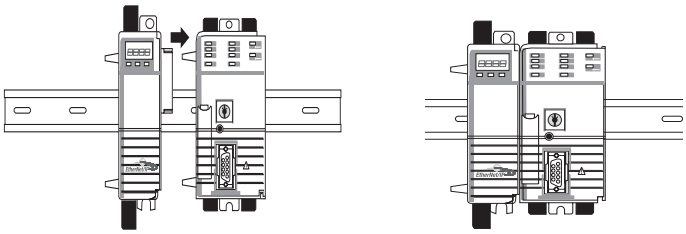
To install a CompactLogix Enhanced web Server module, complete the following steps.

1. Open the DIN rail latches on the module.



2. Align and press the module onto the DIN rail to the left of the controller.

3. Slide module snugly against the controller.



4. Install a power supply and other modules.

5. Close all the DIN rail latches.

6. Obtain an IP address.

   For more information, see chapter 2.

System Requirements

Browser Requirements

The following table describes browser requirements for specific tasks related to the Enhnaced Web Server module.

To	You Need
Access web pages generated by the Enhanced Web Server module	Any standard web browser
Create and edit data view web pages on the Enhanced Web Server module	Internet Explorer 5.5 or 6 with XML support
View sample code	Internet Explorer 5.5 or 6

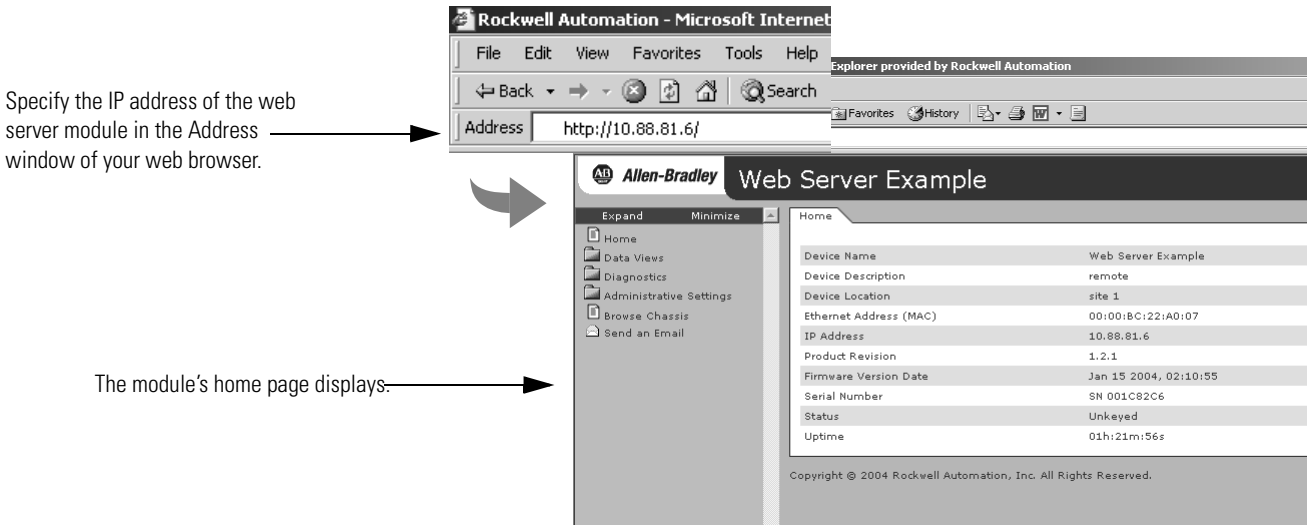
Display Size

The supported display size is 640 x 480 or greater. Smaller display sizes work but might require extensive scrolling to view the information.

Access the Module Using Your Web Browser

Use the following steps to access your EWEB module using you web browser.

1. In the address field of your web browser, enter the IP address of the module to access the module's home page.



2. Log into the module.

TIP Many of the features of the web server module require you to log in with appropriate access. If you select a feature such as New Data View, the web server module prompts you to enter your user name and password.

3. If logging into the module for the first time, enter the default user name 'Administrator' and leave the Password field blank.

Default Access:

User Name: Administrator
(not case sensitive)

Password:
(leave blank, no password)

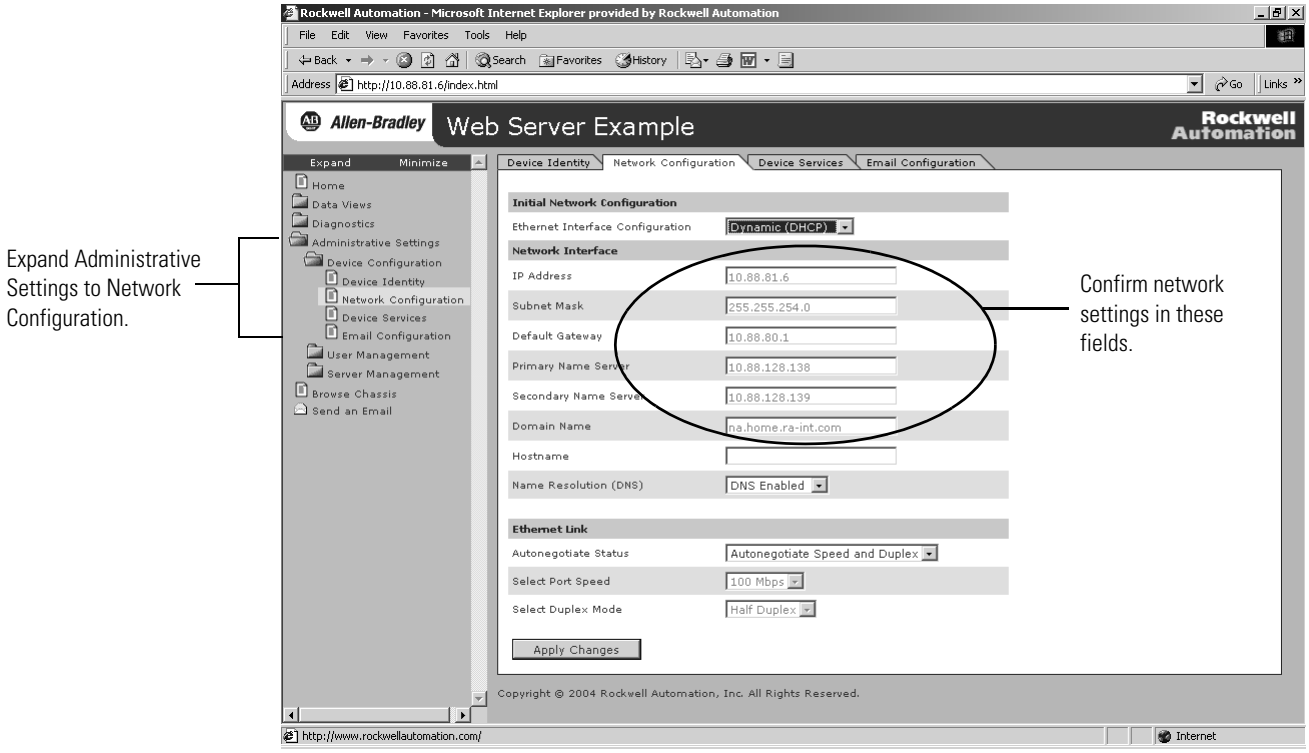
4. Click OK.

TIP You can set up as many as 25 user accounts. Each account can have read, read and write, or administrator access.

For more information, see chapter 6.

5. In the organizer on the left, select Administrative Settings > Device Configuration > Network Configuration.

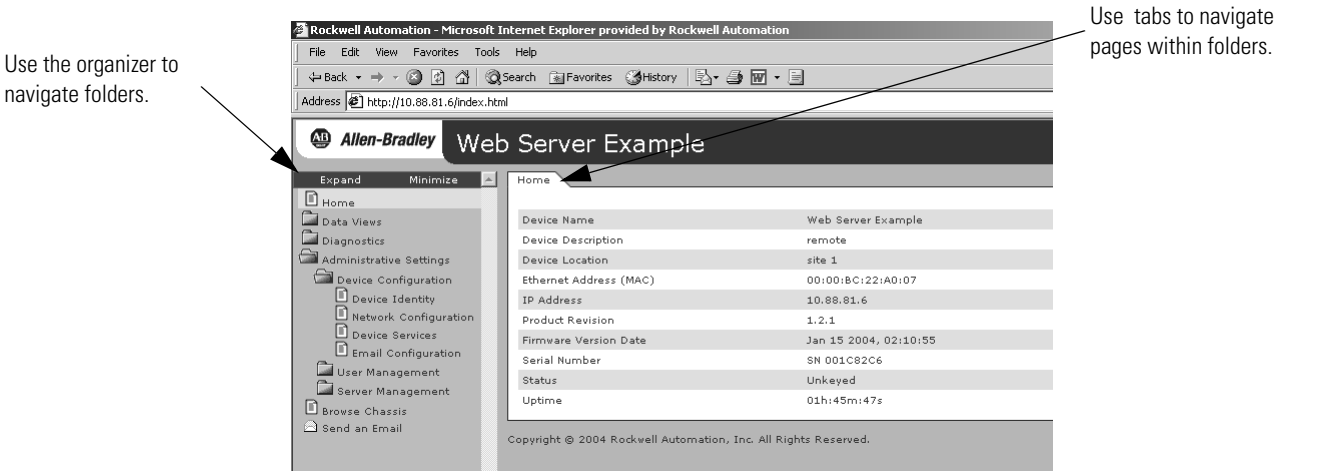
6. Confirm the network configuration by verifying the IP address and other network settings.



For more information, see chapter 2.

Navigate the Web Server Module

You navigate the web server's web pages using the organizer on the left of the screen. You can also use the tabs across the top to navigate the sections within folders.



Use the Web Server Module

To help familiarize yourself with the web server module, perform these basic tasks.

If You Want To	See Page
Create a Data View	23
Access a Data View	24
Configure Email	26
Configure the Time Server	27
Enable/disable Other Services	28

Create a Data View

Before you can create a data view in the web server, the tags you want to view must exist in the local controller (that is, the controller in the same chassis as the EWEB module) program.

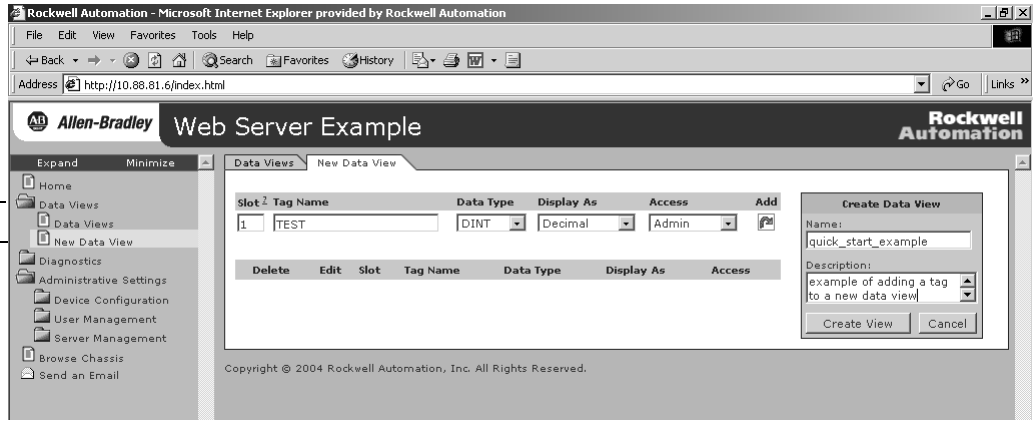
To complete these example, use programming software to create a tag with the following:

– Alias: TEST
Type: DINT
Controller-scope
Value: 12345

Create a Data View

To create a data view, you need Administrator or Write access.

1. In the organizer on the left, select Data Views > New Data View.



Expand Data View to New Data View.

2. In the Create Data View box, specify a name for the data view and enter an optional description.

3. Specify the:
   - slot number of the controller.
   - tag name (case sensitive; must be exactly as it is specified in the controller).
   - type of tag.
   - how to display the tag data.
   - access limit of the data view.

4. Click on the Add button to add the tag to the data view.

   Continue adding as many tags as you want to configure.

5. Click Create View.

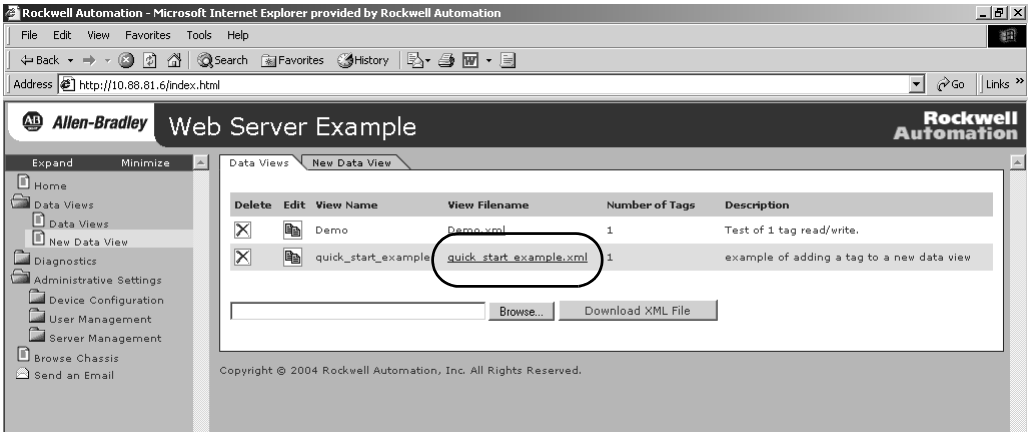
   For more information, see chapter 4.

Access a Data View

1. In the organizer on the left, select Data Views > Data View.

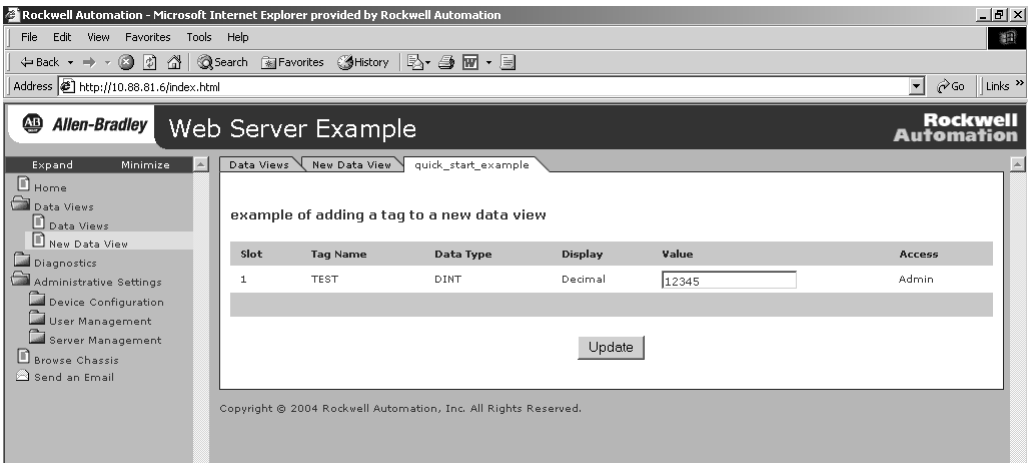
   -OR-

   Click the Data Views tab.



2. Click on the filename of the Data View you just created.

The created tags appear.



TIP To change a data value, you need Administrator or Write access.

3. If you want to change a tag value, enter the new value in the Value field the tag and click the Update button.

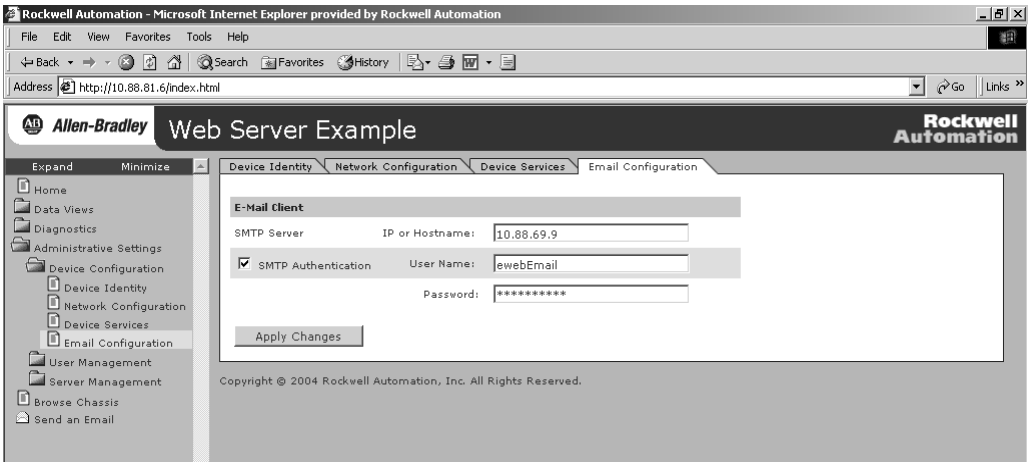
   This changes the value in the controller. You can use RSLogix5000 software to monitor tags and verify that the value changed.

   For more information, see chapter 4.

Configure Email

To configure the SMTP server that manages email, follow this procedure.

1. In the organizer on the left, select Administrative Settings > Device Configuration > Email Configuration.



2. In the organizer, select Send an Email to create and send email.

TIP You can have a controller execute a MSG instruction that initiates email through the web server module.

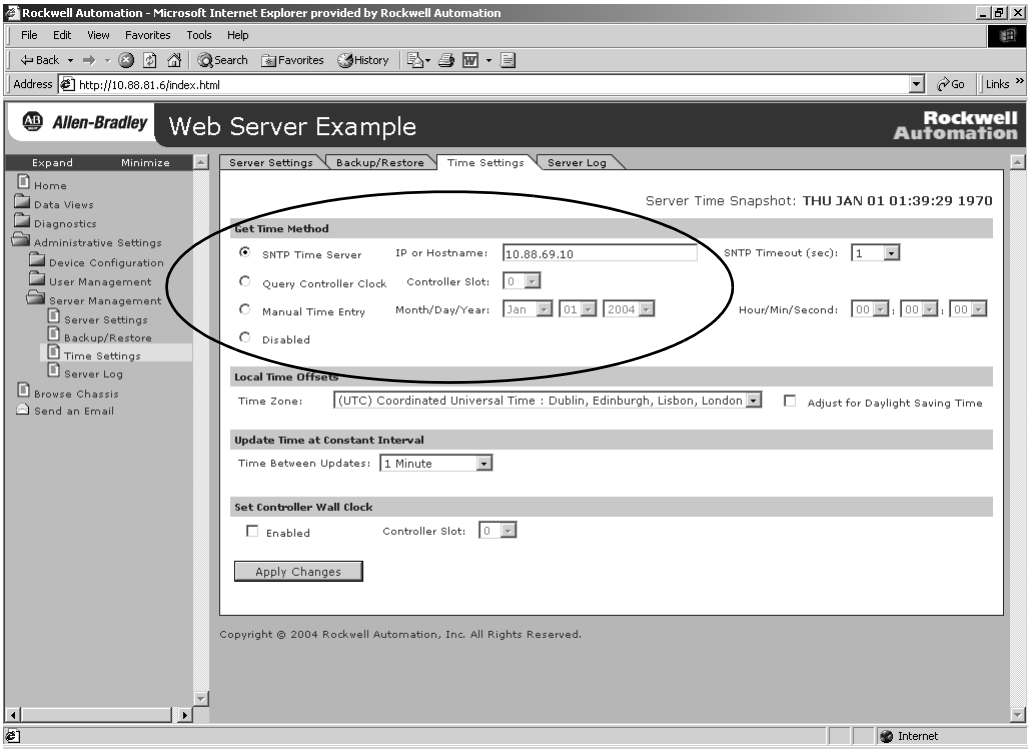
For more information, see chapter 5.

Configure the Time Server

Configuring the Time Server helps ensure that files you save to the web server module have accurate date and time stamps.

Complete the following steps to configure the time server.

1. In the organizer on the left, select Administrative Settings > Server Management > Time Settings.



2. Click the radio button to specify the time/date source according to your system.

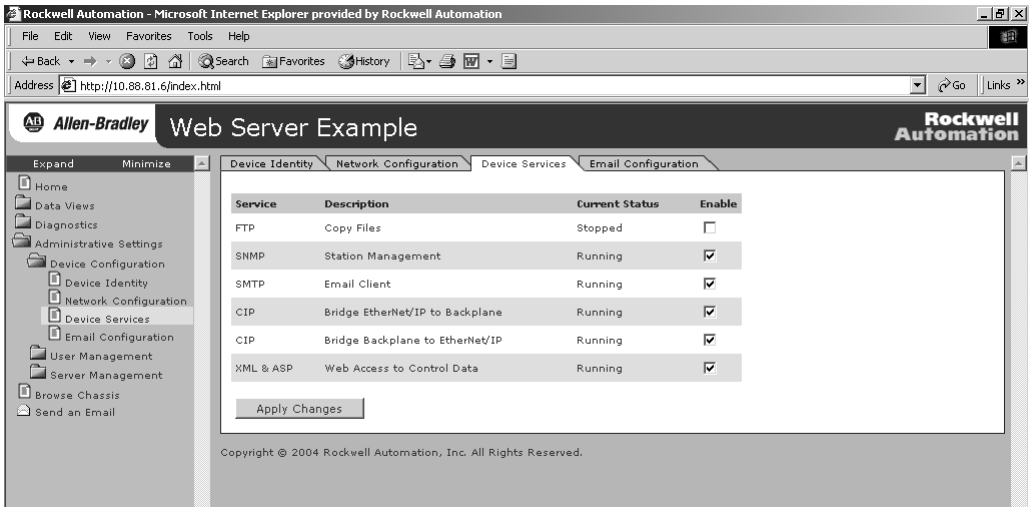
To	Select
Get the date and time from an SNTP server on the network.	SNTP Time Server
Get the time and date from the local controller	Query Controller
Specify your own date and time	Manual Time Entry

For more information, see chapter 3.

Enable/disable Other Services

Use this procedure to enable other services.

1. In the organizer, select Administrative Settings > Device Configuration > Device Services.



2. Select the services you want to use by checking the appropriate checkboxes.

   Use the following table as a reference.

To	Enable
Allow file tyransfers to and from the web server module	File Transfer Protocol (FTP)
Use SNMP management software (if your system has it)	Simple Network Mangament Protocol (SNMP)
Service email	Simple Mail Transfer Protocol (SMTP)
Allow Ethernet/IP devices to bridge through the web server module to devices in the chassis	Common Industrial Protocol (CIP) bridge backplane to EtherNet/IP service
Allow web access to control system data	Extended Markup Language/Active Server Page (XML/ASP)

For more information, see chapter 3.

Additional Resources

Consult the following publications for more information.

Topic	Publication Title	Publication No.
Creating controller tags using RSLogix5000	Logix5000 Controllers Common Procedures	1756-PM001
EtherNet/IP network	EtherNet/IP Modules in Logix5000 Control Systems User Manual	ENET-UM001
1756-EWEB module installation	EtherNet/IP Web Server Module Installation Instructions	1756-IN588
1768-EWEB module installation	EtherNet/IP Web Server Module Installation Instructions	1768-IN007

You can view or download publications at http://www.literature.rockwellautomation.com. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

Configure a Network Address For a Web Server Module

How to Use This Chapter

This chapter describes how to configure a module to operate on an Ethernet network.

Determine Which Network Parameters Are Required

For the module to operate on an Ethernet network, you must define these parameters.

Ethernet Parameter	Description
IP Address	The IP address uniquely identifies the module. The IP address is in the form *xxx.xxx.xxx.xxx* where each *xxx* is a number between 0...255. You cannot use these reserved values: • 127.0.0.1 • 0.0.0.0 • 255.255.255.255
Subnet mask	Subnet addressing is an extension of the IP address scheme that lets a site to use a single network ID for multiple physical networks. Routing outside of the site continues by dividing the IP address into a net ID and a host ID via the class. Inside a site, the subnet mask is used to redivide the IP address into a custom network ID portion and host ID.
Gateway	A gateway connects individual physical networks into a system of networks. When a node needs to communicate with a node on another network, a gateway transfers the data between the two networks.

If you use the module to initiate MSG instructions that use host names or to initiate emails, you must also define these parameters.

Ethernet Parameter	Description
Host Name	A host name is part of a text address that identifies the host for a module. The full text address of a module is host_name.domain_name.
Domain Name	A domain name is part of a text address that identifies the domain in which the module resides. The full text address of a module is host_name.domain_name. The domain name has a 48-character limit. If you specify a DNS server, you must enter a domain name. Also, if you send email from the module, some mail relay servers require a domain name be provided during the initial handshake of the SMTP session.
Primary DNS Server Address	This identifies the DNS server(s), if used in the network. You must have a DNS server configured if you specified a domain name or a host name in the module's configuration. The DNS server converts the domain name or host name to an IP address that can be used by the network.
Secondary DNS Server Address	

Check with your network administrator to determine if you need to specify all of the above parameters.

How you configure these network parameters depends on whether the Ethernet network has a DHCP server.

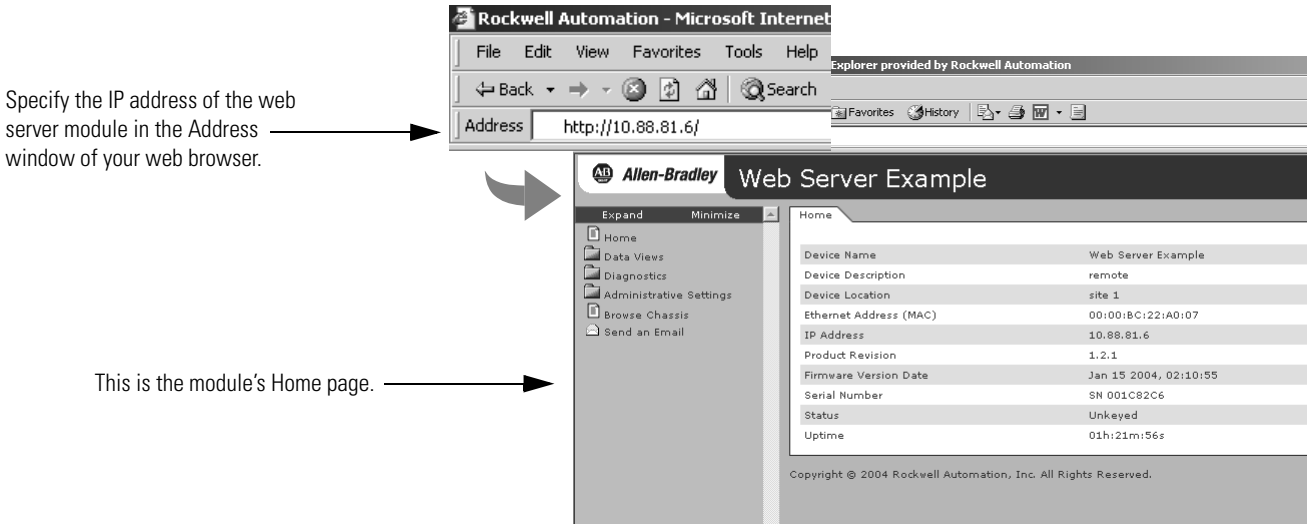
Assign Network Parameters When the Network Has a DHCP Server

By default, the web server module is DHCP enabled. DHCP (Dynamic Host Configuration Protocol) software automatically assigns IP addresses to client stations logging onto a TCP/IP network.

If you connect the web server module to a network that has a DHCP server, that server will assign an IP address to the web server module and the four-digit display on the front of the web server module will display each of the four numbers of the IP address.

In the Address field of your web browser, enter the IP address that displays on the front of the module.

The module home page displays.



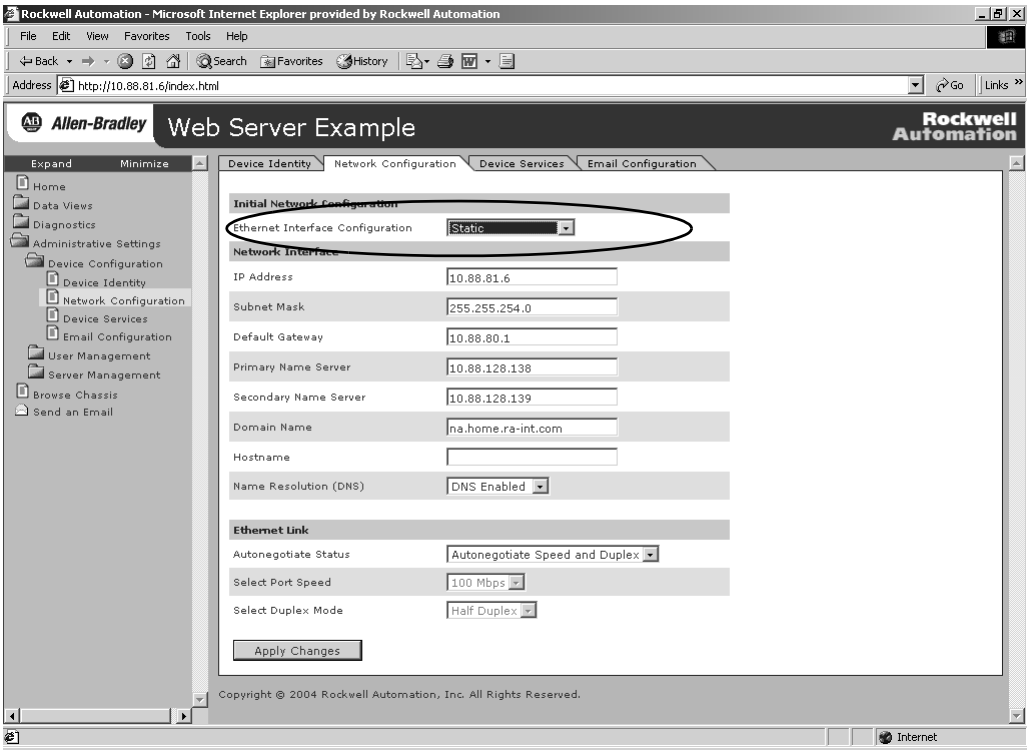
The IP address from the DHCP server provides initial access to the web server module. Check with your network administrator on whether you need to modify the IP address for future access to the module. The network administrator might have you:

- convert the initial IP address to a static IP address.
- enter a different, unique IP address and configure that new address as a static address.
- do nothing because the DHCP server was configured so that the initial IP address is already permanently assigned to the web server module.
- assign a static IP address.

If your network configuration requires a static IP address, configure the IP address by selecting Administrative Settings > Device Configuration > Network Configuration.

IMPORTANT Do not simply configure the initial address assigned by the DHCP server as a static IP address. Contact your network administrator for an appropriate static IP address.

1. Access the Network Configuration page.
2. Enter the static IP address
3. Select Static for the Ethernet Interface Configuration

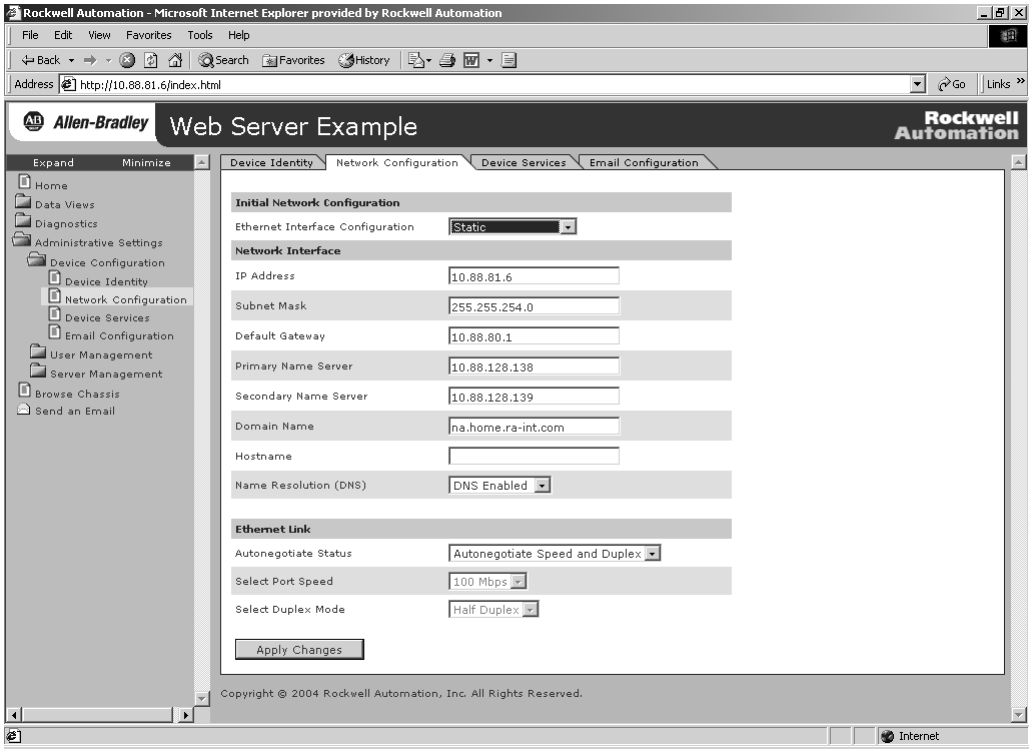


Assign Network Parameters Without A DHCP Server

If a DHCP server is not available, you must assign a static IP address to the module. Select one of these methods:

If You Are Working in These Conditions	Use This Method For Assigning Network Parameters	See Page
In any condition, the Rockwell Automation BOOTP/DHCP utility is recommended.	Rockwell BOOTP/DHCP utility (available with RSLinx and RSLogix 5000 software)	36
The module is connected to other NetLinx networks.	RSLinx software	38
The RSLogix 5000 project is online with the controller that communicates to or through the web server module.	RSLogix 5000 software	39

After using one of these utilities, select Administrative Settings > Device Configuration > Network Configuration to set additional parameters.



Use the Rockwell Automation BOOTP/DHCP Utility

The module ships with DHCP enabled. The BOOTP/DHCP utility is a stand alone program that lets you interactively define the IP address of a module that is issuing DHCP or BOOTP requests. The utility is located in the:

- BOOTP-DHCP Server folder in the Rockwell Software program folder on the Start menu.

  The utility is automatically installed when you install RSLinx software.

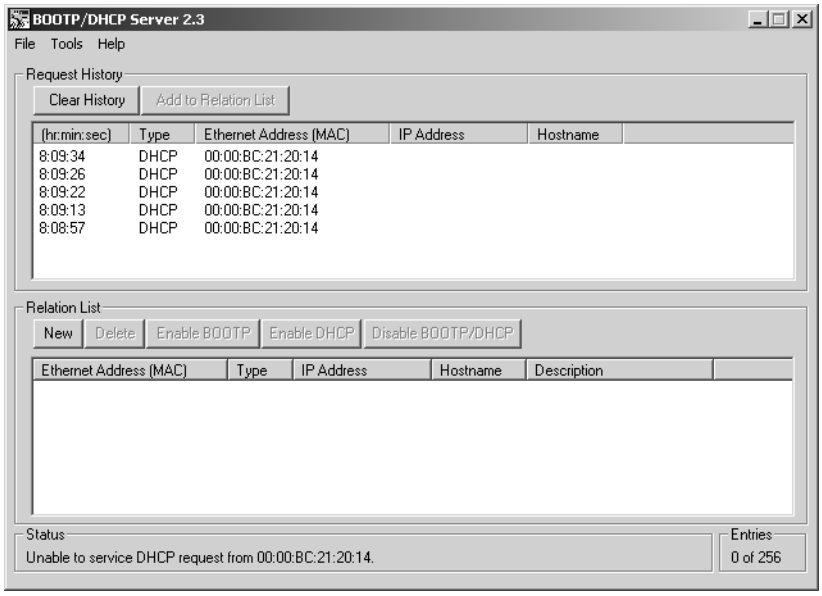
- Tools directory on the RSLogix 5000 installation CD.

IMPORTANT Before you start the BOOTP/DHCP utility, make sure you have the hardware (MAC) address of the web server module. The hardware address is on a sticker located on the side of the module. The hardware address in a format similar to: 00-0b-db-14-55-35.

To use the BOOTP/DHCP utility:

1. Launch the BOOTP/DHCP software.

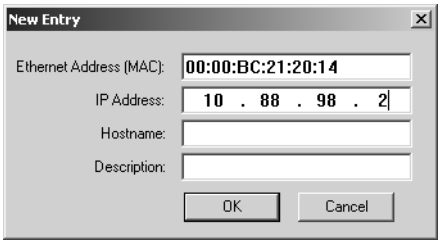
   In the Request History panel you see the hardware addresses of modules issuing requests.



2. Double-click on the hardware (MAC) address of the module you want to configure.

The hardware address is on a sticker located on the side of the web server module. The hardware address will be in a format similar to: 00-0b-db-14-55-35.

The New Entry window displays the MAC address you selected and prompts you to enter the IP address.

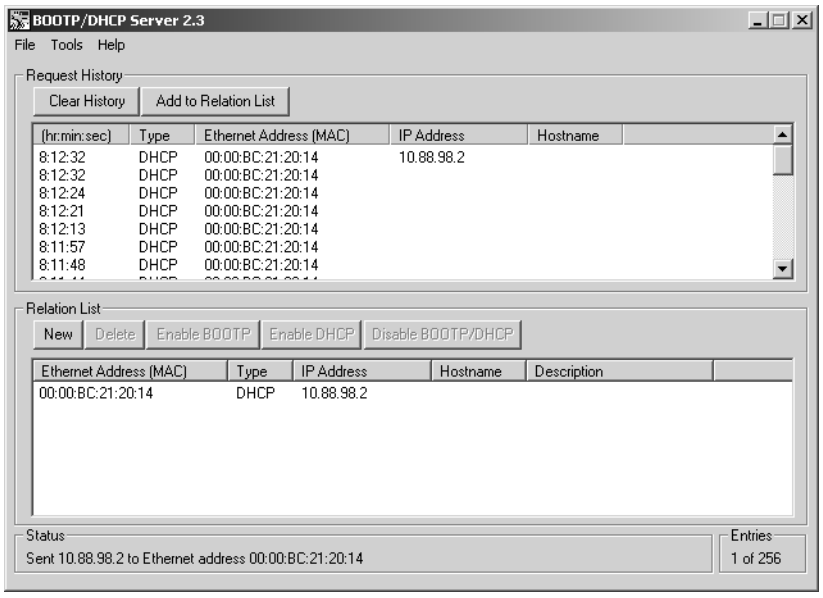


3. Enter the IP address of the module.

   You can also enter the host name and a description of the module.

4. Click OK.

   The device is added to the Relation List.



5. To permanently assign this configuration to the module, highlight the module and click on the Disable BOOTP/DHCP button.

   When power is recycled, the module uses the configuration you assigned and does not issue a request.

If you do not select the Disable BOOTP/DHCP button, on a power cycle, the web server module clears the current IP configuration and will again begin sending requests.

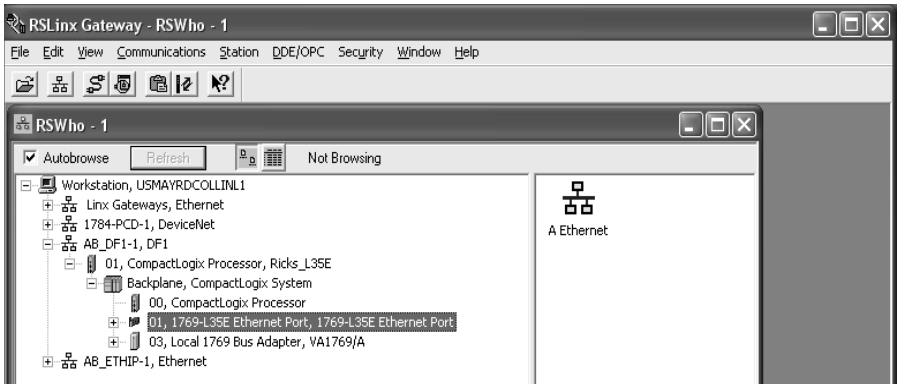
If you use the BOOTP/DHCP utility in an uplinked subnet where an enterprise DHCP server exists, the module may get an IP address from the enterprise server before the BOOTP/DHCP utility even sees the module. To avoid this, disconnect from the uplink to set the address and have the module remember its static address before reconnecting to the uplink. This is not a problem if you have node names configured in the module and leave DHCP enabled.

Use RSLinx Software to Configure the IP Address

To use RSLinx software to configure the IP address:

1. Make sure the web server module is installed and powered.
2. Start RSLinx software.
3. Click the RSWho icon.

4. Expand the network configuration organizer until you reach the module.
5. Right-click on the module and select Module Configuration.



6. Select the Port Configuration tab.

7. In the Network Configuration box, click Static to permanently assign the configuration.

   If you select Dynamic on a power cycle, the controller clears the current IP configuration and will again begin sending requests.

8. Enter the IP address and the other network parameters, if needed.

Use RSLogix 5000 Software to Configure the IP Address

To use RSLogix 5000 software to configure the IP address:

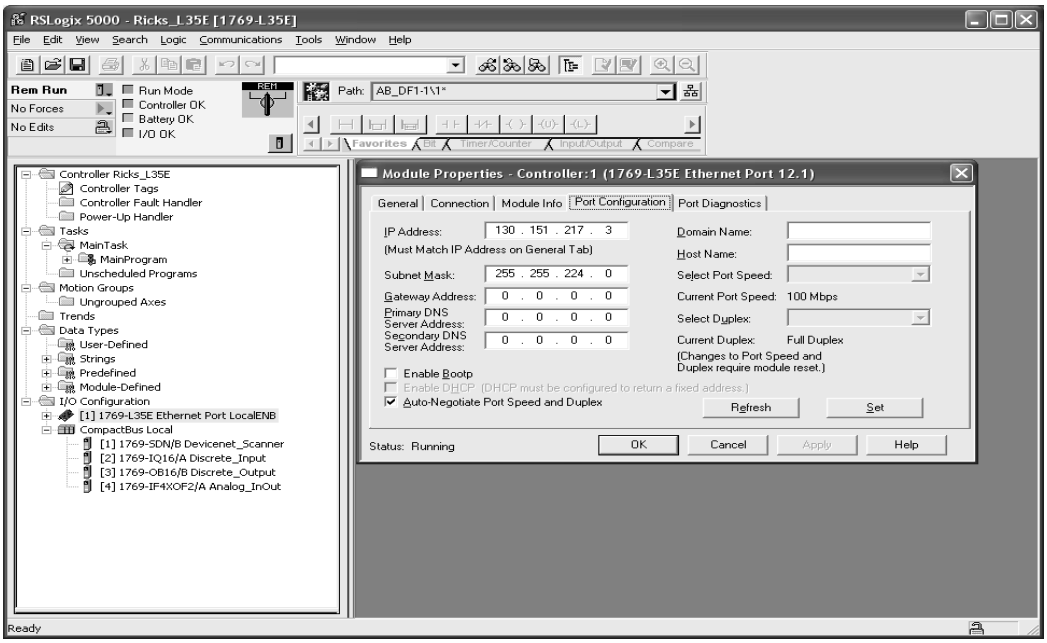
1. Make sure the module is installed and powered up.

2. Connect to the controller via a serial or other network connection.

3. Start RSLogix 5000 software.

4. In the Controller Organizer, select the EtherNet/IP module and right-click.

5. Select Properties.



6. Select the Port Configuration tab and specify the IP address and the other network parameters, if needed.

7. Click Apply.

8. Click OK.

   This sets the IP address in the hardware. This IP address should be the same IP address you assigned under the General tab.

   On this screen, you can also specify port speed (10 Mbps or 100 Mbps) and duplex mode (autonegotiate, half duplex, or full duplex). The module configuration needs to agree with how the switch is configured. See your network administrator for more information.

Duplicate IP Address Detection

1756-EWEB modules with firmware revision 2.2 or later support duplicate IP address detection.

All 1768-EWEB module firmware revisions support duplicate IP address detection.

For more information about EtherNet/IP modules that support duplicate IP address detection, see the EtherNet/IP Modules in Logix5000 Control Systems User Manual, publication ENET-UM001.

When you change the IP address or connect a web server module to an EtherNet/IP network, the module checks to make sure that the IP address assigned to this module is not the same as that for any other device already on the network.

If the module determines that there is a conflict (some other device on the network already has the IP address), the EtherNet/IP port of the module goes into conflict mode, where the module's:

- OK LED indicator blinks red.
- network (NET) LED indicator is solid red.
- front display indicates the conflict.

  The display scrolls: OK <IP_address_of_this_module> Duplicate IP <Mac_address_of_duplicate_node_detected>

  For example: OK 10.88.60.196 Duplicate IP - 00:00:BC:02:34:B4

To correct this conflict, use the instructions in this chapter to change the IP address of the module. Then cycle power to the module or reset the module (such as disconnecting the EtherNet/IP cable and reconnecting the cable).

There is also the possibility that two modules can detect a conflict simultaneously. If this occurs, remove the module that has the incorrect IP address or correct its conflict. To get the second module out of conflict mode, cycle power to the module or disconnect its EtherNet/IP cable and reconnect the cable.

Duplicate Detection Scenarios

The behavior of devices that are in conflict over an IP address varies, depending on whether connections have been established to either of the modules and whether both modules support duplicate IP address detection.

If	Then
Both modules support duplicate IP address detection	The module that powers up first and uses the IP address keeps the IP address. The other module will detect a conflict, give up the IP address, and enter conflict mode.
Both modules support duplicate IP address detection and both modules power up at roughly the same time	Both modules give up the IP address and enter conflict mode.
One module supports duplicate IP address detection and a second module does not	the second module generally keeps its IP address, regardless of which module obtains the IP address first. The module that supports duplicate IP address detection will detect the conflict and give up the IP address.

IP Address Swapping

1756-EWEB modules with firmware revision 2.2 or later support IP address swapping.

During a switchover in ControlLogix redundancy systems, these modules swap their IP addresses with their partner modules in the other redundant chassis.

For more information about IP address swapping, see the ControlLogix Redundancy User Manual, publication 1756-UM523.

DNS Addressing

To further qualify an address of a module, you can use DNS addressing to specify a host name for a module, which also includes specifying a domain name and DNS servers. DNS addressing lets you set up similar network structures and IP address sequences under different domains.

DNS addressing is necessary only if you refer to the module by host name and use the web server module to initiate MSG instructions out of the web server module to another device.

To use DNS addressing, you must:

1. Assign a host name to the module.

Your network administrator should be able to assign a host name. Valid host names should be IEC-1131-3 compliant.

2. Configure the module parameters.

   In addition to the IP address, subnet mask, and gateway address, you must also configure a host name for the module, domain name, and primary/secondary DNS server addresses. In the DNS server, the host name must match the IP address of the module.

IMPORTANT Make sure the DNS enable bit is set.

If you configure your module using RSLinx software, version 2.41.00 or later, the enable bit is cleared and DNS addressing will not work. If you configure your module using the Port Configuration tab in RSLogix 5000 software, the enable bit is set, so DNS addressing should work.

3. In RSLogix 5000 software, add the module to the I/O configuration tree and enter the host name in the General tab of the module.

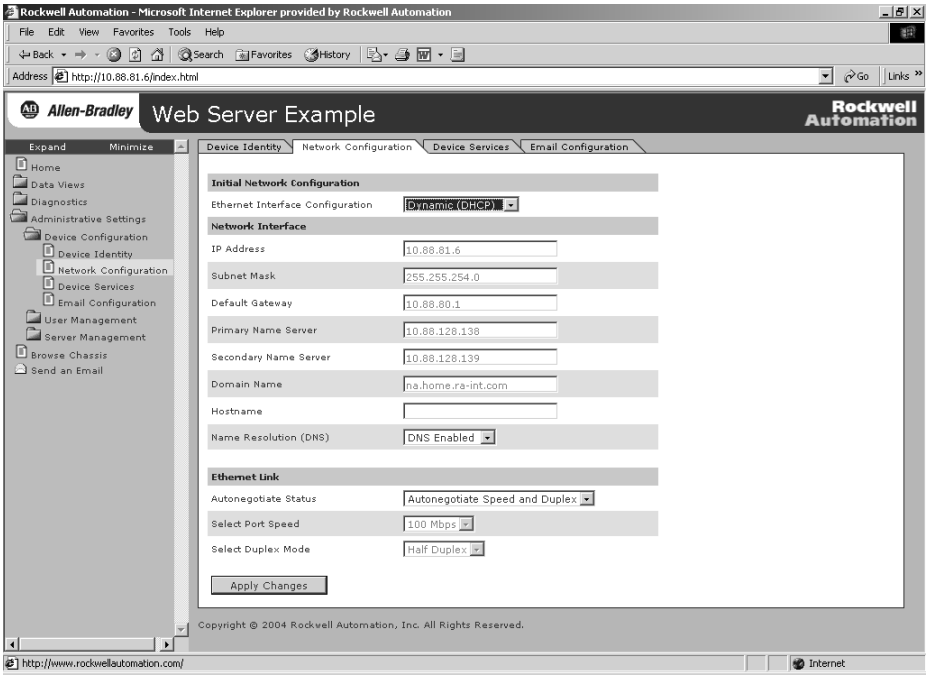
   If a child module resides in the same domain as its parent module, just enter the host name. If the child module is in a different domain that its parent module, you must enter the host name and the domain name (host.domain)

You can also use DNS addressing in a module profile in the I/O controller tree or in a message path. If the domain name of the destination module is different from the source module, use a fully-qualified DNS name (hostname.domainname). For example, to send a message from ENBT1.location1.companyA to ENTB1.location2.companyA, the host names are the same, but the domains are different.

If you do not enter a fully-qualified DNS name, the module appends the default domain name to the specified host name.

Verify Network Settings

Select Administrative Settings > Device Configuration > Network Configuration. An authenticated user may modify network parameters.



In This Field	Specify
Ethernet Interface Configuration	The network configuration scheme: • Dynamic BOOTP • Dynamic DHCP (default) • Static
IP Address	IP address for the web server module: If you want to specify a static IP address for the web server module, you must also select Static for the Ethernet Interface Configuration field towards the bottom of this page.
Subnet Mask	Subnet mask for the web server module.
Default Gateway	Gateway address for the web server module.
Primary Server Name Secondary Server Name	DNS server names, if using DNS addressing.
Domain Name	Domain name for the web server module, if using DNS addressing.
Host Name	Host name for the web server module, if using DNS addressing.
Name Resolution (DNS)	Whether the web server module uses DNS addressing.

In This Field	Specify
Autonegotiate Status	How to determine port speed and duplex: • Autonegotiate speed and duplex • Force speed and duplex
Select Port Speed	Port speed (10 Mbps or 100 Mbps), if you selected to force speed and duplex.
Select Duplex Mode	Duplex (full or half), if you selected to force speed and duplex.

Additional Resources

Consult the following publications for more information.

Topic	Publication Title	Publication No.
EtherNet/IP modules and networks.	EtherNet/IP Modules in Logix5000 Control Systems User Manual	ENET-UM001
EtherNet/IP modules, IP swapping, and redundancy systems.	ControlLogix Redundancy User Manual	1756-UM523

You can view or download publications at http://www.literature.rockwellautomation.com. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

Manage Module Settings

How to Use This Chapter

This chapter describes how to configure module settings other than network parameters for the web server module.

Manage Module Information

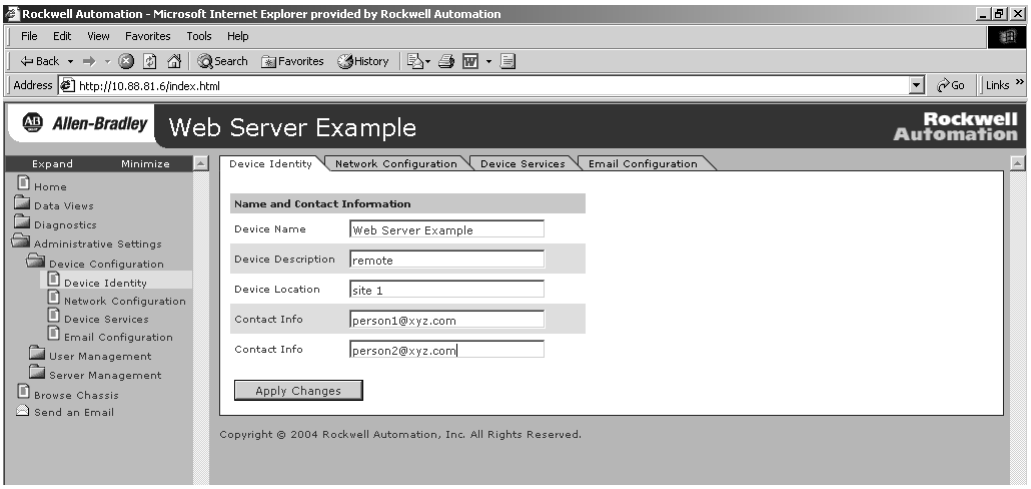
To access and modify module-specific information, select Administrative Settings > Device Configuration from the organizer on the left. You can:

- define the module-specific information that displays on the Home page.
- modify network parameters.
- enable/disable communication services.

These settings are stored in flash memory and persist over power cycles.

Define Module-specific Information For the Home Page

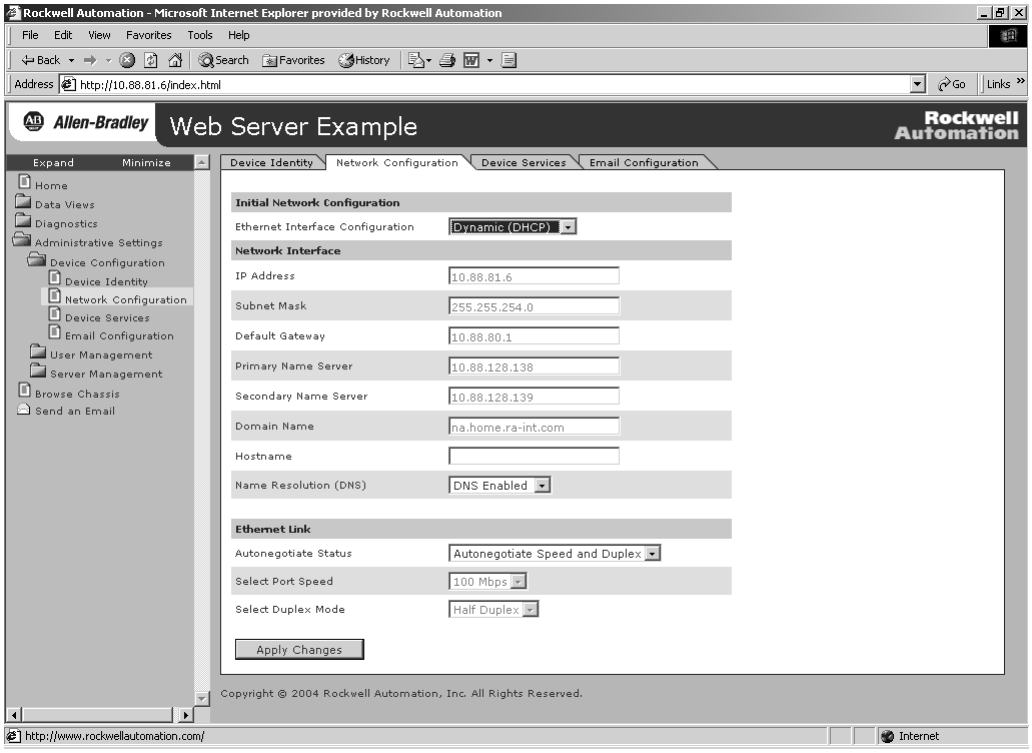
Select Administrative Settings > Device Configuration > Device Identity to set specific text that identifies the module. This information appears on the home page.



In This Field	Specify
Device Name	A name for the web server module (32 characters maximum) The device name you enter appears in the title bar of the web server module's web pages. This device name also appears in RSLinx when you browse the network.
Device Location	Description of the location of the web server module (64 characters maximum)
Device Description	Description of the web server module (64 characters maximum)
Contact Information	Contact information, such as name, phone number, or email address (512 characters maximum) There are two fields so that you can specify contact information for two individuals.

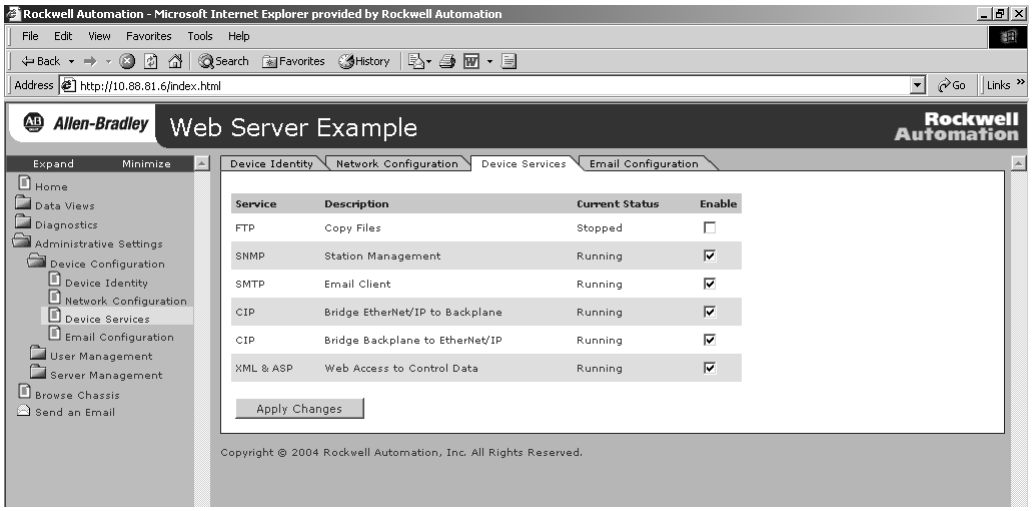
Modify Network Parameters

Select Administrative Settings > Device Configuration > Network Configuration lets to modify network parameters.



Enable and Disable Communication Services

Select Administrative Settings > Device Configuration > Device Services to specify which communication services are enabled or disabled on the web server module.



In This Field	Select Whether To Enable Or Disable the
FTP	FTP (File Transfer Protocol) server Disable FTP to prevent users from accessing the file system on the web server module. **Important:** For security purposes, keep FTP disabled unless you frequently transfer files to or from the web server module.
SNMP	SNMP (Simple Network Management Protocol) agent Enable SNMP if your system uses SNMP management software.
SMTP	SMTP (Simple Mail Transfer Protocol) agent SMTP manages email capability. Disable SMTP if you do not send emails from the web server module.
CIP Bridge Ethernet to Backplane	CIP (Common Industrial Protocol) bridging Enable this CIP bridging to allow EtherNet/IP devices to bridge through the web server module to devices in the chassis.
CIP Bridge Backplane to Ethernet	CIP (Common Industrial Protocol) bridging Enable this CIP bridging to allow other devices in the chassis to bridge through the web server module to EtherNet/IP devices.
XML/ASP	XML/ASP (Extended Markup Language/Active Server Page) support Enable XML/ASP to allow web access to control system data.

Manage Server Settings

Select Administrative Settings > Server Management to customize some of the server settings of the module, as well as back up the file system on the web server module. You can:

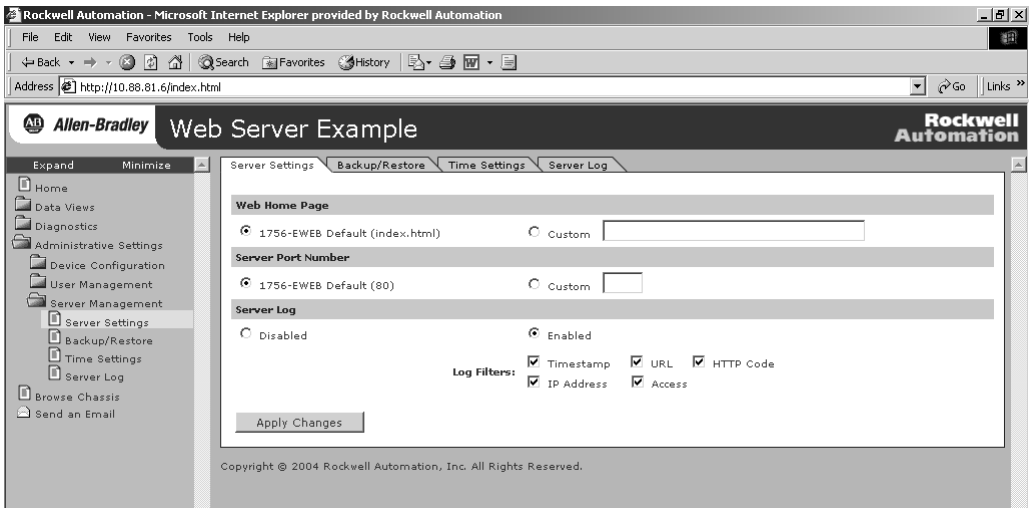
- customize server settings, including web home page.
- lock access to the module during backup or restore procedures.

  See chapter 6 for more information on backing up the web server module.

- configure the time server.
- display a server log.

Customize Server Settings

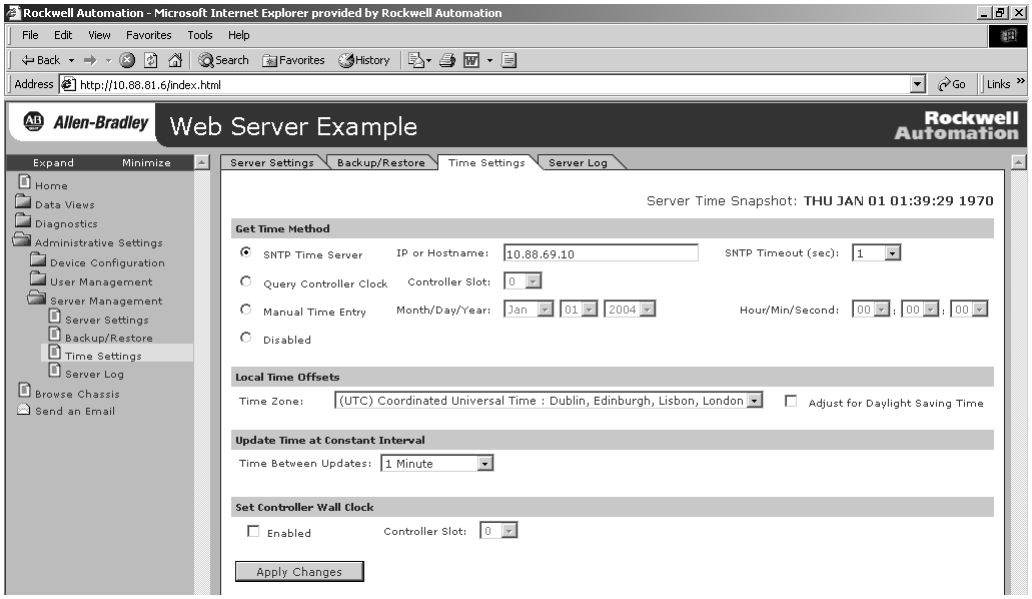
Select Administrative Settings > Server Management > Server Settings to customize the web home page and server settings of the web server module.



In The Field	Take This Action
Web Home Page	Select which home page is the default, 1756-EWEB Default (index.html) or select and specify a custom home page address. For example, a custom web page could be: /user/Web/mypage.html mypage is the name of the file for the custom web page. You must copy a custom home page to the web server module before you can use it. See chapter 8 for information on creating a custom web page.
Server Port Number	Select the default port number (80) for the HTTP port on the web server module or specify a custom port number.
Server Log	Enable or disable the server log. You view the server log from the Server Log page under the Server Management folder. See page 54 for how to display the server log.
Log Filters	If you enable the Server Log, specify which of the following information you want to be recorded for the web server log. Timestamp of HTTP request (access) URL requested on the web server module Server HTTP Code IP Address of the requestor Access (Administrator, Write, or Read)

Configure the Time Server

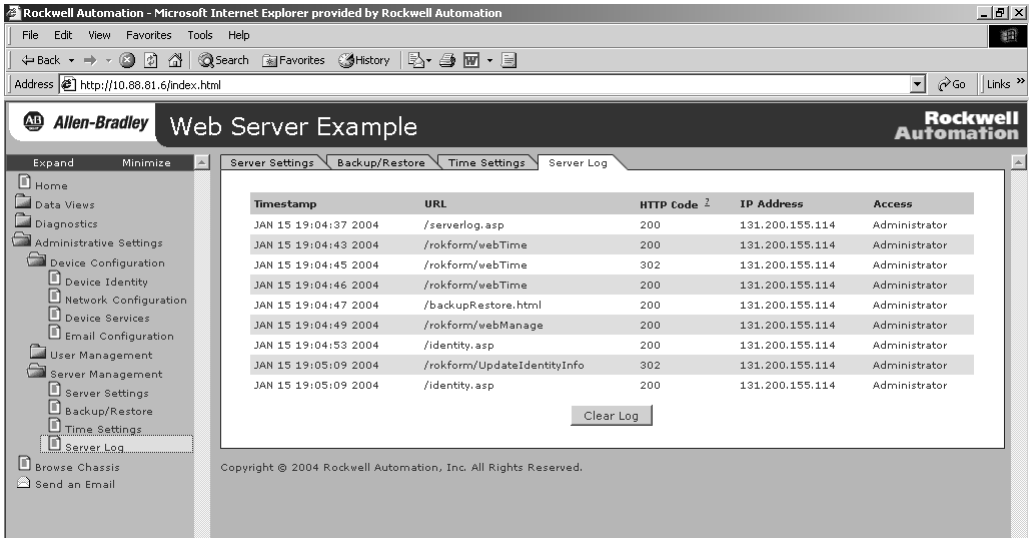
To configure the Time Server, select Administrative Settings > Server Management Server > Time Settings. Doing so helps makes sure that files you save to the web server module have accurate date and time stamps. You can also enable the local controller to get its time and date from the web server module.



Select This Field	If You Want To
SNTP Time Server	Use the time from the Network Time Protocol (NTP). Specify the IP address or host name of an SNTP server on the network. The web server uses port 123 for this service. The IP address you enter persists over power cycles.
Query Controller Clock	Use the time from the local Logix controller (Wall Clock Time). Specify the slot number of the controller. The web server module queries the Wall Clock Time of the controller for both time and date. At subsequent power ups, the web server module queries the controller.
Manual Time Entry	Manually set the time and date. Manual settings do not persist over power cycles.
Local Time Offsets	Select the appropriate time zone. This selection is only available when you select SNTP time server as your "Get Time Method."
Update Time at Constant Interval	Select how often the web server module updates its date and time.
Set Controller Date/Time	Use the date and time in the web server module to set the date and time in the local controller. You must also specify the slot number of the local controller. **Important:** This feature provides accurate time synchronization to within one second.

Display the Server Log

To display the server log, select Administrative Settings > Server Management Server > Server Log. This page, when enabled, displays records of web accesses to the web server module. Only those information fields that are enabled on the Server Setting page (see page 51) appear in the server log. The information displayed on this page is stored in RAM and does not persist over power cycles.



This Field	Specifies
Timestamp	Timestamp of HTTP request (access).
URL	Requested URL on the web server module.
HTTP code	HTTP code request.
IP address	IP address of the requestor.
Access	Type of access.

The web server module has 30 K memory allocated for server log entries. If all the log options are enabled, the server log memory can hold about 200 entries. Once this allocation is full, the web server module stops storing server log entries. Click Clear Log to empty the server log so that the web server module can again log entries.

Use Data Views to Access Controller Data

About This Chapter

The module provides access to controller data for monitoring and data modification of controller tags.

This chapter shows you how to set up data views of controller tags.

For data views, the module must be in the same chassis as the controller.

Data Views Overview

Data views give you the ability to read from and write to RSLogix controller tags from a browser interface or an external application. The module provides web pages that let you configure a set of tags (a data view) that can be read or written.

A data view consists of an XML file with data tag information The XML file is in a readable ASCII format. It contains the tag name, data type, path, display formatting, and privilege access level. Each tag value is exposed as a separate element and an error attribute is optional.

Tags Supported In Data Views

To configure tags in data views:

- You can only access tags in controllers that reside in the local chassis (same chassis as the web server module).
- Tags must be controller-scoped.
- Tags must be an atomic type (BOOL, SINT, INT, DINT, REAL, STRING).

  You can specify a member of a structure or an array, but you cannot specify an entire structure or array. BOOL arrays are not supported.

- A tag can appear only once in a particular data view. You cannot, for example, have two instances of the same tag with different display formats.
- There is no limit to the number of data views as long as the total number of entries in all data views on one web server module does not exceed 2500 entries.
- Each tag you configure on a data view is one entry. If you configure the same tag in multiple data views, each tag is considered one entry.

Performance Estimates

For access to the XML data views, the module can produce data according to the values listed in this table. This table assumes the absence of significant CIP traffic and does not take into account the amount of the time for the browser to render the data view page.

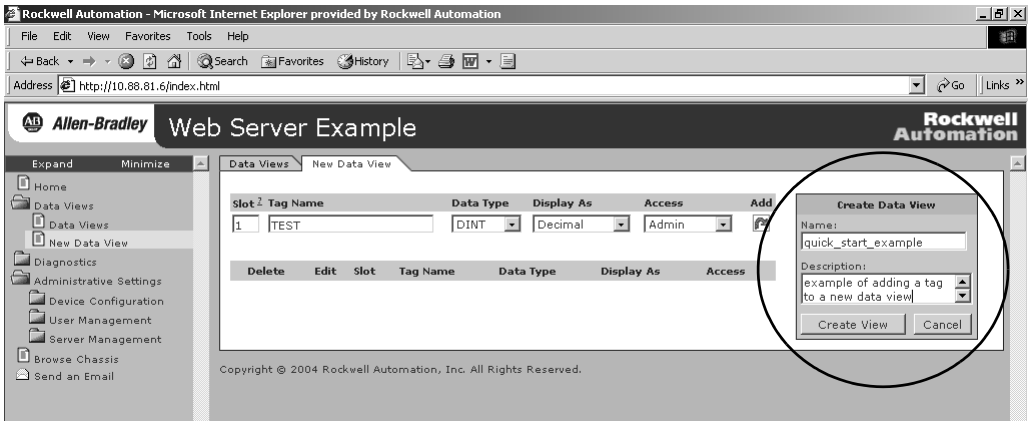
Tags Per Data View	Time Per Data View
10	100 ms
100	350 ms
1000	3 sec

Create a Data View

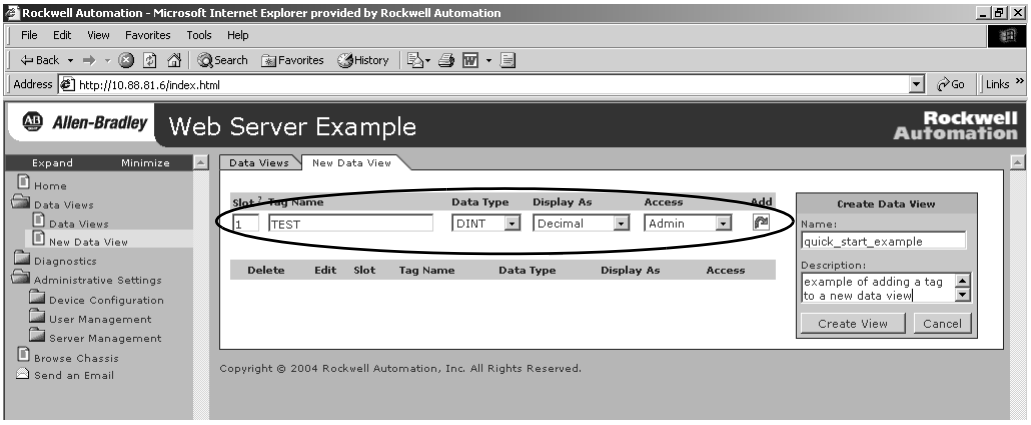
Each data view contains a group of tags that you want to monitor. Each module can support multiple data views.

You create a data view by selecting Data Views > New Data View.

1. Use the Create Data View window on the right of the window to enter a data view name (required) and description (optional).



2. Add at least one tag to the data view.



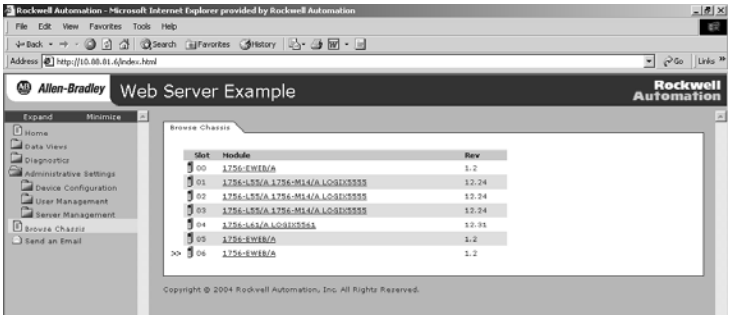
3. Click the Add button  to add the tag you just specified.

   You can add multiple tags to the data view, as long as there are no more than 2500 tags in all the data views of one web server module.

4. Click the Create View button to create the data view.

Add Tags to a Data View

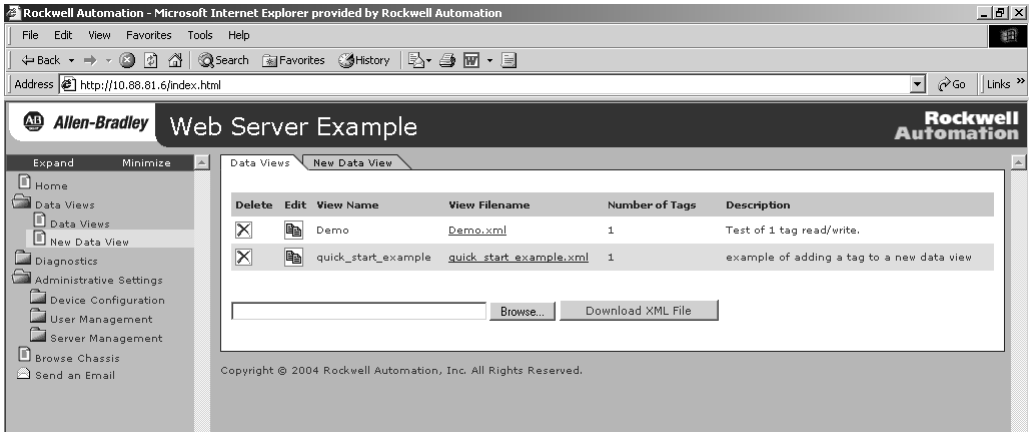
When you add a tag to a data view, you specify the following information.

In This Field	Specify	Details
Slot	Slot number of the controller	Click on the question mark next to Slot  -OR- Use the Chassis Browse page to validate the controller slot. 
Tag Name	Name of the tag	These fields must match exactly what is specified for the tag in the selected controller. To verify tag information, you can use: • RSLogix 5000 software to view the controller project. • RSLinx software to navigate to the controller and view tags.
Data Type	Data type of the tag	
Display As	Display type to use for the tag	
Access	Whether you require Administrator, Write, or Read access to view the tags in this data view	The default access level is Administrator. The access you specify applies to the whole data view, not just the tag. If you have multiple tags with different access levels in the same data view, the web server assigns the highest (most access) level to the data view. See chapter 6 for details on access levels.

Monitor Data Views and Tag Data

Select Data Views > Data Views to view existing data views.

Click on the file name to view the tags within a data view.



The data view displays in an XML format using an XSL style sheet. To quickly access the XML file, right-click in the data view and in:

- Internet Explorer, select View Source and save the resulting text.
- Netscape or Mozilla, select This Frame > Save As.

You can also use the backup/restore function to FTP a copy of the XML file. See chapter 7.

If the fields specified for the tag do not match the tag as it is specified in the controller, this page indicates an error and the tag value shows xsi:nil for its value.

From this page, you can modify the value of a tag if you have Administrator or Write access. Enter the new value and click Update.

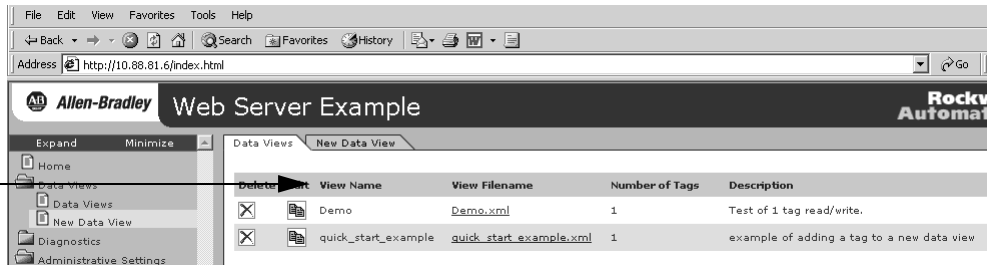
TIP To avoid impacting controller execution, data view pages do not auto-refresh.

Sort Data Views

You can sort data views alphabetically by name, filename, or description, or numerically by number of tags.

1. Click a column name.

Click a column heading to sort the column.



The first click sorts in ascending order.

2. Click again to sort in descending order.

   An arrow next to the column name shows the direction of the current sort.

   You can also sort the tags within a data view by clicking on the slot, tag name, data type, display as, value, or access headings in the column title.

Interface with the Logix Controller

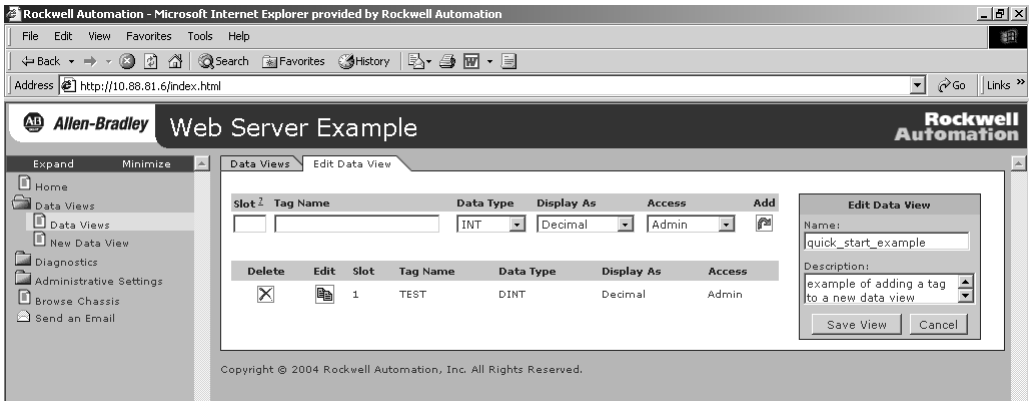
When you request to display a data view, the module establishes one connection to the target controller. Tag values are read and written over this connection. After the module retrieves the data view or updates the data view, the module closes the connection.

If someone changes tag names in the target controller and does not update the tags in the data view, the data view will display an error message indicating that the tag was not available.

Edit a Data View

Follow this procedure to edit an existing data view.

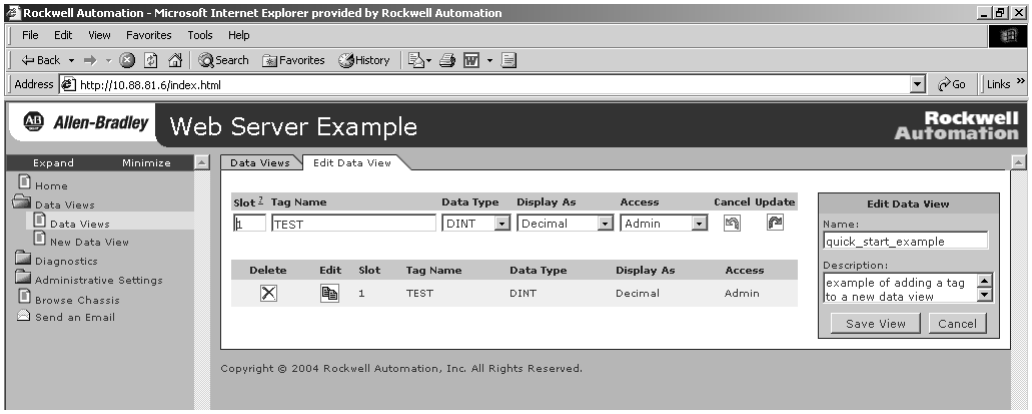
1. Click the edit symbol  next to the data view you want to edit.



At this screen, you can add additional tags or edit existing tags.

2. To edit an existing tag, click the edit symbol  next to the tag name.

3. When you edit a tag, the tag you selected is highlighted and the tag fields are populated with the previously-configured information.



4. Change entries in the fields to meet your needs.

Create Data Views Offline

You can create data views offline as XML files and later copy them into the web server module. To create a data view offline:

1. Use a text editor to create an XML data view file.

   Right-click in the data view and in:

   - Internet Explorer, select View Source and save the resulting text.
   - Netscape or Mozilla, select This Frame > Save As.

   You can also use the backup/restore function to FTP a copy of the XML file. See chapter 7.

2. Scroll to the bottom of the Data Views page on the web server module.
   a. Use the Browse button to locate the XML data view file.
   b. Use the Download XML File button to copy the XML data view file to the module.

   Data views are stored in the /user/system/dataviews/ directory on the web server module.

Use an External Application to Access Data Views

The XML format of data views makes the data views files accessible by user-written programs. Many programming languages, such as Java and Visual Basic, can process XML files.

User programs access data views by making HTTP requests. This is just like a web browser, except instead of displaying the data view, the user program processes the XML data. The browser uses an XSL stylesheet to display the XML files. The XSD schema files validate data views.

File Format	Description
XSL	An XML data view specifies an external XSL stylesheet that contains the rules for transforming this XML information into HTML. A web browser uses the XSL stylesheet to display the data view. The XSL file is stored in address/dataview/dataview.xsl where address is the IP address or host name of the web server module.
XSD	The web server module provides an XML schema (dataview.xsd) for validating data views. This schema also references the CIPDataTypes.xsd schema. The XSD files are stored in address/schema/dataview.xsd and address/schema/CIPDataTypes.xsd where address is the IP address or host name of the web server module.

Read a Data View with an External Application

For an external application to read a data view, the application issues an HTTP GET command that specifies the location and filename of the data view.

Data views are located in the /user/system/dataviews directory. The URL for a data view named myview would be:

http://IP_address/user/system/dataviews/myview.xml

Change Data In a Data View with an External Application

When an external application completes modifying tag data in a data view, it should post the modified data view, either as a file attachment (in a multi-part form) or in a single form field named xml, to the URL of the data view itself.

If all the modified tags are successfully written, the web server module redirects the application to the newly modified data view. If any tag cannot be written to the controller, the web server module returns an HTTP error code with a status message indicating the error.

Example: Data View XML

This is an example XML markup for a data view named alltypes. The data view contains one tag for each of the supported data types. The tags are in the controller residing in slot 1.

```
<?xml version="1.0"?>
<?xml-stylesheet href="/dataview/dataview.xsl" type="text/xsl"?>
<view
xmlns="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.rockwellautomation.com/technologies/data_access/data_
views/1.0/ /schema/dataview.xsd"
xmlns:cip="http://www.rockwellautomation.com/technologies/data_access/data_types/1.0
/" name="alltypes" description="">
   <tag name="test_tag_bool" valueType="cip:dt_BOOL" path="1,1" display="String"
access="write">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_sint" valueType="cip:dt_SINT" path="1,1" display="Decimal"
access="write">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_int" valueType="cip:dt_INT" path="1,1" display="Decimal"
access="write">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_dint" valueType="cip:dt_DINT" path="1,1" display="Decimal"
access="write">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_real" valueType="cip:dt_REAL" path="1,1" display="Decimal"
access="write">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_string" valueType="cip:dt_STRINGI" path="1,1"
display="String" access="write">
      <value xsi:nil="true"/>
   </tag>
</view>
```

Example: Data View XML with Tag Values

This is an example XML markup for a data view named alltypes loaded with current tag values. The data view contains one tag for each of the supported data types. The tags are in the controller residing in slot 1.

```
<?xml version="1.0"?>
<?xml-stylesheet href="/dataview/dataview.xsl" type="text/xsl"?>
<view
xmlns="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.rockwellautomation.com/technologies/data_access/data_
views/1.0/ /schema/dataview.xsd"
xmlns:cip="http://www.rockwellautomation.com/technologies/data_access/data_types/1.0
/" name="alltypes" description="">
   <tag name="test_tag_bool" valueType="cip:dt_BOOL" path="1,1" display="String"
access="write">
      <value xsi:type="cip:dt_BOOL">TRUE</value>
   </tag>
   <tag name="test_tag_sint" valueType="cip:dt_SINT" path="1,1" display="Decimal"
access="write">
      <value xsi:type="cip:dt_SINT">123</value>
   </tag>
   <tag name="test_tag_int" valueType="cip:dt_INT" path="1,1" display="Decimal"
access="write">
      <value xsi:type="cip:dt_INT">28416</value>
   </tag>
   <tag name="test_tag_dint" valueType="cip:dt_DINT" path="1,1" display="Decimal"
access="write">
      <value xsi:type="cip:dt_DINT">1459879936</value>
   </tag>
   <tag name="test_tag_real" valueType="cip:dt_REAL" path="1,1" display="Decimal"
access="write">
      <value
xsi:type="cip:dt_REAL">-247882776235710380000000000000000000.000000</value>
   </tag>
   <tag name="test_tag_string" valueType="cip:dt_STRINGI" path="1,1"
display="String" access="write">
      <value xsi:type="cip:dt_STRINGI">aazz</value>
   </tag>

</view>
```

Example: Data View XML with Tag Errors

This example a data view named alltypes with error messages for tags that could not be retrieved.

```
<?xml version="1.0"?>
<?xml-stylesheet href="/dataview/dataview.xsl" type="text/xsl"?>
<view
xmlns="http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.rockwellautomation.com/technologies/data_access/data_
views/1.0/ /schema/dataview.xsd"
xmlns:cip="http://www.rockwellautomation.com/technologies/data_access/data_types/1.0
/" name="alltypes" description="">
   <tag name="test_tag_bool" valueType="cip:dt_BOOL" path="1,1" display="String"
access="write" error="Couldn't read tag!">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_sint" valueType="cip:dt_SINT" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_int" valueType="cip:dt_INT" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_dint" valueType="cip:dt_DINT" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_real" valueType="cip:dt_REAL" path="1,1" display="Decimal"
access="write" error="Couldn't read tag!">
      <value xsi:nil="true"/>
   </tag>
   <tag name="test_tag_string" valueType="cip:dt_STRINGI" path="1,1"
display="String" access="write" error="Couldn't read tag!">
      <value xsi:nil="true"/>
   </tag>
</view>
```

Send Email

About This Chapter

This chapter describes how to send an email message.

For email, the web server module can be located locally or remotely to the controller chassis.

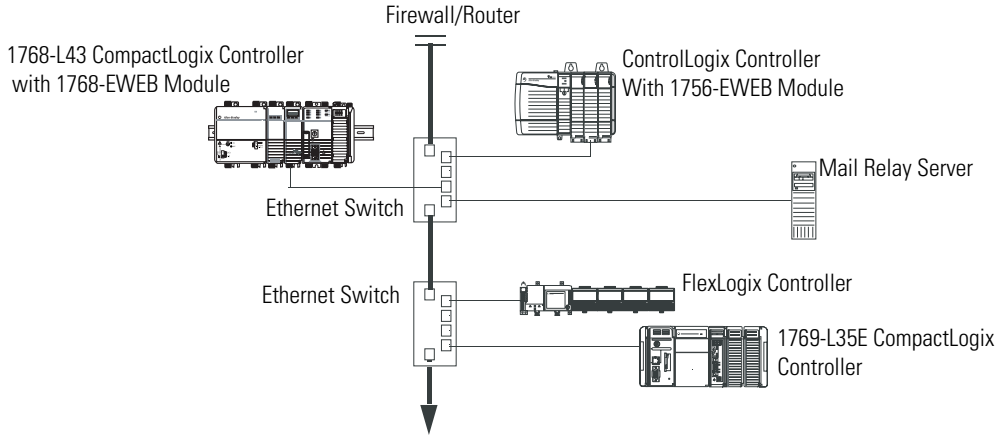
Overview

The web server module is an email client that uses a mail relay server to send email. There are two ways you can use the web server modules to send email.

If You Want To	Then
Send an email to specific personnel when a controller application generates an alarm or reaches a certain condition.	Program the controller to send a MSG instruction to the web server module. The MSG instruction then instructs the web server module to send the email text (contained within the MSG instruction) to the mail relay server. Multiple controllers can use the same web server module to initiate email.
Send controller or application status information on a regular basis to a project manager.	
Test the email configuration of the web server module.	Use the Send an Email link on the web server's Home page.
Use the web server email interface to send an email (you must enter all email information each time you use this interface).	

The web server module only sends the content of a MSG instruction (or the content of the message entered on the email web page) as an email to a mail relay server. Delivery of the email depends on the mail server. The web server module does not receive email.

See the following sample system.



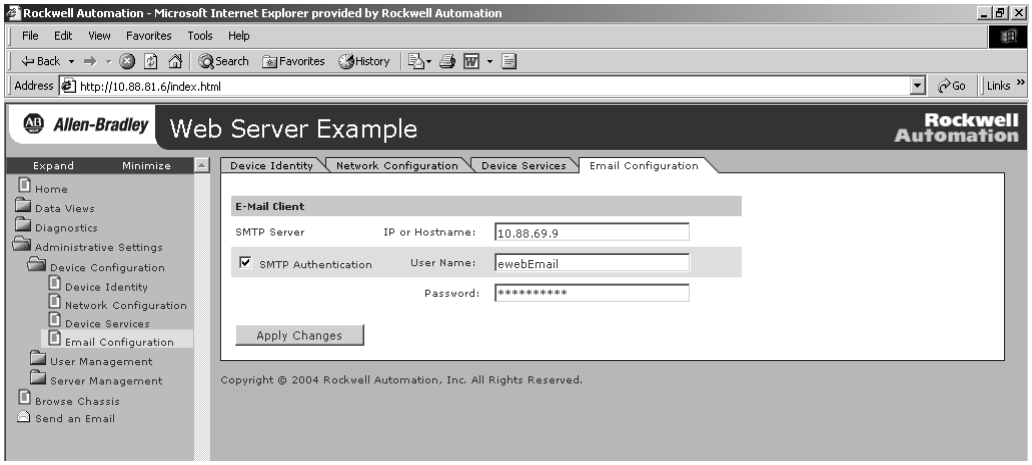
This Device	Can
ControlLogix controller	Send a MSG instruction to the 1756-EWEB web server module to initiate sending an email to the mail relay server. Use the path of the MSG instruction to identify the web server module as the target of the MSG instruction.
FlexLogix controller	
1768 or 1769 CompactLogix controller	
1756-EWEB or 1768-EWEB module	Send an email to the mail relay server from the email interface on the Send an Email link. Each time you use this interface, you must enter all email information.
Mail relay server	Send email to specified recipients. The mail relay server determines the delivery of any email send through a web server module, whether via a MSG instruction or from its built-in interface.

Configure the Web Server to Send Email

The web server module uses the standard SMTP protocol to forward an email to a mail relay server. You must configure the web server module to recognize the appropriate mail relay server.

Some mail servers require a domain name be provided during the initial handshake of the SMTP session. For these mail servers, make sure you specify a domain name when you configure the network settings for the module.

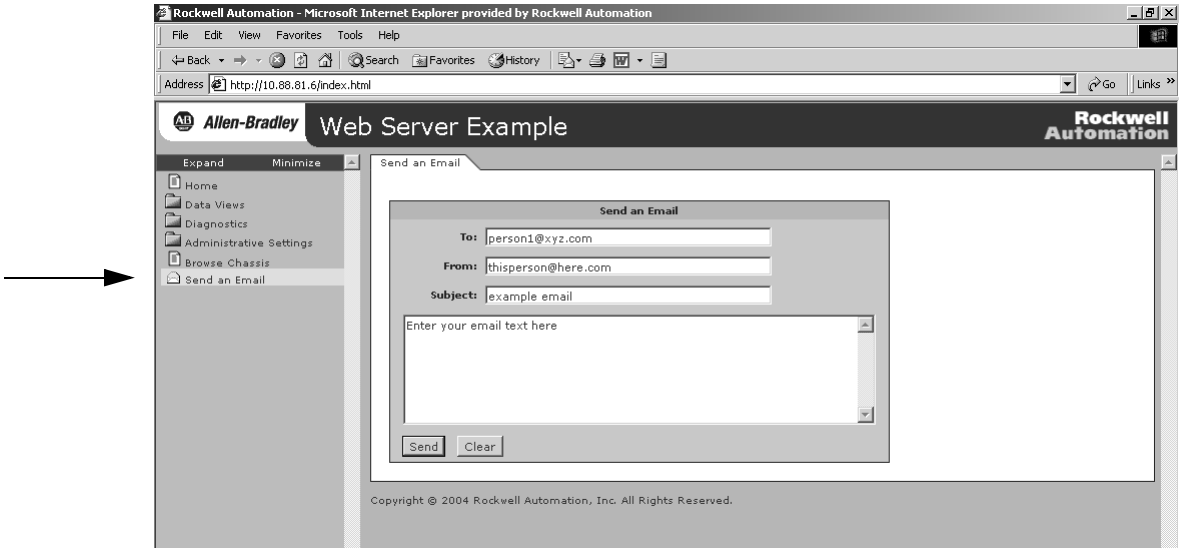
You configure the SMTP server and domain name by selecting Administrative Settings > Device Configuration > Email Configuration.



Enter the address of the SMTP server that manages email. You can also select whether the web server module should authenticate to the SMTP server. The web server module supports only LOGIN authentication. Check with your network administrator for more information.

Send an Email Via the Web Page

Use the Send an Email link to enter and send email text. This method is a one-time approach to sending an email because you have to enter all the email information each time you use the link. This link is most useful for testing the email configuration you specified on the Administrative Settings > Device Configuration > Email Configuration.



In This Field	Enter the
To	Email address of the recipient.
From	Email address of the sender. This address is where you want any replies to this email to go. It is not an email address of the web server module. The web server module only sends email and does not receive email.
Subject	Subject line of the email.
Text window	The email text.

Click Send after you specify the email address and enter the text.

Send an Email with a Controller-initiated Message Instruction

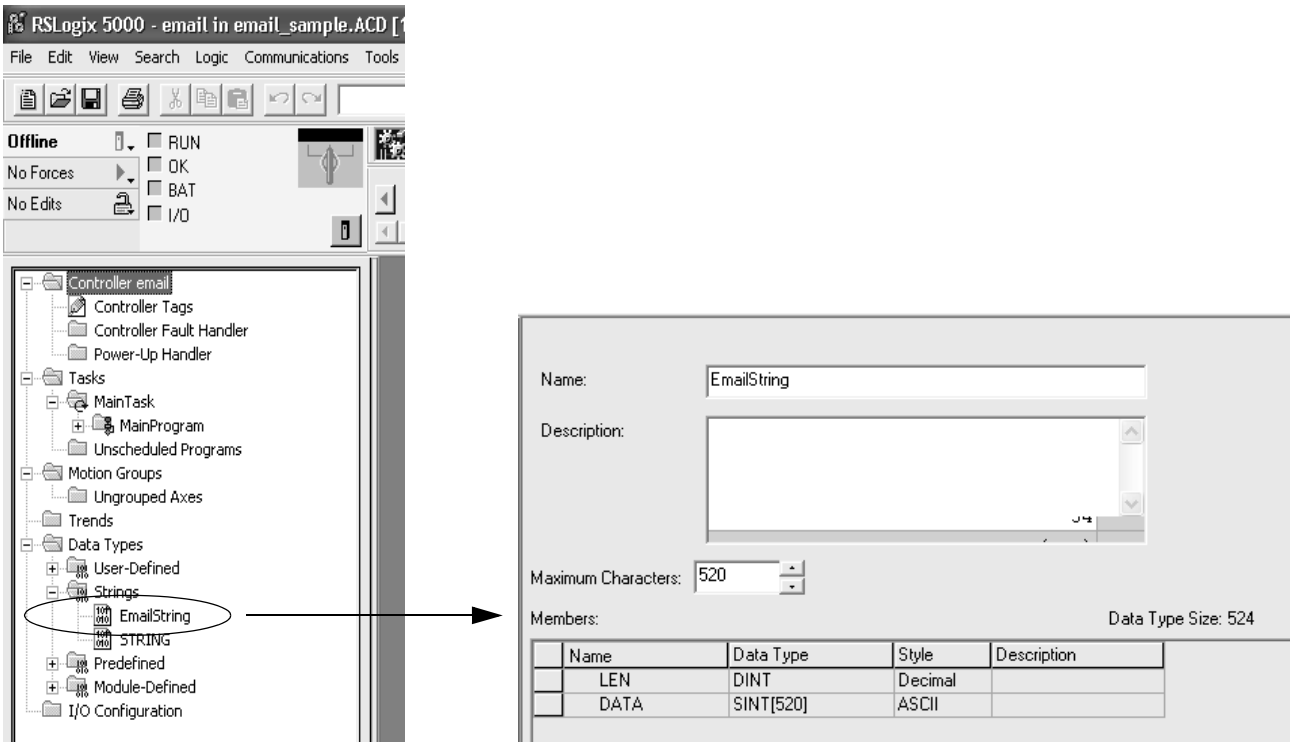
A Logix controller can send a generic CIP message instruction to the web server module that instructs the web server module to send an email message to a SMTP mail server using the standard SMTP protocol. This is useful to automatically communicate controller data and/or application conditions to appropriate personnel.

IMPORTANT Be careful to write the ladder logic to ensure the MSG instructions are not continuously triggered to send email messages.

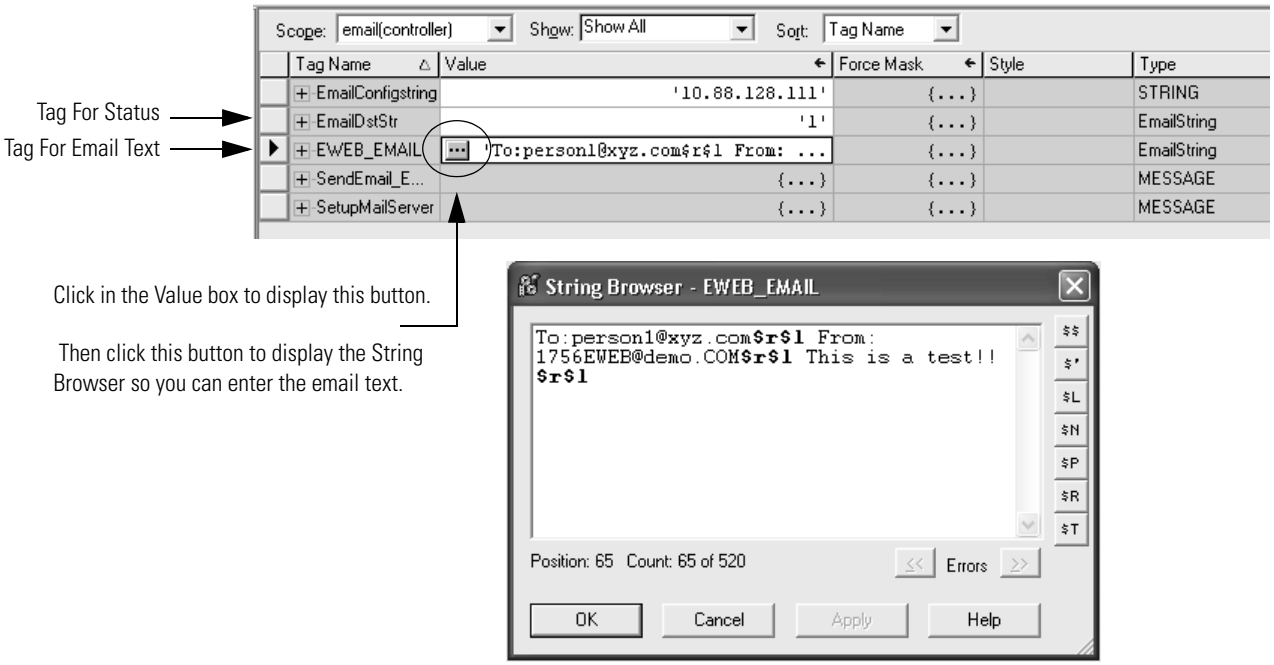
Create String Tags

You need two controller-scoped string tags, one to contain the email text and the other to contain the status of the email transmission. These tags can contain as many as 474 characters.

You must create a user-defined STRING data type (the default STRING data type in RSLogix 5000 software is not large enough for most email text). For example, create a STRING data type named EmailString.



Create one controller-scoped tag of this new data type to contain the email text. Create a controller-scoped second tag of this new data type to contain the transmission status. For example, create tag EWEB_EMAIL (to contain the email text) and EmailDstStr (to contain the transmission status). Both of these tags are of type EmailString.

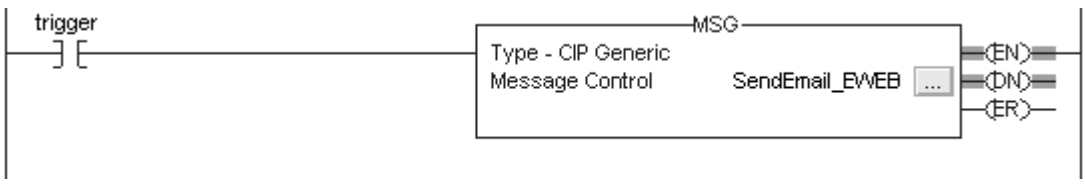


The text of the email does not have to be static. You can program a controller project to collect specific data to be sent in an email.

For more information on using ladder logic to manipulate string data, see the Logix5000 Controllers Common Procedures Programming Manual, publication 1756-PM001.

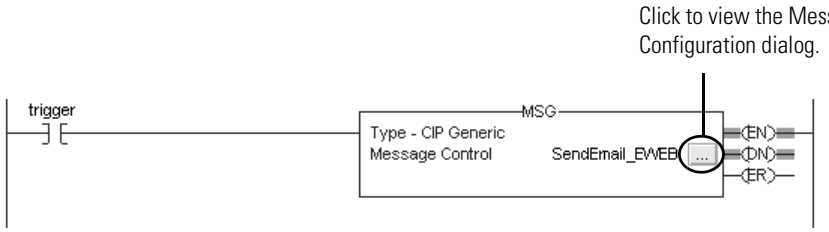
Enter the Ladder Logic

Add the MSG instruction that triggers the email. Execute this email MSG instruction as often as needed.



Configure the MSG Instruction

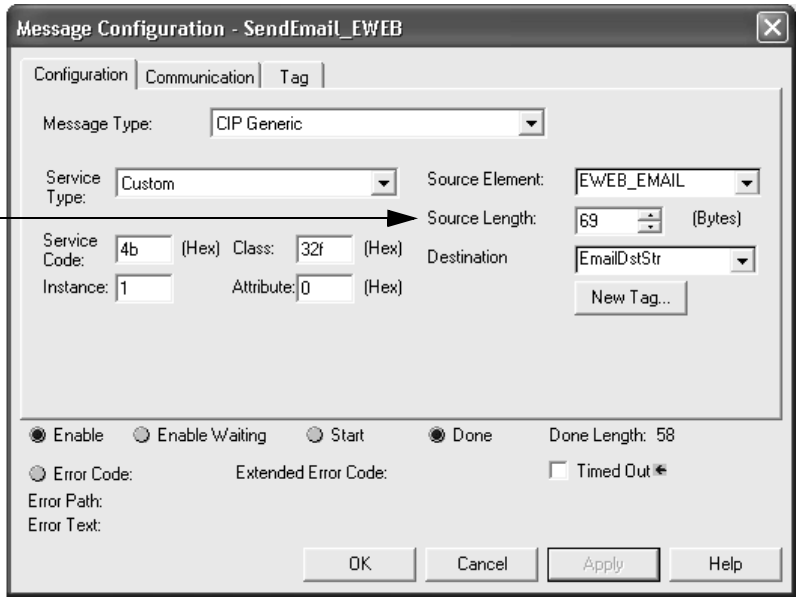
Use the following process configure the MSG instruction that contains the email text.



On the Configuration tab of the MSG instruction, configure the MSG parameters for sending an email.

The Source Length is the number of characters in the email tag plus 4 characters.

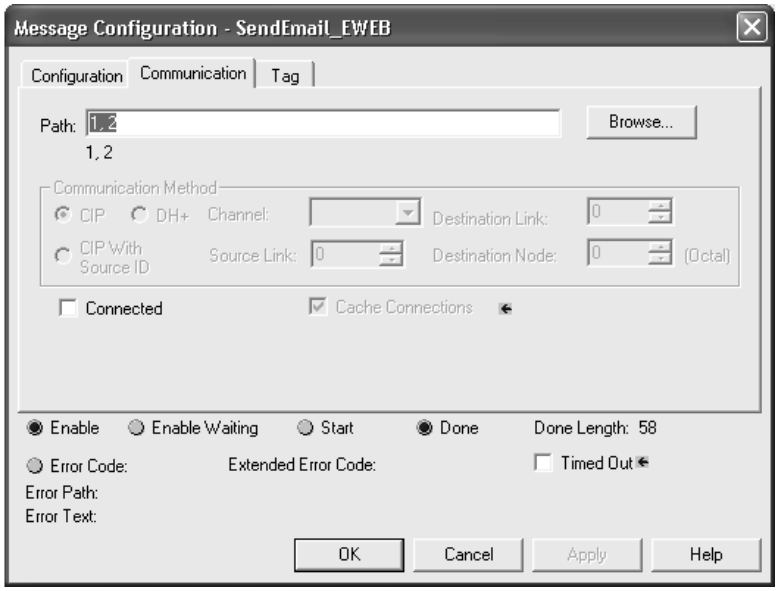
In this example, the email text contains 65 characters.



where:

In This Field	Enter
Service Type	Custom
Service Code	4b
Instance	1
Class	32f
Attribute	0
Source Element	The tag that contains the email text This tag is of the STRING data type you created to contain the email text. In this example, enter EWEB_EMAIL which is of type EmailString
Source Length	The number of characters in the email text plus 4 In this example, enter 69 (65 characters in the email + 4)
Destination	A tag to contain the status of the email transmission This tag is also of the STRING data type you created to contain the email text. In this example, enter EmailDstStr which is of type EmailString

On the Communication tab of the MSG instruction, configure the path from the controller to the web server module.



The path starts with the controller initiating the MSG instruction module. Then enter the port the message exits and the address of the next module in the path. For example, if the web server module is in the same chassis as the controller and is in slot 2, the path is: 1, 2.

If all the devices in the path are configured in the initiating controller's I/O Configuration tree, you can use the Browse button to select the

target web server module and the software automatically fills in the path.

For more information on configuring the path of a MSG instruction, see the Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003.

Enter the Text of the Email

Use the string browser to enter the text of the email. In the example above, you enter the email text into the EWEB_EMAIL tag. To include "To:", "From:", and "Subject:" fields in the email, use <CR><LF> symbols to separate each of these fields. The "To:" and "From"" fields are required; the "Subject:" field is optional. Use a second set of <CR><LF> symbols after the last one of these fields you enter. For example:

To: email address of recipient rl
From: email address of sender$r$l
Subject: subject of message rlrl
body of email message

The maximum length of an email message is 474 characters. An additional 4-byte string-length value is added to the tag. As a result, the maximum source length is 478 characters.

Possible Email Status Codes

Examine the destination element of the email MSG to see whether the email was successfully delivered to the mail relay server. This indicates that the mail relay server placed the email message in a queue for delivery. It does not mean the intended recipient successfully received the email message.

Here are possible codes that could be in this destination element.

Error Code (Hex)	Extended-error Code (Hex)	Description
0x00	None	Delivery successful to the mail relay server.
0x02	None	Resource unavailable. The email object was unable to obtain memory resources to initiate the SMTP session.
0x08	None	Unsupported Service Request. Make sure the service code is 0x4B and the Class is 0x32F.

Error Code (Hex)	Extended-error Code (Hex)	Description
0x11	None	Reply data too large. The Destination string must reserve space for the SMTP server reply message. The maximum reply can be 470 bytes.
0x13	None	Configuration data size too short. The Source Length is less than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x15	None	Configuration data size too large. The Source Length is greater than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x19	None	Data write failure. An error occurred when attempting to write the SMTP server address (attribute 4) to non-volatile memory.
0xFF	0x0100	Error returned by email server; check the Destination string for reason. The email message was not queued for delivery.
	0x0101	SMTP mail server not configured. Attribute 5 was not set with a SMTP server address.
	0x0102	"To:" address not specified. Attribute 1 was not set with a "To:" address AND there is not a "To:" field header in the email body.
	0x0103	"From:" address not specified. Attribute 2 was not set with a "From:" address AND there is not a "From:" field header in the email body.
	0x0104	Unable to connect to SMTP mail server set in Attribute 5. If the mail server address is a hostname, make sure that the device supports DNS, and that a Name Server is configured. If the hostname is not fully qualified, i.e., "mailhost" and not "mailhost.xx.yy.com" then the domain must be configured as "xx.yy.com". Try "ping <mail server address>" to insure the mail server is reachable from your network. Also try "telnet <mail server address> 25" which attempts to initiate a SMTP session with the mail server via telnet over port 25. (If you connect then enter "QUIT").
	0x0105	Communication error with SMTP mail server. An error occurred after the initial connection with the SMTP mail server. See the ASCII text following the error code for more details as to the type of error.
	0x0106	SMTP mail server host name DNS query did not complete. A previous send service request with a host name as the SMTP mail server address did not yet complete. Note that a timeout for a DNS lookup with an invalid host name can take up to 3 minutes. Long timeouts can also occur if a domain name or name server is not configured correctly.

Manage User Accounts and Access Levels

About This Chapter

This chapter describes how to configure levels of user access to different information on the module.

By assigning user accounts with different access levels, you can manage which users have access to change network configuration or have access to view and change data views.

Several pages on the web server module, such as module configuration pages and data views pages, have default access protection. Before accessing these pages, you must authenticate your access by entering a user name and password. The module displays the log-in box when you access these web pages.

IMPORTANT Once authenticated, you do not have to re-enter a user name or password when accessing subsequent pages. You **must** close your browser to log out.

The default user name is Administrator with no password.

IMPORTANT It is strongly recommended that you set a password for the default Administrator account.

User Accounts and Privilege Classes

The web server module supports multiple user accounts, each with a user name and password. Each user account is configured for one of these access levels:

- Administrator (all access)
- Write (read and write access)
- Read (read access only)

The access level determines which web pages each user can access. You configure access limits for individual web pages.

Configure Access Limits For Web Pages

You protect individual web pages and data views on a per URL basis. Each page in the web server module has one of these protection levels:

- Administrator
- Write
- Read

The protection levels are hierarchical. Administrator users can access Write and Read protected pages, and Write users can access Read protected pages.

These predefined pages (those web pages that come with the web server module) in the web server module have these default access levels. You can change these access levels, if needed.

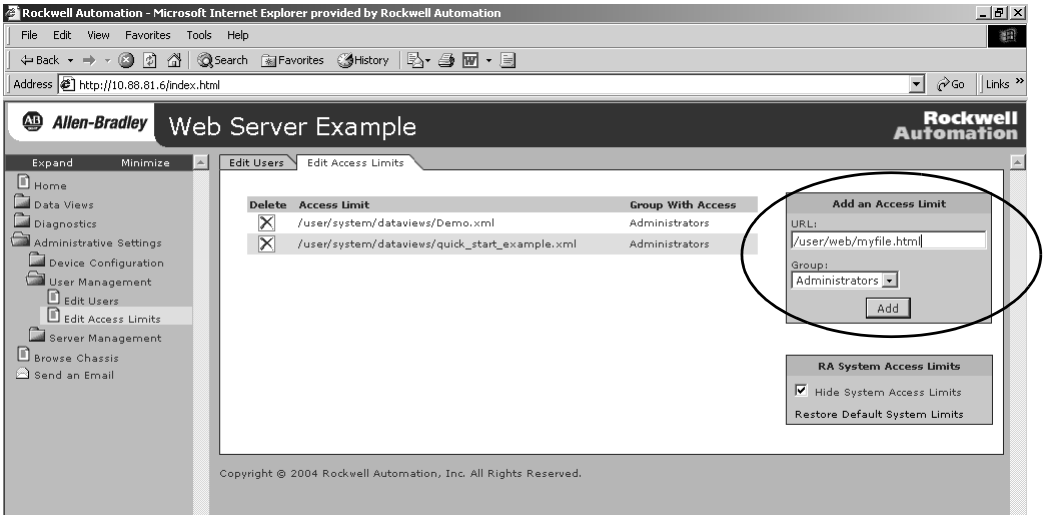
Web Page	Required Protection Level
Home page	No protection
Diagnostics pages	
Chassis Browse page	
Data Views with read-only tags	Read protection
Data Views with write tags	Write protection
Data Views with administrator tags	Administrator protection
Device configuration pages	
Server Management page	
User Management page	
Send Email page	

In Data Views, the access limits you specify for a tag applies to the whole data view, not just the tag. If you have multiple tags with different access levels in the same data view, the web server assigns the highest (most access) level to the data view. For more information, see chapter 3.

If you develop custom web pages, you must explicitly specify the access limits for the page if you want access protection. Otherwise, the custom web page will have no access limits.

You need Administrator access to modify access limits for web pages. You specify the access limit for a web page by selecting Administrative Settings > User Management > Edit Access Limits.

You can change the default access limits for the predefined web pages or you can add pages to the protection list, such as custom web pages. The Edit Access Limits page shows the current list of pages the user has selected for protection. The predefined pages, though they have default protection, do not show up in the list.

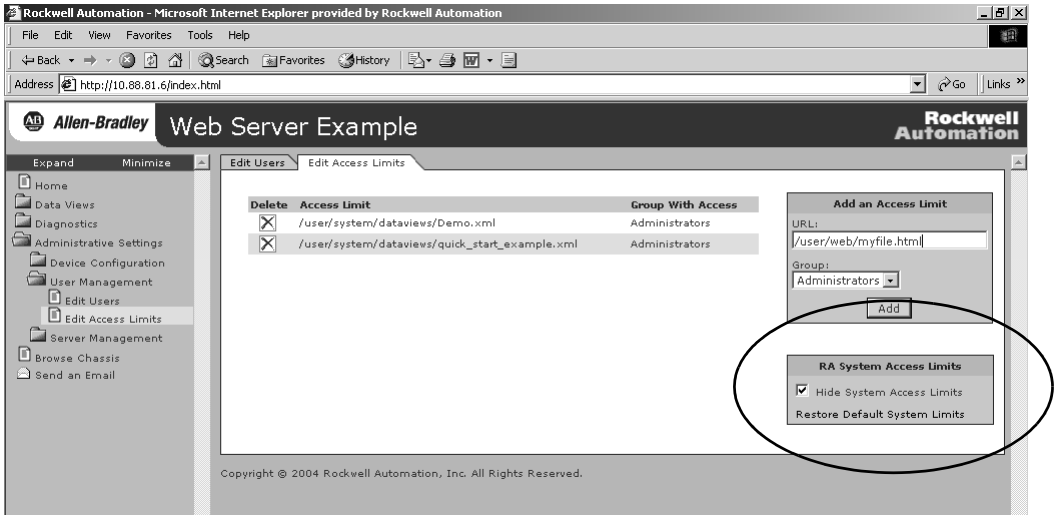


In This Field	Do This
URL	Enter the URL for the web page (80 characters maximum, including slashes). Enter only the relative path of where the page is stored on the web server module, such as '/user/Web/mypage.html'
Group	Select Administrator, Write, or Read access limit for the specified URL

You can configure protection limits for predefined pages, as well as for user-supplied pages. You can also apply protection to directories so that all the files in a specified directory have the same access limit.

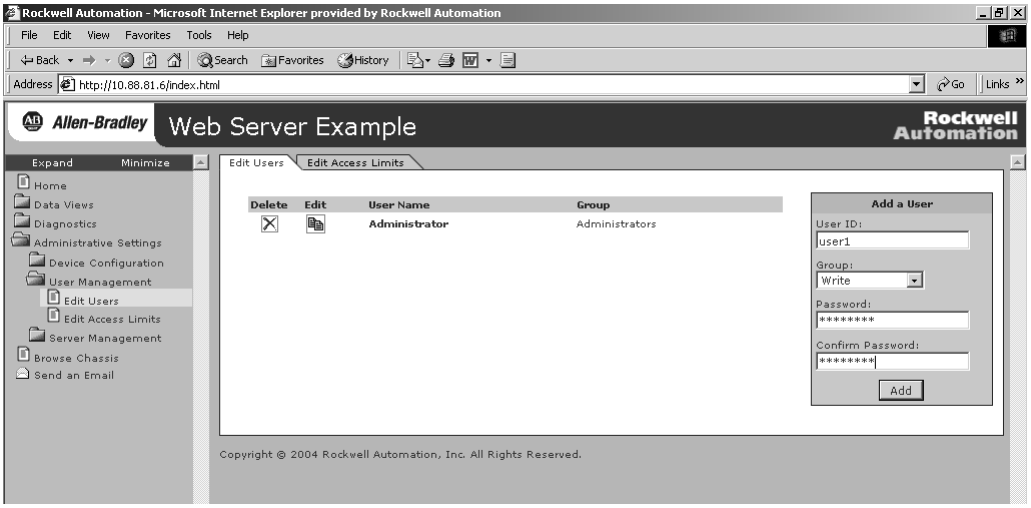
To specify a directory, enter the URL with a front slash (/) on the end. If you do not enter the front slash, the protection limit you select is assigned to the web page. To specify an access limit for the entire web site, enter only a front slash (/).

To see the protection limits for predefined web pages and directories, unclick the Hide System Access Limits selection. This also provides access to change the predefined protection limits. You can then use the Restore System Access Limits selection to return to the default protection limits.



Create User Accounts

You need Administrator access to create and modify user accounts. You can create as many as 25 individual accounts. You manage accounts by selecting Administrative Settings > User Management > Edit Users.



Field	Action
User ID	Enter the user name for the account (80 characters maximum). Can contain these characters: A-Z, a-z, 0-9, underscore (_), and dash (-).
Group	Select Administrator, Write, or Read access for the user account.
Password	Enter the password for the account (80 characters maximum).
Confirm Password	Re-enter the same password for the account.

IMPORTANT If you use Internet Explorer, the number of characters allowed for a user ID or password depends on how many characters "fit in the box." Larger characters (such as "W") take more room and reduce the total number of allowed characters.

This limitation does not apply if you use Netscape, Mozilla, or some other browser.

Recover with Unknown Password

There are no back-door accounts or passwords in the event that you forget the web server module's passwords or inadvertently delete all the Administrator accounts.

To recover a web server module with unknown passwords, you must use ControlFlash to restore the web server's flash file system to the factory default. This operation deletes all user accounts, data views, and user-loaded web pages. Contact technical support for the appropriate recovery script and binary file.

Access Files in the Web Server Module

About This Chapter

This chapter describes how to use FTP to access the file system on the web server module.

You use FTP access to store custom web pages and applications on the web server module.

Access the Web Server's File System

The web server module has a flash file system that stores web pages and data views. The following table contains the amount of space available in each EWEB module's flash file system.

Catalog No.	Mbytes Available
1756-EWEB	5
1768-EWEB	2

Some predefined directories exist to store specific types of data.

Use This Directory	For
/user	This is the highest directory level you can access on the web server module. It contains two subdirectories: • /Web for you to store your custom-created pages or other files • /system to store configuration files and data views You can only access this directory (via FTP) when the web server module is in backup/restore mode.
/user/Web	Use this directory to store your custom-created pages or own files. Standard FTP provides access to this directory; the module does not need to be in any special mode.
/user/system	This predefined directory contains two predefined directories: • /configuration to store network and module configuration files • /dataviews to store data view files You can only access this directory (via FTP) when the web server module is in backup/restore mode.
/user/system/configuration	This predefined directory contains network and module configuration files in an XML format. You can only access this directory (via FTP) when the web server module is in backup/restore mode.

Use This Directory	For
/user/system/dataviews	This predefined directory contains data view files in an XML format. You can only access this directory (via FTP) when the web server module is in backup/restore mode.
/schema	This directory contains dataview.xsd and CIPDataTypes.xsd schema files for validating data views. You can only access these .xsd files with a web browser: • http://ip_address/schema/dataview.xsd • http://ip_address/schema/CIPDataTypes.xsd
/dataview	This predefined directory contains dataview.xsl which is the external XSL stylesheet for data views. You can only access .xsl file http://ip_address/dataview/dataview.xsl with a web browser.

There are no restrictions on the type of files you can copy to the web server module. You are restricted only by the amount of memory available.

You access this file system using any standard FTP client. By default, FTP is disabled for the web server module. You enable FTP by selecting Administrative Settings > Device Configuration > Device Services.

Connect to the Web Server Module

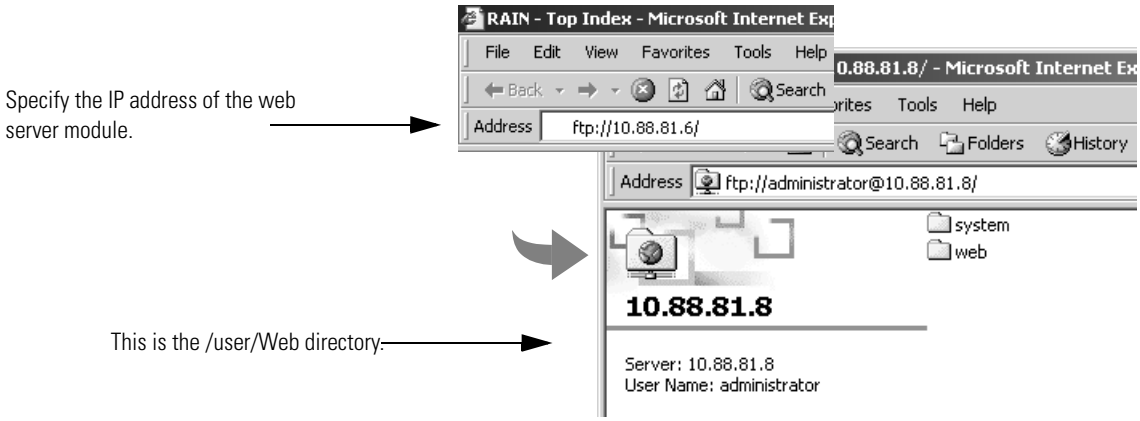
IMPORTANT FTP access must be enabled for the web server module before you can use FTP to access the module. You enable FTP on the Administrative Settings > Device Configuration > Device Services page.

To establish a FTP connection to the web server module, you need Administrator access to the module. You will also need to enter your user name and password.

IMPORTANT The password is transmitted over the network in plain text. We recommend that you create a temporary Administrator account for use when you use FTP to access to the web server module.

Leave FTP disabled except for when actively copying files to or from the web server module.

To connect to the web server module via FTP, specify the IP address of the module.



Once connected to the web server module, you have access to the /user/Web directory. You can create subdirectories, but you cannot access any directories higher than this user directory (which contain predefined web pages and files).

The web server module supports four FTP sessions (four different users with simultaneous FTP access to the module). If you use Internet Explorer to FTP to the web server module, there are only two FTP sessions available.

When the web server module is in backup/restore mode, an FTP session points to a different directory than when the web server module is not in backup/restore mode. If you have a standard FTP session open to the web server module, close this before placing the module in backup/restore mode. Once in backup/restore mode, the previously open FTP session will no longer point to the correct directory and files.

File Names and Types

File names can have no more than 80 characters. In addition, the complete path for any file in the file system can have no more than 80 characters. For example, the user-created web pages go in the /user/Web folder, which is actually in the file system as "/root/user/Web" (already 14 characters). Any path underneath this must have 65 or less characters, including slashes, dots, and extensions.

File names are case-insensitive and can contain any characters, including spaces, except for the following characters: ? " / \ < > * | :

When user files are accessed via HTTP, certain file extensions result in specific values returned in the Content-Type field.

Here are some Content-Types for the more commonly-used files.

File Extension	Content-Type
.htm	text/html
.html	text/html
.asp	text/html
.gif	image/gif
.jpg	image/jpeg
.css	text/cascading style sheet
.txt	text/plain
.js	application/x-javascript
.exe	application/binary
.z	application/compressed
.gz	application/gzip
.bin	application/octet-stream
.oda	application/oda
.pdf	application/pdf
.ai	application/postscript
.eps	application/postscript
.ps	application/postscript
.xml	text/xml
.xsl	text/xml

In addition, files with an .asp extension are processed by the web server as Active Server Pages.

For more information on creating custom web pages, see chapter 8.

Back Up the File System On the Web Server Module

There are several items stored on the web server module that you might want to archive in a backup copy:

- User accounts and passwords.
- List of URLs selected for read, write, or administrator protection.
- Data views.
- Custom web pages.
- Module configuration data.

To back up these files, you use standard FTP to access the web server module and then copy or restore files. Using FTP lets you copy files to or from the web server module without interrupting operation of the module. FTP also lets your standard FTP-capable clients, such as Internet Explorer or WinZip to copy the system files.

FTP access during a backup/restore procedure differs from normal FTP access in that you get access to the /user directory, which contains the /Web directory that is accessible during normal FTP access. This lets you back up predefined web pages and data views, as well as any custom files you have copied to the web server module.

The Administrative Settings > Server Management > Backup/Restore lets you lock access to the module while you back up or restore files.

Locking the site prevents other users from HTTP access to the web server module. Locking the site does not affect bridge functionality through the web server module, however, restoring files to the web server module that modify network or module configuration settings (such as IP address) can affect bridge functionality.

Back Up Files

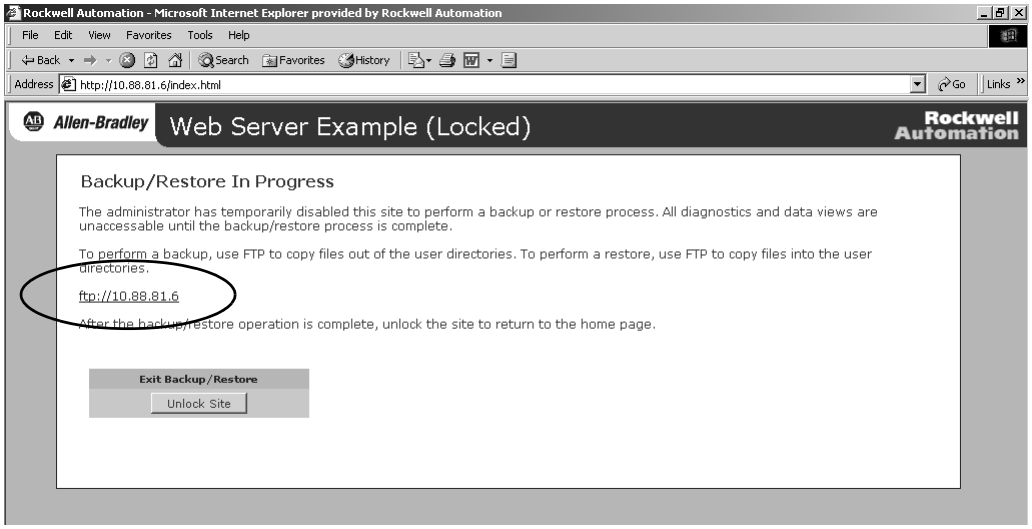
You must have Administrator access to back up files. To back up files that reside on the web server module:

1. Select Administrative Settings > Server Management > Backup/Restore and click Lock Site to lock access to the web server module.

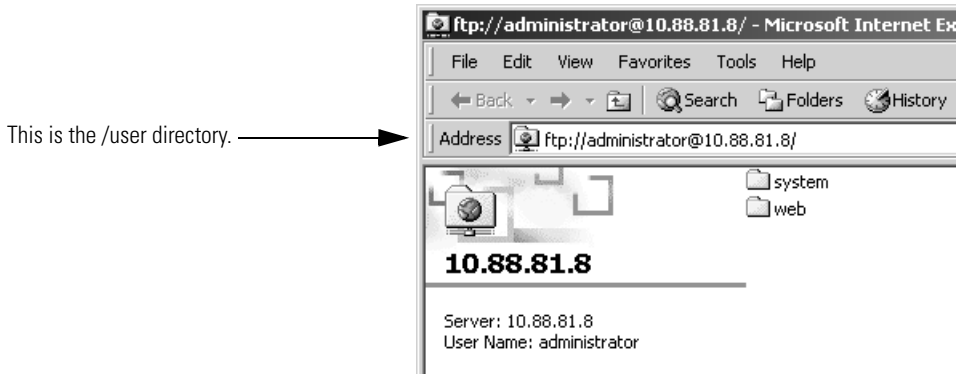
   You must lock access to the web server module to gain FTP access to the /user directory. The web server module asks to confirm that you want to lock the site.

   This automatically enables FTP access to the web server module.

2. Click the link to the web server module. This appears on the locked backup/restore page.



The module will require that you enter a valid user name and password that allows administrator access. Once authenticated, you have access to the /user directory. This is one directory higher than the /user/Web directory available during normal FTP operations.



3. Select the files you want to back up.

4. Copy the selected files to a directory on your computer.

5. Unlock access to the module by selecting Administrative Settings > User Management > Backup/Restore.

Restore Files

You must have Administrator access to restore files. To restore files from your computer to a web server module:

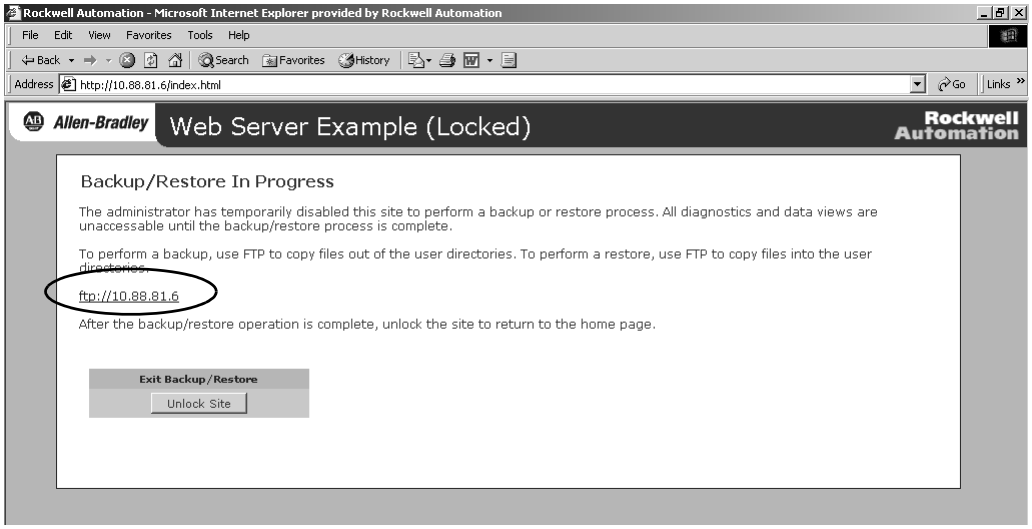
1. Select Administrative Settings > Server Management > Backup/Restore and click Lock Site to lock access to the web server module.

   You must lock access to the web server module to gain FTP access to the /user directory. The web server module asks to confirm that you want to lock the site.

   This automatically enables FTP access to the web server module.

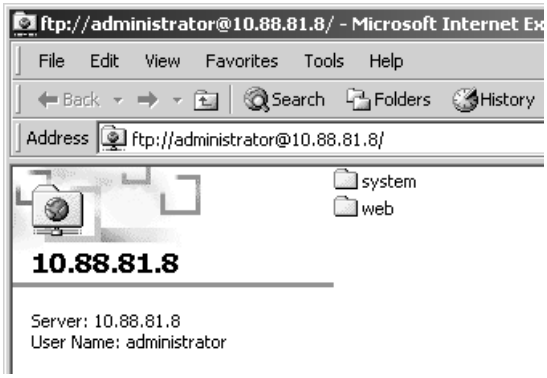
2. Click the link to the web server module.

This appears on the locked backup/restore page.



The module will require that you enter a valid user name and password that allows administrator access. Once authenticated, you have access to the /user directory. This is one directory higher than the /user/Web directory available during normal FTP operations.

This is the /user directory. →



3. Select the files from your computer that you want to restore to the web server module.

4. Copy the selected files to a directory on the web server module.

5. Unlock access to the module by selecting Administrative Settings > User Management > Backup/Restore.

Create Custom Web Pages

About This Chapter

This chapter describes how to use ASP functions in custom web pages and how to load custom web pages into the web server module.

IMPORTANT The web server module provides access to tags within a local controller via a web browser. However, the web server module is not recommended for use as a real-time HMI or operator interface.

The predefined web pages that come with the module provide one method of accessing these tags. The ability to load and run custom web pages gives you the flexibility to design web pages that better fit your application. For example, you can define a web page with standard web content, such as a company logo, contact information, and links to other web pages. Add the ASP functions to display live controller data.

Overview

Use your own editor or application to develop the appropriate HTML and ASP files for your custom web pages. Once these files are ready, you copy the files to the web server module and configure the pages as needed.

The following steps outline the process of developing custom web pages and getting them ready to use.

1. Develop the appropriate HTML and ASP files.

   See Develop a Custom Web Page on page 93 for information about the available ASP functions and other features for displaying and changing controller data.

2. Use FTP to copy the custom pages to the web server module.

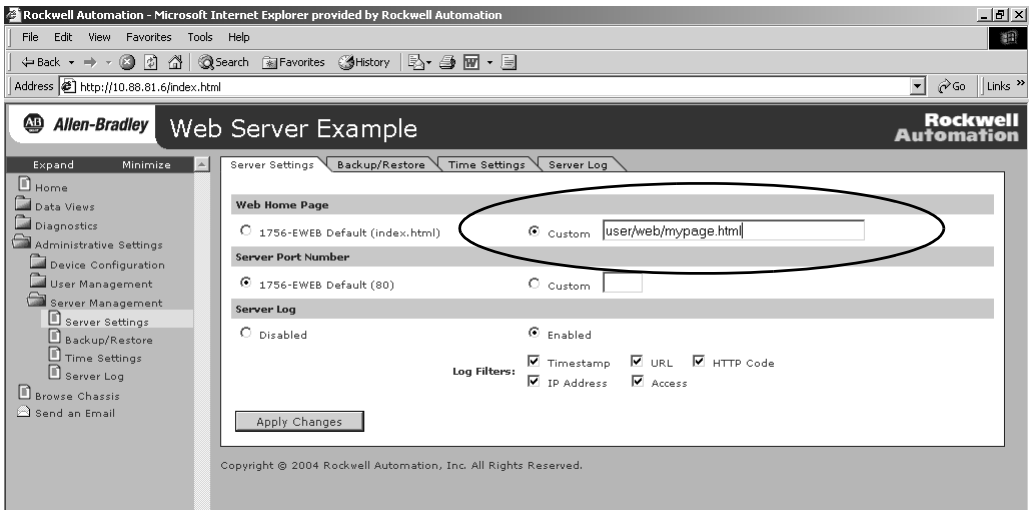
3. Copy your files into the /user/Web directory.

   See chapter 7 for more information.

4. Decide whether to make your custom page the default Home page for the web server module.

TIP

Select Administrative Settings > Server Management Server > Server Settings. When you click the Custom button, the web server module automatically fills in the /user/Web portion of the location of the custom file. Enter in any further directories and the file name. Click Apply Changes.



5. Determine whether you want to set access limits, such as Administrator, Write, or Read, on the custom page.

   Select Administrative Settings > User Management > Edit Access Limits page to set access limits on specific pages.

   See chapter 6 for more information.

Access Custom Web Pages

Once a custom web page is copied into the web server module, you can display that page from your web browser:

If You Custom Web Page Is	In Your Web Browser
Configured as the default home	Enter the IP address of the web server module. The web browser accesses the defined web page for the web server module. If you specify a custom page as the default home page, you can still access the predefined home page by entering: http://ip_address/index.html
Not configured as the default home page, but is copied into the /user/Web directory	Enter: http://ip_address/user/Web/my_file.html The web browser displays any custom file if you specify the complete URL to that file.

Develop a Custom Web Page

Custom web pages can contain standard web content, such as a company logo, contact information, and links to other web pages. In addition, you can use ASP functions and other features that come with the web server module to display and manipulate live controller data.

An ASP file is essentially the same as an HTML file, with embedded scripting constructs supported by ASP. An ASP file has the file extension .asp. When an ASP file is requested by a browser, the web server interprets the ASP file and executes the ASP-specific scripts. The browser sees only the resulting HTML, not the embedded ASP scripts.

Use	To	See Page
ASP function calls	Display controller data	93
Javascript libraries	Manipulate the controller data returned by ASP function calls	98
Web page forms and POST handlers	Change controller data	102

ASP Function Calls

The web server module provides these ASP functions.

To	Use This ASP Function	See Page
Read controller tags	ReadLogixTag	94
	ReadLogixTagUnconnected	94
Read CIP data	CIPMessage	95
	CIPMessageUnconnected	96
Retrieve information about the module	GetSetting	97

Read Controller Tags

There are two ASP functions you can use to read controller tags:

If You Want	Use This ASP Function	See Page
Connected messaging	`ReadLogixTag`	94
Unconnected messaging	`ReadLogixTagUnconnected`	94

Function: `ReadLogixTag(`*`path, tagname, tagtype`*`)`

The `ReadLogixTag` function connects to a controller at a specified *`path`*, retrieves the value of a tag with a name `tagname` and of `tagtype`.

Parameters:

Parameter	Description
`path`	The path is a CIP path, with no spaces, and segments separated by commas. For example: **Path Example**: 1,3 — **Description**: 1 = backplane, 3 = slot 3
`tagname`	The tagname identifies a controller-scoped tag.
`tagtype`	The tagtype must be an atomic type (BOOL, SINT, INT, DINT, REAL) or string. Use standard dot notation to specify a member of an array or a user-defined structure (for example, 'timer1.ACC' for the accumulator of a timer tag named 'timer1').

Example: For example, retrieve a DINT tag named 'my_dint_tag' from a controller in slot 3 of the local chassis:

```
<p> "my_dint_tag" value:
<% ReadLogixTag("1,3", "my_dint_tag", "DINT"); %>
</p>
```

Function: `ReadLogixTagUnconnected(`*`path, tagname, tagtype`*`)`

The `ReadLogixTagUnconnected` function performs the same task as `ReadLogixTag` function and uses the same parameters. This function, however, retrieves the tag value through unconnected messaging, rather than connected messaging.

Read CIP Data

There are two ASP functions you can use to read CIP data.

If You Want	Use This ASP Function	See Page
Connected messaging	`CIPMessage`	96
Unconnected messaging	`CIPMessageUnconnected`	96

Function: `CIPMessage(`*`path`*`, `*`service`*`, `*`class`*`, `*`instance`*`, `*`attribute`*`, `*`member`*`, `*`data`*`, `*`returntype`*`)`

The `CIPMessage` function performs the CIP service specified by the `service` parameter on an object or object instance specified by the `class`, `instance`, `attribute`, and `member` parameters, using the data passed in the `data` parameter (if necessary) and returning the value with a type specified by the `returntype` parameter (if appropriate).

Parameters:

Parameter	Description
`path`	The path is a CIP path, with no spaces, and segments separated by commas. This is the same as the path parameter in the read-controller-tag functions.
`service`	The service identifies the CIP function to perform.
`class`	The class, instance, attribute, and member parameters identify the object for the service. If a service does not use an instance, attribute, and/or member, pass the parameters as 0.
`instance`	
`attribute`	
`member`	
`data`	The data to be passed. If no data is passed in the request, set the data parameter to an empty string.
`returntype`	The returntype parameter is the same as the tagtype parameter used in the read-controller-tag functions, except when a returntype of "STRING" is specified, the data returned from the service is written to the browser as a space-delimited string of hex digits.

Example: For example, a GET_ATTRIBUTE_ALL to the identity object of a device in slot 1 of the local chassis:

```
<% CIPMessage("1,1", 1, 1, 1, 0, 0,"","STRING"); %>
```

This ASP call returns a string similar to:

```
01 00 0E 00 03 00 0C 0C 70 30 63 2E 08 00 1D 31 37 35
36 2D 4C 31 2F 41 20 31 37 35 36 2D 4D 30 2F 30 20 41
52 47 5F 31 32 5F 33 38
```

While the value returned by specifying the “STRING” data type may not be very useful to display in the browser by itself, some built-in Javascript libraries help the ASP developer parse and use the information contained in these string structures. See page 98 for more information on the Javascript libraries.

Function: `CIPMessageUnconnected(`*`path`*`, `*`service`*`, `*`class`*`, `*`instance`*`, `*`attribute`*`, `*`member`*`, `*`data`*`, `*`returntype`*`)`

The `CIPMessageUnconnected` function performs the same task as the `CIPMessage function` and expects exactly the same parameters. This function, however, performs the specified service through unconnected messaging, rather than connected messaging.

Update Control System Data

You can use the `CIPMessage` and `CIPMessageUnconnected` functions to perform any CIP service, such as a SET_ATTRIBUTE_SINGLE, RESET, or any other type of write or update service to an object or object instance. However, since ASP pages are parsed and executed server-side, it is impossible to send any dynamic data to these services.

As an example, consider a page which performs a reset to the identity object of a device in slot 1, then immediately redirects the browser to the main user page.

```
<HTML>
<HEAD>
<% CIPMessageUnconnected("1,1", 5, 1, 1, 0, 0, "",
"STRING"); %>
<META HTTP-EQUIV="refresh"
content="0;URL=/user/index.html">
</HEAD>
<BODY></BODY>
</HTML>
```

Retrieve Information About the Web Server Module

You can retrieve specific information about the web server module.

Function: `GetSetting (settingname)`

The `GetSetting` function retrieves a specific piece of information about the web server module itself and writes this data to the browser.

Parameters:

Parameter	Description
`settingname`	The settingname specifies the piece of information to retrieve.

where `settingname` can be:

Value	Description
"uptime""device_status" "firmware_revision""firmware_version" "serial_number"	These fields as displayed on the predefined home page. See page 22 for more information.
"name""description" "location""contact1" "contact2"	These fields as configured on the Administrative Settings → Device Configuration →Device Identity page. See page 48 for more information.
"ip_address""subnet_mask" "default_gateway""primary_name_server" "secondary_name_server""domain_name" "hostname""obtain_configuration" "dns_enable""smtp_server" "ethernet_address""autonegotiation" "port_speed""duplex_mode"	These fields as configured on the Administrative Settings → Device Configuration →Network Configuration page. See page 49 for more information.
"cpu_utilization""file_sys_utilization" "server_errors""server_redirects" "server_timeouts""server_access_violations" "server_page_hits""server_form_hits" "server_total_hits""user_free_space" "server_data_views""server_total_tags" "tcp_conns""tcp_conn_limit" "tcp_max_conns""cip_conns" "cip_conn_limit""cip_max_conns" "cip_conn_opens""cip_conn_open_errors" "cip_conn_closes""cip_conn_close_errors" "cip_conn_timeouts""cip_msg_sent" "cip_msg_rcv""cip_ucmm_sent" "cip_ucmm_rcv"	The diagnostic information on the Diagnostics →Diagnostic Overview page. See page 114 for more information.

Value	Description
"if_in_octets""if_in_ucast" "if_in_nucast""if_in_discards" "if_in_errors""if_in_unknown_protos" "if_out_octets""if_out_ucast" "if_out_nucast""if_out_discards" "if_out_errors""media_alignment_errors" "media_fcs_errors""media_single_collisions" "media_multiple_collisions""media_sqe_test_errors" "media_deferred_trans""media_late_collisions" "media_excessive_collisions""media_mac_trans_errors" "media_carrier_sense_errors""media_frame_too_long" "media_mac_receive_errors"	The diagnostic information on the Diagnostics → Ethernet Statistics page. See page 119 for more information.
"time" "asc_local_time"	Time in seconds since January 1, 1970 ASCII string of time and date These values display the time the web page is accessed.

Javascript Libraries

The web server module provides built-in Javascript functions designed to help manage control system data in custom web pages.

Use This Javascript Library	To	See Page
conversion.js	Convert values in the string returned from `CIPMessage` and `CIPMessageUnconnected` functions using the STRING data type into atomic, numeric or string values.	99
XMLObjectLoaderLib.js	Convert the string returned from a GET_ATTRIBUTE_ALL service into a Javascript object which lets you refer to the object or instance attributes by name.	100

Javascript Library: Conversion.js

The conversion.js library contains six functions. To include this library in your custom web page, include this line.

```
<script type="text/javascript"
src="/scripts/conversion.js">
</script>
```

TThe conversion.js library contains these functions.

Function	Description

`parseStruct(structStr, type, stringlength)`	The `parseStruct` function is the main function, where:
	structStr — Contains the string returned from the ASP call, or a substring thereof, beginning at the hex number which starts the value to be retrieved from the string.
	type — Indicates the data type to be parsed from the string. It should be one of "SINT", "USINT", "INT", "UINT", "WORD", "DINT", "UDINT", "DWORD", or "STRING".
	stringlength — (Optional) only used when the type to be parsed is a "STRING". In this case, it identifies how many digits of the input string to parse into ASCII characters.
	For example, to retrieve the device name from a device in slot 1 of the local chassis:
	`var idobj = "<% CIPMessage("1,1", 1, 1, 1, 0, 0,"0","STRING"); %>";`
	`var namelen = parseStruct(idobj.substring(42, idobj.length), "USINT");`
	`var name = parseStruct(idobj.substring(45, idobj.length), "STRING", namelen);`
	The Device Name attribute of the identity object instance starts 14 bytes into the GET_ATTRIBUTE_ALL response. Each byte of the response in the "STRING" return style takes 3 bytes (2 hex digits and a space), so the Device Name attribute begins at the 42nd character of the string. The first byte of the string contains the number of characters in the string. After parsing this string length, pass the length on to the next function, which parses the actual string beginning at the 15th byte (45th character) of the string.

`decToHex(decimalnumber)`	The `decToHex` function takes an unsigned decimal number as a parameter and returns a string representing *decimalnumber* in hexadecimal notation. The return value does not have a preceding "0x" and the returned string is always 8 characters long, with leading zeros when necessary.
`hexToDec(hexnumber)`	The `hexToDec` function takes a string containing a hexadecimal number with no preceding "0x" as a parameter, and returns a decimal number with the value of *hexnumber*.
`decToOct(dintnumber)`	The `decToOct` function takes an unsigned decimal number as a parameter and returns an 11-character string representing *dintnumber* in octal notation. The return value does not have a preceding "0" and the returned string is always 11 characters long, with leading zeros when necessary.

Function	Description
`decToBin(decimalnumber)`	The `decToBin` function takes an unsigned decimal number as a parameter and returns a string representing *decimalnumber* in binary notation. The return value does not have a prefix and breaks the resulting binary string into groups of four characters.

Javascript Library: XMLObjectLoaderLib.js

This library uses the conversion.js library. To include these libraries in your custom web page, include these lines.

```
<script type="text/javascript"
src="/scripts/conversion.js">
</script>
<script type="text/javascript"
src="/scripts/XMLObjectLoaderLib.js">
</script>
```

The XMLObjectLoaderLib library requires an input file in XML format which describes the CIP object being returned from a GET_ATTRIBUTE_ALL service. The following example shows the format of this input file:

```
<object name="IdentityInstance">
      <attribute index="1" name="VendorID" type="UINT"/>
      <attribute index="2" name="DeviceType" type="UINT"/>
      <attribute index="3" name="ProductCode" type="UINT"/>
      <object index="4" name="Revision">
                                    <attribute index="1" name="MajorRevision"
type="USINT"/>
                                    <attribute index="2" name="MinorRevision"
type="USINT"/>
      </object>
      <attribute index="5" name="Status" type="WORD"/>
      <attribute index="6" name="SerialNumber" type="DWORD"/>
      <attribute index="7" name="ProductName" type="STRING"/>
</object>
```

This description format is recursive - structures within the object can be described by declaring an "`object`" element, then describing the members of the structure under consideration. The names of the elements are used to construct the Javascript object, and the members of the Javascript object have names that match the "name" attributes in the description file. The "`index`" attributes describe the order in which these elements occur in the object; each attribute and object except the root object must contain an index attribute.

These functions are contained in the XMLObjectLoaderLib.js library.

Function	Description
`XMLObjectLoader()`	This routine initializes the `XMLObjectLoader` object used to parse the CIP return data. Set it to a declared variable in the Javascript. For example: `var xol = new XMLObjectLoader();`
`LoadObject(objectname, xmlfilename, objectstruct)`	The `LoadObject` function is a member function of the XMLObjectLoader object that parses the CIP object return string passed in objectstruct, using the XML description file xmlfilename, and placing the result in the objectname variable, where: objectnameglobal variable that contains the result, accessible to all scripts within the page. xmlfilenameabsolute or relative path to the XML file containing the object description objectstruct CIP object return string

For example, the following code is a snippet from a web page which displays information about the identity object of a device in slot 1 of the local chassis.

```
<script type="text/javascript" src="/scripts/conversion.js"></script>
<script type="text/javascript" src="/scripts/XMLObjectLoaderLib.js"></script>
<script type="text/javascript" language="javascript"> var idobj;

function loadPage() {
      document.getElementById("vendor").innerText = "" + idobj.VendorID;
      document.getElementById("type").innerText = "" + idobj.DeviceType;
      document.getElementById("code").innerText = "" + idobj.ProductCode;
      document.getElementById("rev").innerText = idobj.Revision.MajorRevision + "."
+
        idobj.Revision.MinorRevision;

      document.getElementById("status").innerText = "0x" + decToHex(idobj.Status);
      document.getElementById("serial").innerText = "0x" +
decToHex(idobj.SerialNumber);
      document.getElementById("prodname").innerText = idobj.ProductName;
}

function loadme() {
   var id = "<% CIPMessage("1,1", 1, 1, 1, 0, 0,"0","STRING"); %>"
   var xl = new XMLObjectLoader();
   xl.LoadObject("idobj", "/CIPXMLobj/CN_01_Identity.xml", id);
   loadPage();
</script>

<body onload="loadme();" ...>
```

Web Page Forms and POST Handlers

Any web page that displays controller data can include a way to change that data by creating a form on the web page.

Include this statement in your web page file.

ACTION="change_method" METHOD="POST"

where change_method is one of these.

Use This Method	To	See Page
/user/system/dataviews/filename.xml	Change all data in a data view or only specific data in a data view	102
/rokform/WriteLogixTags	Change any data in a controller, whether it is in a data view or not	108
/rokform/ReadLogixTags	Enter a tag name and receive its current value	110
/rokform/CIPMessage	Send an instant CIP message request	111

ACTION="/user/system/dataviews/filename.xml"

This method follows the same user access restrictions as the restrictions for the user requesting the data view. You can only post the tags that appear on the data view. You must have the same access level as the level configured for the data view.

You can post the same XML that you got in the requested file with the desired data values changed or only the controller tags you want to change.

To recreate the following example:

1. Create a Logix controller tag named TEST of data type DINT.
2. In the web server module, create a data view named Sample.

   Include tag TEST of data type DINT in this data view. All names are case sensistive.
3. Verify that the data view Sample works.
4. Copy the example code (from a PDF version of the user manual) and paste it into a text editor.

Remove the newlines from the alert text or shorten the alert text to one line.

IMPORTANT In alert statements, the text of the alert must be one complete line (that can wrap within your text editor) or you must enter line continuation characters to separate lines. All data view and tag names are case sensitive.

If you copy the following example, you must either shorten the example alert text or remove the newlines (carriage returns or linefeeds) that appear when you paste the code into your text editor. For example, either of these alert statements follows correct syntax.

```
alert("You need Internet Explorer 5.5 or greater with XML support.");
```

or

```
alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with\
XML support. Please upgrade your browser or load the appropriate patches to support
XML.");
```

5. Save the edited code as an html file, such as xmlSample.html.
6. FTP the file Transfer xmlSample.html to the web server module.
7. Test http://ip_address/user/Web/xmlSample.html.

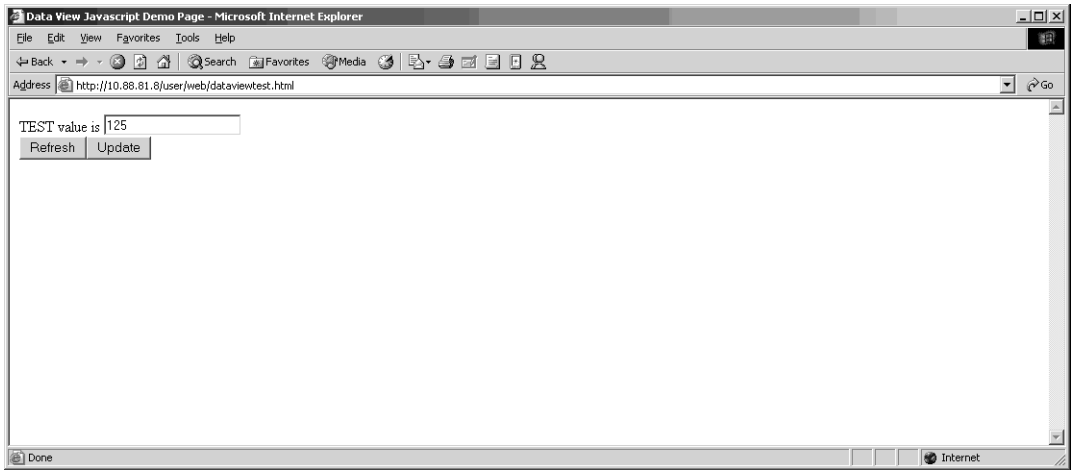
You can download application example files from the RA Knowledgebase. See Technote A142365164 - Example files for 1756-EWEB module.

For example, this code:

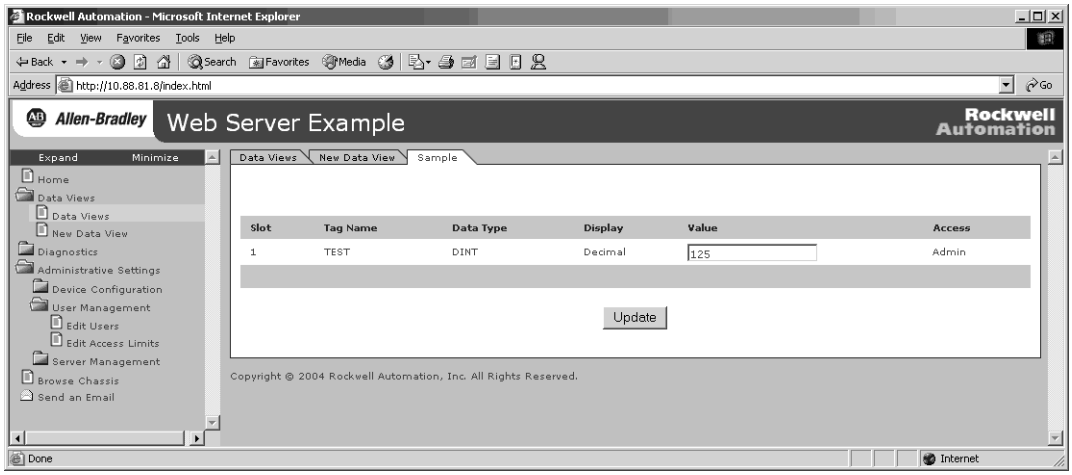
```
<html>
<head>
<title>
Data View Javascript Demo Page
</title>

<script type="text/javascript">

var xmlDoc;
var xmlhttp;

var DATAVIEW_URI = "/user/system/dataviews/Sample.xml";

var DATAVIEW_NAMESPACE = "http://www.rockwellautomation.com/technologies/data_access/data_views/1.0/";
var DATATYPES_NAMESPACE = "http://www.rockwellautomation.com/technologies/data_access/data_types/1.0/";

var ARR_ACTIVEX = ["MSXML4.DOMDocument", "MSXML3.DOMDocument", "MSXML2.DOMDocument", "MSXML.DOMDocument",
"Microsoft.XmlDom"];
var STR_ACTIVEX = "";

var isIE = navigator.userAgent.toLowerCase().indexOf("msie") > -1;

if (isIE) {
  var bFound = false;

  for (var i=0; i < ARR_ACTIVEX.length && !bFound; i++) {
    try {
      var objXML = new ActiveXObject(ARR_ACTIVEX[i]);
      STR_ACTIVEX = ARR_ACTIVEX[i];
      bFound = true
    } catch (objException) {}
  }

  if (!bFound) {
    /* No DOM found, so throw a message and go back */
   alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with XML support. Please
upgrade your browser or load the appropriate patches to support XML.");
    history.go(-1);
  }
} else {
  /* Not IE, so throw a message and go back */
  alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with XML support. Please
upgrade your browser or load the appropriate patches to support XML.");
  history.go(-1);
}

xmlDoc = new ActiveXObject(STR_ACTIVEX)
xmlDoc.async = false;
xmlDoc.setProperty("SelectionLanguage", "XPath");
```

```
if (STR_ACTIVEX.indexOf(".DOMDocument") != -1) {
  xmlhttp = new ActiveXObject(STR_ACTIVEX.substring(0, STR_ACTIVEX.indexOf(".DOMDocument")) +
".XMLHTTP");
} else {
  alert("Creating/editing a data view requires Internet Explorer 5.5 or greater with XML support. Please
upgrade your browser or load the appropriate patches to support XML.");
  history.go(-1);
}


function refresh_value() {
  xmlDoc.load(DATAVIEW_URI);


  xmlDoc.setProperty("SelectionNamespaces", "xmlns:dv='" + DATAVIEW_NAMESPACE + "'");
  document.getElementById("tagvalue").value =
xmlDoc.selectSingleNode("/dv:view/dv:tag[@name='TEST']/dv:value").childNodes[0].nodeValue;
}


function update_value() {
  xmlDoc.selectSingleNode("/dv:view/dv:tag[@name='TEST']/dv:value").childNodes[0].nodeValue =
document.getElementById("tagvalue").value;
  xmlhttp.Open("POST", DATAVIEW_URI, false);
  xmlhttp.Send(xmlDoc);
  refresh_value();
}
</script>


</head>


<body onLoad="refresh_value();">


TEST value is <input type="text" name="tagvalue" id="tagvalue" /><br/>
<input type="button" onClick="refresh_value();" value="Refresh"/><input type="button"
onClick="update_value();" value="Update"/>


</body>
</html>
```

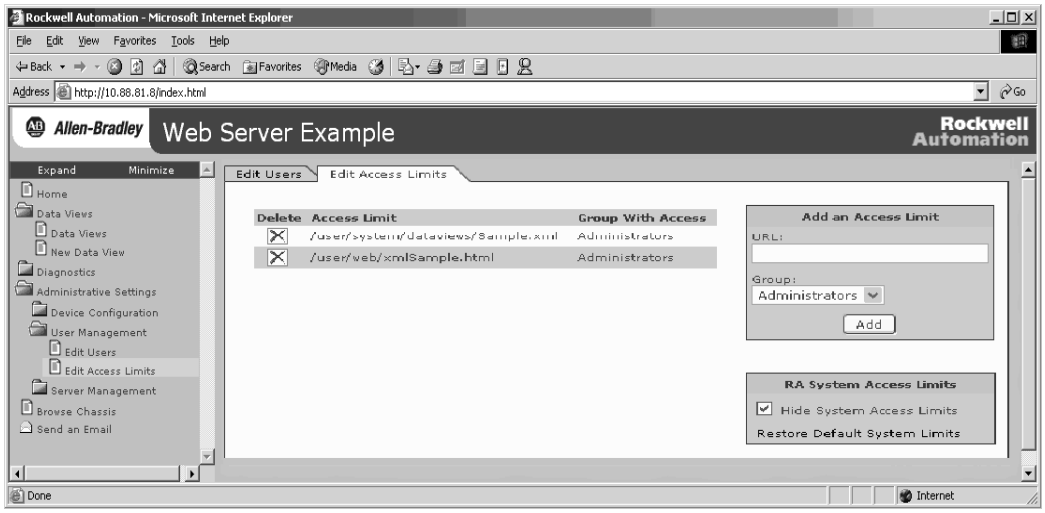
displays this screen:



In the web server module, the data view looks like this:



This example also requires a custom access level.



This method (updating the data view) is geared toward programmatic updates. When an external application that loads a data view wants to change values in that data view, the application can post a modified version of the data view to a special URL on the web server module:

1. Load the XML data view from its URL (for example, “/user/Web/system/dataviews/myview.xml”).

2. Systematically modify the values of any tags.

3. Remove any tags that were not modified from the XML document.

4. Either post the XML file as a file attachment or include its contents in a parameter named ’xml’ posted to the URL of the data view iteself.

5. Include a redirect parameter to direct a user to a page showing an up-to-date copy of the data view being modified.

Consider:

- Only tags marked as ’Write’ or ’Administrator’ can be changed. All changes to the values of ’Read’ tags are ignored.
- When the program posts an update to the data view, it must present the same user authentication that was necessary to initially load the data view.
- Any errors in the XML, whether ’valid’ (in XML) manipulations of the original document or malformed XML content, return an HTTP error to the client program.

ACTION="/rokform/WriteLogixTags"

This method is a web POST form handler that requires Administrator access. With the correct access level, a user can access and post to any tag in the controller, not just tags in a data view. You can lower the access restriction, but this opens access to all controller data to more users.

The form must include the following hidden inputs.

Hidden Input	Description
"redirect"	An URL where the browser should be redirected after the tag values have been written. Usually equal to the URL of the current page.
"numtags"	The total number of controller tags on the page.

For each controller tag, define these input fields (where "@" represents the index of the tag on the page, for example, "1" for the first tag, "2" for the second).

Input	Description
"t_@_tagname"	(Hidden) The name of the controller tag to be updated. Should be the same as the name used in a data view, or in a call to `ReadLogixTag`.
"t_@_slot"	(Hidden) The slot number of the controller containing the tag to be written.
"t_@_type"	(Hidden) The data type of the tag to be written. Should be one of the types defined in "/schema/CIPDataTypes.xsd" on the web server module (for example, "DINT").
"t_@_display"	(Hidden) The display format of the tag to be written. Should be "Decimal", "Hexadecimal", "Octal", "Binary", or "String". Both `ReadLogixTag`and `ReadLogixTagUnconnected` return numbers in decimal format. Unless you convert these numbers into another format, use "Decimal" for these values.
"t_@_changed"	(Hidden) A "0" or a "1" to indicate whether or not the value has been changed from its time of display. This can either be statically set to "1" or you can update this variable when the data field changes to indicate to the web server module that the value should be written to the controller.
"t_@_value"	(Text) The data value to be written to the controller tag.

The following example page displays a DINT tag from a controller and lets you update the tag with a Submit button.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<script language="JavaScript">
</script>
</head>
<body >
<form action="/rokform/WriteLogixTags" method="POST">
TEST:
<input type="hidden"    name="redirect          "value="/user/Web/writeOneTag.asp">
<input type="hidden"    name="numtags"          value="1">
<input type="hidden"    name="t_1_tagname"      value="TEST">
<input type="hidden"    name="t_1_slot"         value="1">
<input type="hidden"    name="t_1_type"         value="DINT">
<input type="hidden"    name="t_1_display"      value="Decimal">
<input type="hidden"    name="t_1_changed"      value="1">
<input type="hidden"    name="t_1_value"        value="0">
<!--Read the current tag value -->
<font
color=<%ReadLogixTag("1,1","test_tag_string","STRING");%>><%ReadLogixTag("1,1","TEST
","DINT");%></font>
<input type="submit" name="submit               "value="Clear Alarms">
</form>
</body>
</html>
```

ACTION="/rokform/ReadLogixTag"

This method performs an instant data table read. The response is an HTML stream.

The form accepts these parameters.

Input	Description
Path	Path to the data, such as “1,1”
Name	Name of tag you want to read Tag names are case sensitive
Type	Any of the following types: BOOL, SINT, INT, DINT, REAL or STRING
Connected (optional)	For an unconnected request, enter one of the following: FALSE,`false,F,f,0,NO,no,OFF,off` For a connected request, enter one of the following: TRUE,true,T,t,1,YES,yes,ON,on

For example:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<script language="JavaScript">
function post()
{
document.getElementById('submitit').submit();
}
</script>
</head>
<body onload="post();">
<form action="/rokform/ReadLogixTag" id="submitit" method="POST">
<input type="hidden" name="name"                value="test_tag_dint">
<input type="hidden" name="path"                value="1,1">
<input type="hidden" name="type"                value="DINT">
<input type="hidden" name="connected" value="true">
</form>
</body>
</html>
```

ACTION="/rokform/CIPMessage"

This method performs an instant CIP message request. The response is an HTML stream.

The form accepts these parameters.

Input	Description
Path	Path to the data, such as “1,1”
Service	Enter the CIP service name
Class	Enter the class ID number
Instance (optional)	Enter the instance number
Attribute (optional)	Enter the attribute number
Member (optional)	Enter the CIP member
Data (optional)	Size of the data
Type	Any of the following types: BOOL, SINT, INT, DINT, REAL or STRING
Connected (optional)	For an unconnected request, enter one of the following: FALSE,`false,F,f,0,NO,no,OFF,off` For a connected request, enter one of the following: TRUE,true,T,t,1,YES,yes,ON,on

For example:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<script language="JavaScript">
</script>
</head>
<body >
<form action="/rokform/CIPMessage" method="POST">
<input type="hidden"    name="path           "value="1,1">
<input type="hidden"    name="service"       value="5">
<input type="hidden"    name="class"         value="1">
<input type="hidden"    name="instance"      value="1">
<input type="hidden"    name="attribute"     value="">
<input type="hidden"    name="member"        value="">
<input type="hidden"    name="data"          value="">
<input type="hidden"    name="connected"     value="false">
<input type="submit"    name="submit"        value="RESET">
</form>
</body>
</html>
```

Monitor Diagnostics

About This Chapter

The web server modules provide several levels of diagnostics. There are user-oriented diagnostics, as well as more detailed diagnostics for technical support personnel. This chapter describes the diagnostics presented on the user-oriented diagnostic pages.

Web Server Module Diagnostics

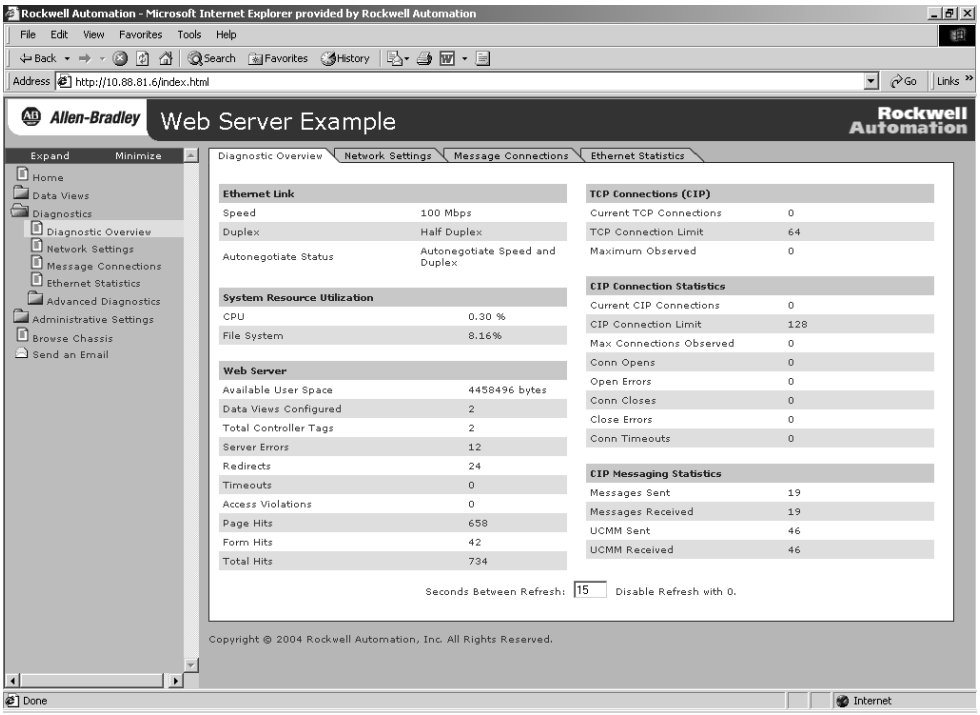
The web server module provides four diagnostic pages of user-oriented diagnostics.

Topic	Access This Web Page
Overview of the current configuration of the web server module	Diagnostics > Diagnostic Overview
Summary of the network settings configured for the web server module	Diagnostics > Network Settings
Statistics about messages initiated by the web server module and their associated connections	Diagnostics > Message Connections
Ethernet statistics	Diagnostics > Ethernet Statistics

Diagnostics Overview

The Diagnostics > Diagnostic Overview page presents a summary of the current configuration and overall status of the web server module. This summary includes:

- Ethernet configuration
- Web server file use
- Web server statistics
- TCP connection use
- CIP connection use
- Message statistics



This Field	Specifies
Ethernet Link	
Speed	Whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex	Whether the Ethernet port is operating at half duplex or full duplex
Autonegotiate Status	Whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
System Resource Utilization	
CPU	Current percent CPU utilization for the web server module
File System% Utilization	Current percent utilization of the space available for user files

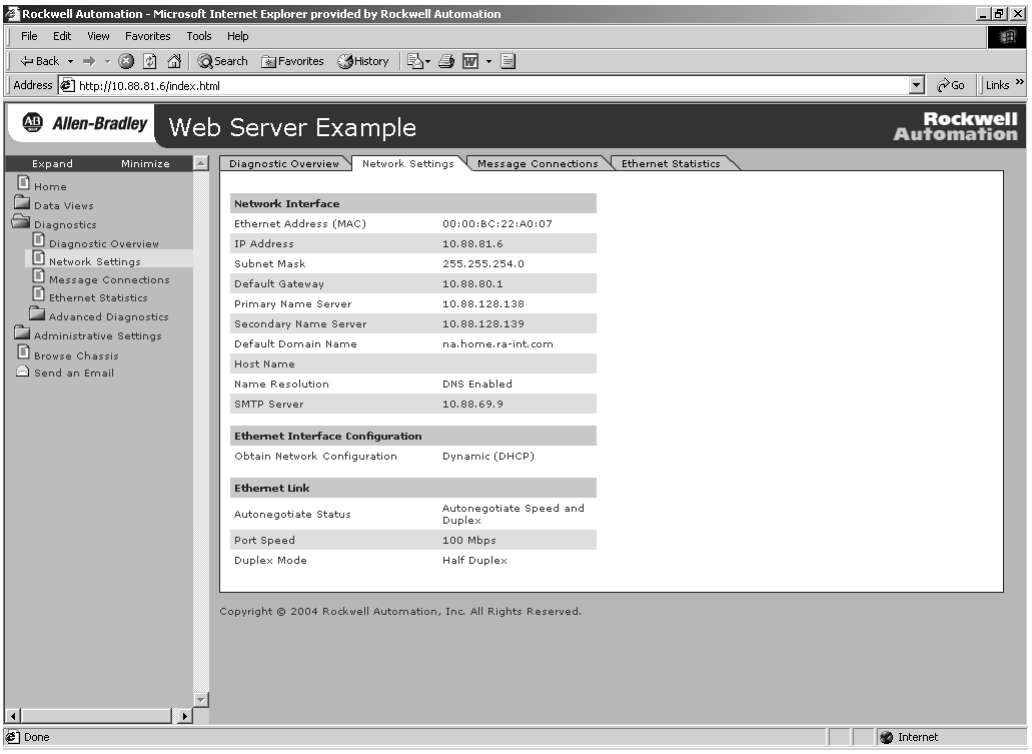
This Field	Specifies
Web Server	
Available User Space	Current number of bytes available for user files
Data Views Configured	Total number of data views configured
Total Controller Tags	Total number of tags in data views
Server Errors	Number of requests to the web server module with an invalid URL
Redirects	Number of requests for a web page that were redirected by the web server (e.g., requesting "/" is redirected to "/index.html")
Timeouts	Number of times a connection timeout occurred while processing a web page
Access Violations	Number of times a page has been requested for which the user has insufficient privilege
Page Hits	Number of times a web page was successfully accessed
Form Hits	Number of times a web page form was accessed
Total Hits	Total number of web page access attempts
TCP Connections (CIP)	
Current TCP Connections	Current number of active TCP connections for CIP messaging
TCP Connection Limit	Maximum number of TCP connections for CIP messaging allowed
Maximum Observed	Maximum observed number of TCP connections for CIP messaging
CIP Connection Statistics	
Current CIP Connections	Current number of CIP connections
CIP Connection Limit	Maximum number of CIP connections allowed
Max Connections Observed	Maximum observed number of CIP connections
Conn Opens	Number of CIP connection open requests
Open Errors	Number of CIP connection open request errors
Conn Closes	Number of CIP connection close requests
Close Errors	Number of CIP connection close errors
Conn Timeouts	Number of CIP connection timeouts
CIP Messaging Statistics	
Messages Sent	Number of CIP connected messages sent
Messages Received	Number of CIP connected messages received
UCMM Sent	Number of CIP unconnected messages sent
UCMM Received	Number of CIP unconnected messages received

CIP connections are for Logix-based communications, such as MSG instructions, RSLinx communications, and PanelView communications. A CIP connection transfers data from one Logix application running on one end-node to a second Logix application running on another end-node. A CIP connection is established over a TCP connection.

Network Settings

The Diagnostics > Network Settings page presents a summary of the current Ethernet configuration for the web server module. This summary includes:

- Ethernet address details
- Ethernet interface type
- Ethernet network configuration



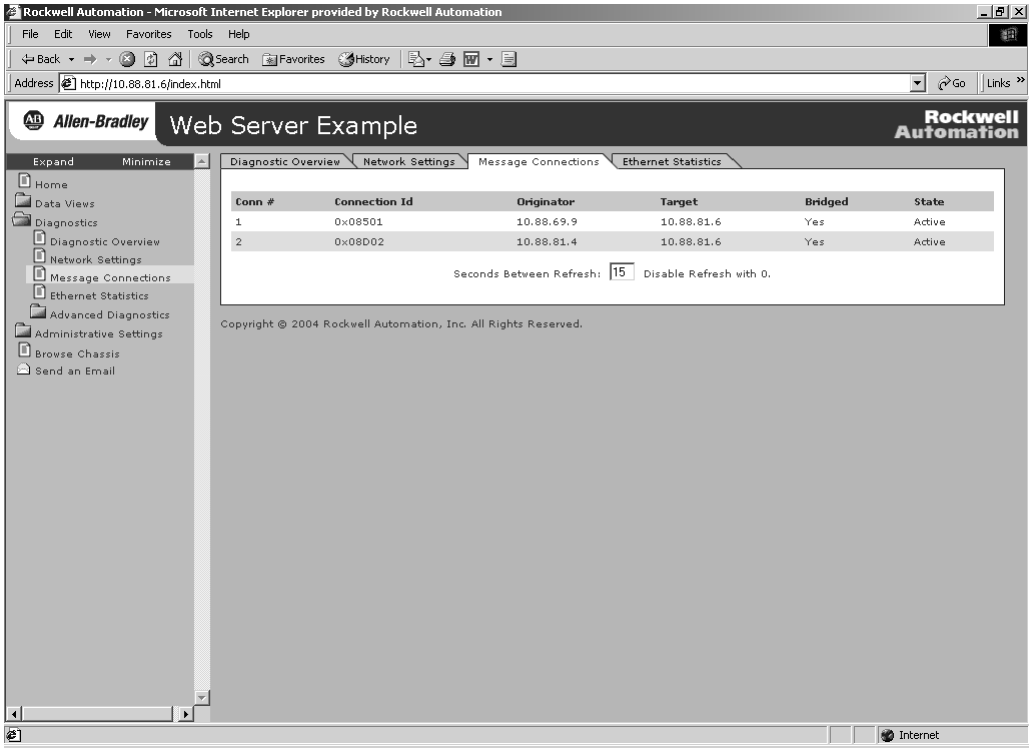
Any fields not configured remain blank.

This field	Specifies
Network Interface	
Ethernet Address (MAC)	Ethernet (MAC) address of the module
IP Address	IP address for the module
Subnet Mask	Subnet mask for the module
Default Gateway	Gateway address for the module
Primary Name Server	Primary name server
Secondary Name Server	Secondary name server
Default Domain Name	Default domain name for the module
Host Name	Host name for the module
Name Resolution	Whether Domain Name System (DNS) resolution is enabled
SMTP Server	SMTP server address for the module (required for email).

This field	Specifies
Ethernet Interface Configuration	
Obtain Network Configuration	Whether the module is configured to obtain its network parameters (such as IP address) via BOOTP, DHCP, or from static configuration
Ethernet Link	
Autonegotiate Status	Whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
Port Speed	Whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex Mode	Whether the Ethernet port is operating at half duplex or full duplex

Message Connections

The Diagnostics > Message Connections page presents a summary of messages bridged through or initiated by the web server module.



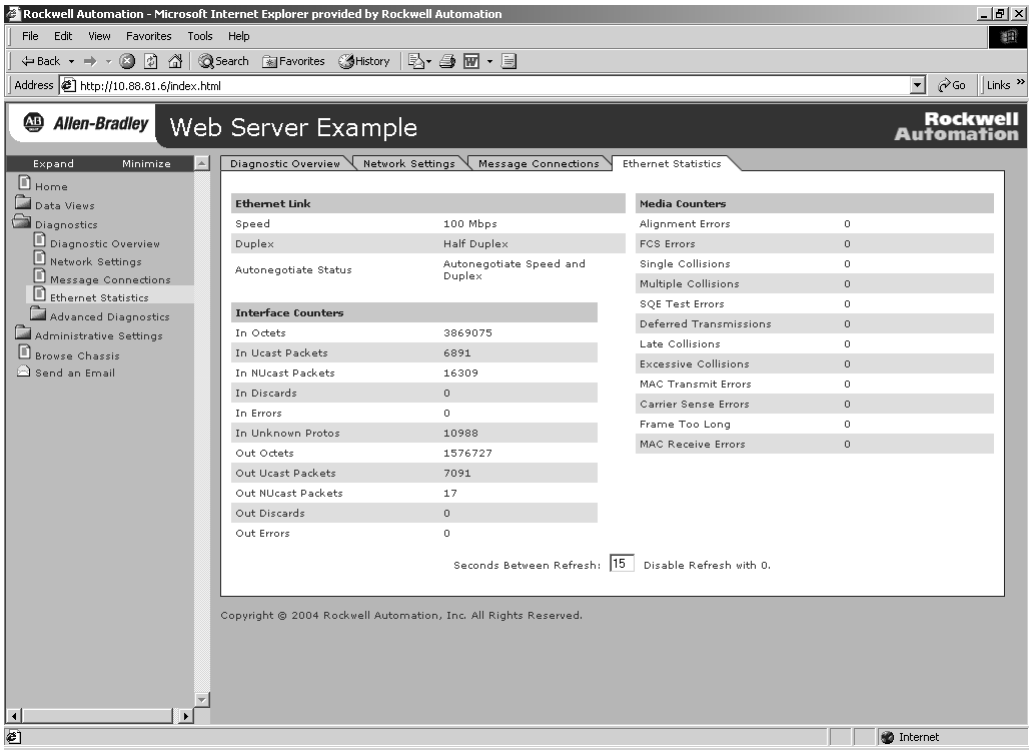
This Field	Specifies
Conn #	The relative index of this connection (on the Message Connections page)
Connection ID	The unique identifier for each connection
Originator	The IP address of the device that originated the connection on Ethernet network

This Field	Specifies
Target	The IP address of the device that is the target of the connection on Ethernet. This may not be the ultimate target of the connection (e.g., the target could be a Logix controller in a chassis).
Bridged	Whether the connection bridges through the web server module
State	The current state of the connection: • Active • Closing • Faulted • Reserved

Ethernet Statistics

The Diagnostics > Ethernet Statistics page presents a summary of the status of communication activity on the Ethernet network. This summary includes:

- Ethernet network configuration
- Packets sent and received over the Ethernet network
- Frames sent and received over the EThernet network



This Field	Specifies
Ethernet Link	
Speed	Whether the Ethernet port is operating at 10 Mbps or 100 Mbps
Duplex	Whether the Ethernet port is operating at half duplex or full duplex
Autonegotiate Status	Whether the port speed and duplex mode were determined via autonegotiation or whether they were manually configured
Interface Counters	
In Octets	Octets received on the Ethernet interface
In Ucast Packets	Unicast packets received on the Ethernet interface
In NUcast Packets	Non-unicast packets received on the Ethernet interface
In Discards	Inbound packets received on the Ethernet interface but discarded
In Errors	Inbound packets that contain errors (does not include In Discards)

This Field	Specifies
In Unknown Protos	Inbound packets with unknown protocol
Out Octets	Octets sent on the Ethernet interface
Out Ucast Packets	Unicast packets sent on the Ethernet interface
Out NUcast Packets	Non-unicast packets sent on the Ethernet interface
Out Discards	Outbound packets discarded
Out Errors	Outbound packets that contain errors
Media Counters	
Alignment Errors	Frames received that are not an integral number of octets in length
FCS Errors	Frames received that do not pass the FCS check
Single Collisions	Successfully transmitted frames which experienced exactly one collision
Multiple Collisions	Successfully transmitted frames which experienced more than one collision
SQE Test Errors	Number of times SQE test error message is generated
Deferred Transmissions	Frames for which first transmission attempt is delayed because the medium is busy
Late Collisions	Number of times a collision is detected later than 512 bit-times into the transmission of a packet
Excessive Collisions	Frames for which transmission fails due to excessive collisions
MAC Transmit Errors	Frames for which transmission fails due to an internal MAC sublayer transmit error
Carrier Sense Errors	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame
Frame Too Long	Frames received that exceed the maximum permitted frame size
MAC Receive Errors	Frames for which reception on the Ethernet interface failed due to an internal MAC sublayer receive error

Use the Web Server Module To Connect Over Ethernet

About This Appendix

A connection is a communication association between two devices. The web server modules uses different types of communications connections. Since connections use internal module resources, the web server module has limits on the number of connections it supports.

This appendix discusses these types of connections.

- CIP connections

  The web server module uses CIP connections to read data view data, to communicate with a module in the local chassis, and to communicate with another EtherNet/IP module. For example, CIP connections are used when a Logix controller sends messages through the web server module to another Logix controller over Ethernet.

- TCP/IP connections for EtherNet/IP communications

  As specified by the EtherNet/IP protocol, when the web server module communicates with another EtherNet/IP module, it first establishes a TCP connection with the target module. The TCP connection is then used for CIP connected and unconnected messages. Multiple CIP connections may be established over a single TCP connection.

CIP Connections

Connections are allocations of resources that provide more reliable communications between modules than unconnected messages.

Examples of web-server functions that use CIP connections include:

- data views (one connection per controller slot).
- sending email initiated by controller MSG instructions.
- bridging messages through the web server module.

CIP communications take priority over HTTP communications. Generally, in the presence of heavy CIP traffic, HTTP traffic may appear sluggish.

There are different types of CIP connections.

CIP Connection Type	Description
Bridged connection	A bridged connection is a connection that passes through the web server module. The end point of the connection is a module other than the web server module. Example: a connection from a controller through a 1756-ENBT to another controller.
End-node connection	An end-node connection is a connection whose end point is the web server module itself. Example: a connection from RSLinx to the web server module to set the IP address.

You cannot bridge I/O or produced/consumed tags through the web server module.

CIP Connected Messaging Limits

Product	CIP Connected Messaging Limits
1756-EWEB, 1768-EWEB	Each module supports • 128 connections for any combination of data views, bridged messages, and ASP function calls. • 32 connections, which can be end-node connections.

CIP Unconnected Messaging Limits

The following limits of unconnected messages are the maximum number of outstanding unconnected messages. These are unconnected messages that are sent to the or bridged through the web server module.

Product	CIP Unconnected Messaging Limits
1756-EWEB, 1768-EWEB	Each module supports 256 CIP unconnected messages, of which: • 128 can be unconnected messages from the Ethernet port to an object on the module or to the backplane. • 128 can be unconnected messages from the backplane to an object on the module or to the Ethernet port.

See the EtherNet/IP Performance and Application Guide, publication number ENET-AP001, for more information on connections.

TCP Connections

The web server module uses one TCP connection for each IP address to which the module is connected. Multiple CIP connections can go through a single TCP connection. Examples of TCP connections are:

- HMI (human-machine interface) to a controller that supports EtherNet/IP communications.
- Logix MSG instruction to a controller or workstation.
- OPC or DDE accessing a controller.

The web server module supports 64 TCP connections that handle CIP connections.

Additional Resources

Consult the following publication for more information.

Topic	Publication Title	Publication No.
EtherNet/IP CIP connections	EtherNet/IP Performance and Application Guide	ENET-AP001

You can view or download publications at http://www.literature.rockwellautomation.com. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

Notes:

Socket Interface

About This Appendix

The socket interface lets you use a Logix controller to communicate via the web server module with Ethernet devices that do not support the EtherNet/IP application protocol, such as bar code scanners, RFID readers, or other standard Ethernet devices.

IMPORTANT Firmware revision 3.1 or later of the 1756-EWEB module and any firmware revision of the 1768-EWEB module supports this socket interface.

Before You Begin

Before you use the socket interface, you should be familiar with:

- basic TCP/IP, UDP, and socket programming concepts.
- how to write socket programs in a programming language, such as C or Visual Basic.
- how to use diagnostic tools, such as a network sniffer.
- the application protocols of the devices and applications with which the Logix controller will communicate.
- how to write ladder logic or structured text for a Logix controller.

Socket Interface Architecture

The socket interface is implemented via the Socket Object in the web server module. Logix controller programs communicate with the Socket Object via MSG instructions. MSG requests to the Socket Object are similar to socket API calls in most computer operating systems. The Socket Object services let you open connections, accept incoming connections, send data, and receive data.

To communicate with another device, you must understand the other device's application protocol. The web server module has no application protocol knowledge – it simply makes the socket services available to programs in Logix controllers.

Number and Type of Sockets

You can create as many as 20 socket instances. Each socket instance can be one of these socket types:

- UDP socket (to send/receive UDP datagrams)
- TCP client socket (your Logix5000 program initiates the connection)
- TCP server socket (another device initiates the connection to your Logix5000 program)
- TCP listen socket (to listen on a specified port number for incoming connections)

You must have a listen socket for each TCP port number that accepts connections. Multiple TCP server sockets can share a listen socket if the connections are made to the same port number.

You can partition the 20 available socket instances between UDP and TCP sockets. For example:

- Use all 20 instances for client TCP connections
- Use one instance to listen for incoming TCP connections and then accept 19 connections from other device
- Perform both TCP client and server operations
- Perform both TCP and UDP operations

These socket services are available.

This Socket Service	The Socket Instance Is	See Page
CreateSocket	Server or client	141
OpenConnection	Client	143
AcceptConnection	• if you issue an AcceptConnection service, the instance is listen type • if the AcceptConnection service returns an instance as a result of an incoming connection request, the socket instance is server type	146
Read	Server or client	148
Write	Server or client	151
DeleteSocket	Server or client	153
DeleteAllSockets	Server or client	154

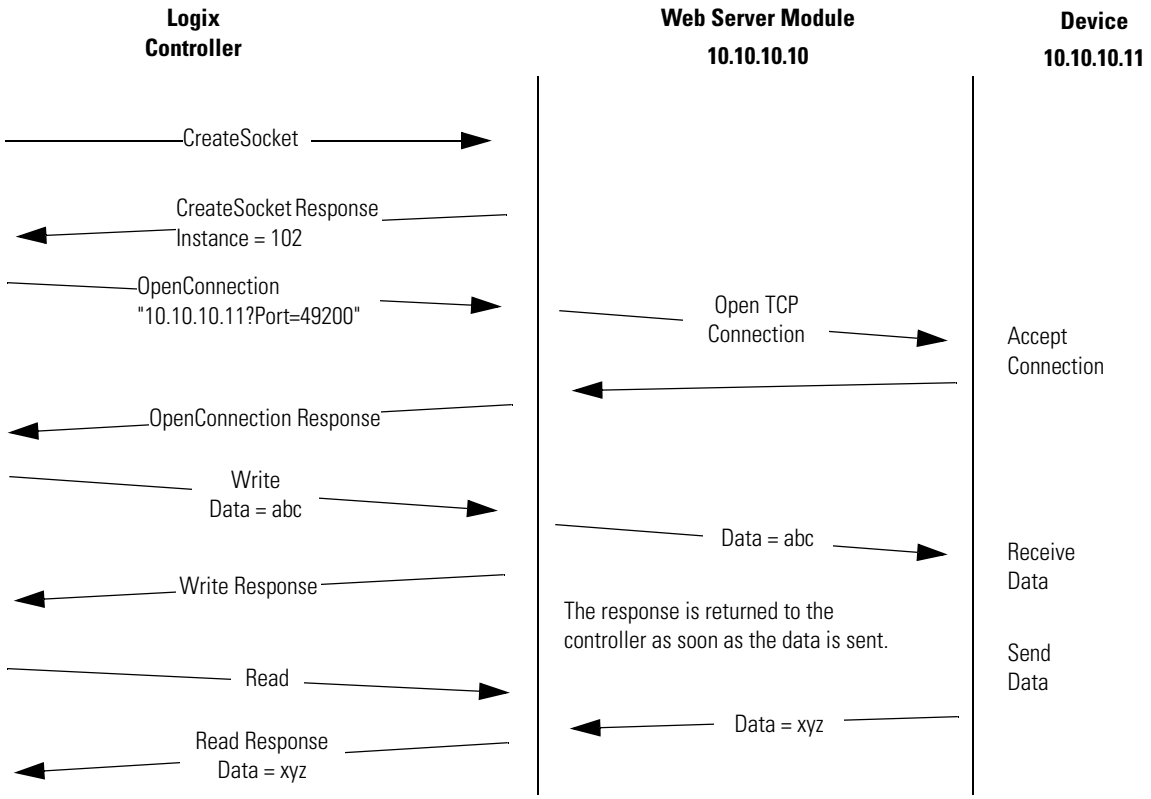
Once you open a connection on a client socket instance, you cannot use the same socket instance to accept incoming connections. Similarly, if you accept connections on a socket instance, you cannot then use the instance to open outgoing connections. This behavior is consistent with standard socket API behavior.

Typical Sequence of Transactions For a TCP Client

The following diagram shows a typical sequence of socket interface transactions with the Logix controller acting as a TCP client. Each transaction between the Logix controller and the web server is a MSG instruction.

The following example shows the Logix controller sending data to a device and then the device sending a response. This is a typical sequence of transactions. Depending on the application protocol, the device could instead initiate sending data to the Logix controller once the connection is open.

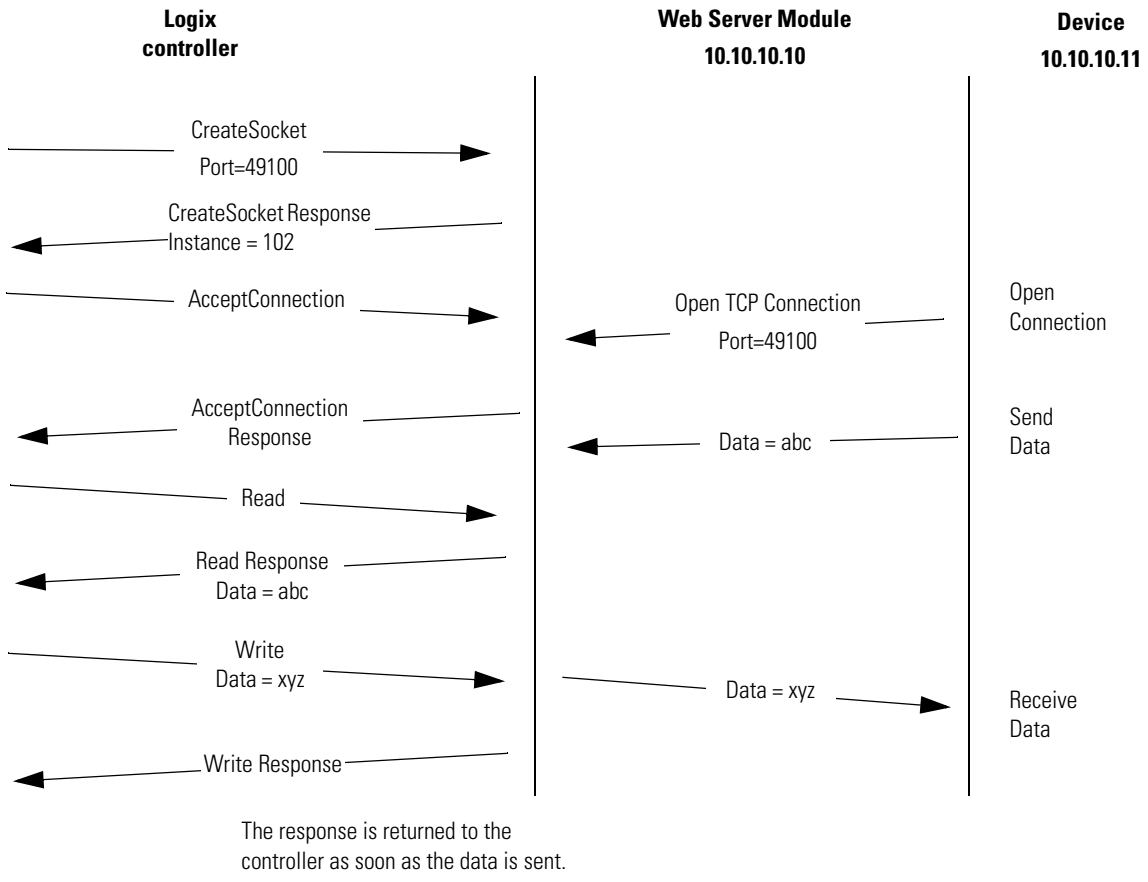
Additionally, each Write does not require an application response or acknowledgement. The application protocol determines the exact sequence of application transactions.



Typical Sequence of Transactions For a TCP Server

The following diagram shows a typical sequence of socket interface transactions with the Logix controller as a TCP server. Each transaction between the Logix controller and web server is a MSG instruction.

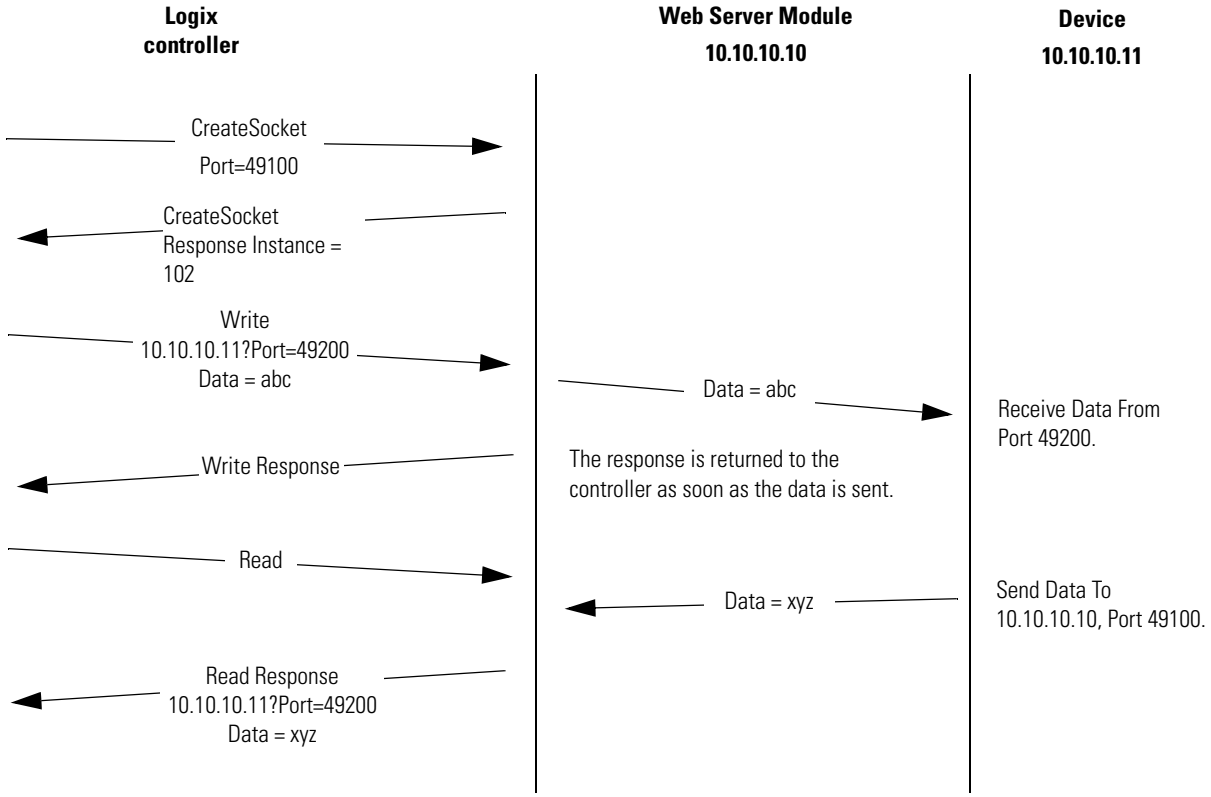
The following is a typical sequence of transactions. The exact sequence of sending and receiving data depends on the application protocol.



Typical Sequence of Transactions For UDP Without OpenConnection

The following diagram shows a typical sequence of socket interface transactions for UDP communications without using the OpenConnection service to specify the destination address. In this case, the Logix controller specifies the destination for each datagram and receives the sender's address along with each datagram it receives. Each transaction between the Logix controller and the web server is a MSG instruction.

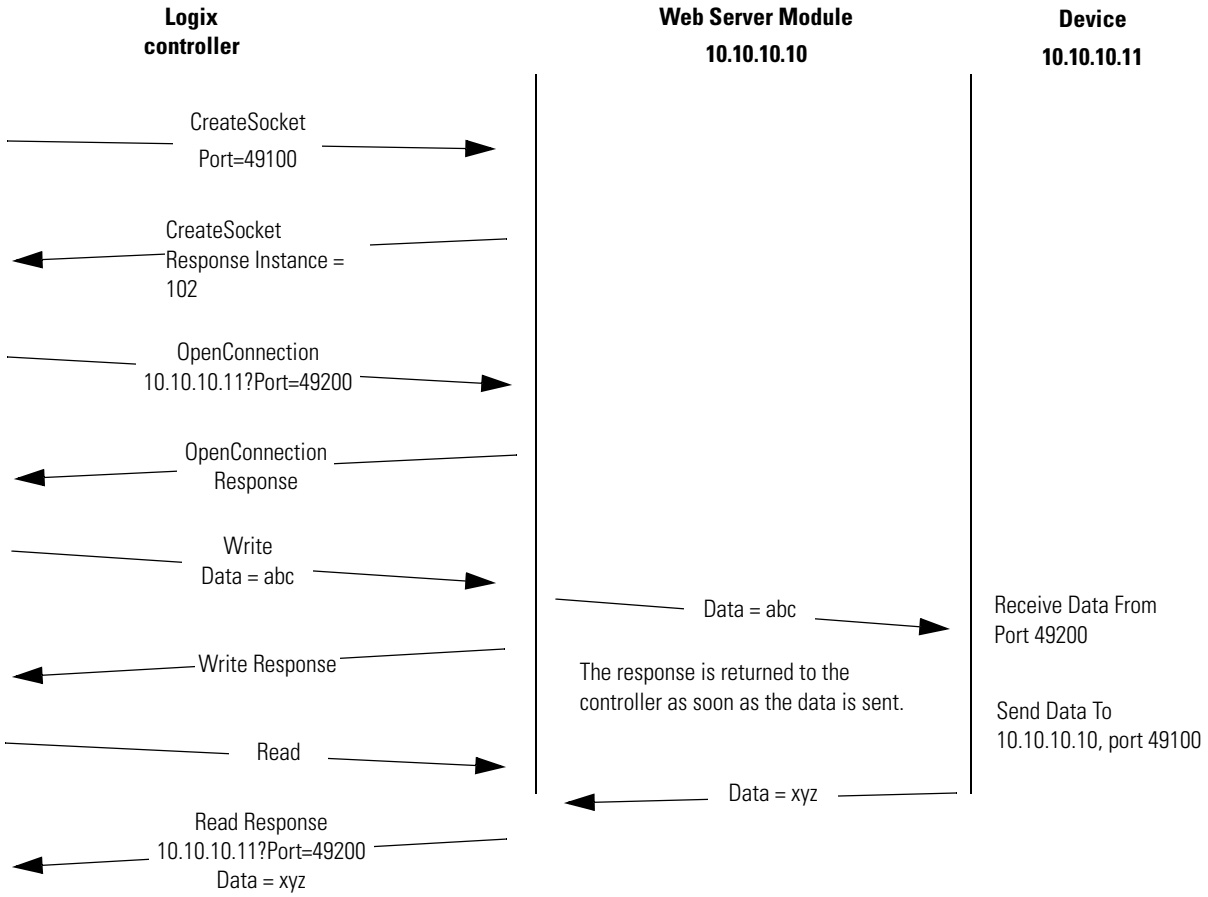
The example below shows the Logix controller sending data to a device and then the device sending a response. This is a typical sequence of transactions. Depending on the application protocol, the device could instead initiate sending data to the Logix controller. Additionally, each Write does not require an application response or acknowledgement. The application protocol determines the exact sequence of application transactions.



Typical Sequence of Transactions For UDP With OpenConnection

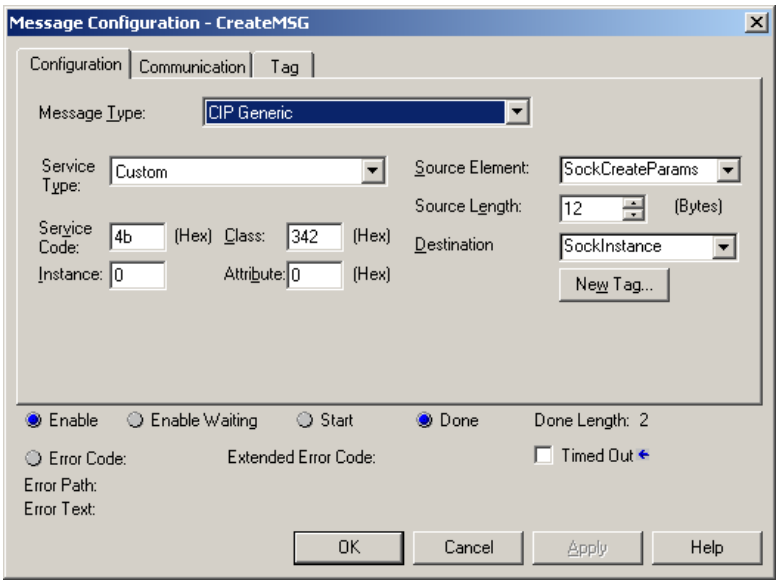
The following diagram shows a typical sequence of socket interface transactions for UDP communications when using the OpenConnection service to specify the destination address. Each transaction between the Logix controller and the web server is a MSG instruction.

The following is a typical sequence of transactions. The exact sequence of sending and receiving data depends on the application protocol.



Communicate With the Socket Object Via a MSG Instruction

In Logix controller programs, use a CIP Generic MSG instruction to request socket services.



Configure the MSG with these values.

CIP Generic Msg Parameter	Description
MSG Type	CIP Generic.
Service Type	With RSLogix software version 15, select a socket service type from the drop down. The software automatically fills in the appropriate Service Code and Class values. With RSLogix 5000 software prior to version 15, select Custom. You have to specify the Service Code and Class values. Select Get Attributes Single or Set Attributes Single when getting or setting a Socket Object attribute (see Socket Attributes on page 157)
Service Code	Each socket service has a unique service code.
Class	Specify 342 (hexadecimal) for the Socket Object
Instance	Specify: • 0 for CreateSocket and DeleteAllSockets services • Instance number returned by CreateSocket for other services Use a relay ladder instruction or structured text statement to move the returned instance number from a CreateSocket service into the .Instance member of a MSG instruction.
Attribute	Specify the attribute value only when getting or setting an attribute, not when using other services.
Source Element	Contains the request parameters for the socket service. Create a user defined type for the request parameters for each service.

CIP Generic Msg Parameter	Description
Source Length	Specify the length of the Source Element.
Destination	Contains the response data returned by the service. Create a user defined type for the response data for each service.
Communication	The communication path must point to the web server.

For details on specific services, see See Socket Object Services on page 140.

Message Transfer Sizes

The maximum amount of application data you can send or receive depends on whether you configure the MSG instruction as connected or unconnected. The size of the application data does not include the parameters in the Read and Write services.

Service	Connected or Unconnected (Configured On the MSG Communications tab)	Maximum Amount Of Application Data
Read	Connected	484 bytes
	Unconnected	484 bytes
Write	Connected	472 bytes
	Unconnected	462 bytes

For TCP sockets, if the application data is larger than the maximum size, you can issue multiple Reads or Writes to receive or send the entire application message.

For UDP sockets, the size application data cannot exceed the maximums listed for Read and Write services.

For more information on MSG instructions, see:

- Logix5000 Controllers General Instructions Reference Manual, 1756-RM003
- Logix5000 Controllers Common Procedures Programming Manual, 1756-PM001
- EtherNet/IP Modules in Logix5000 Control Systems User Manual, ENET-UM001

Service Timeouts

You must specify a Timeout parameter (in milliseconds) for any service that might not complete immediately (OpenConnection, AcceptConnection, Read, and Write). The timeout tells the Socket Object the maximum amount of time it should wait when attempting to complete the service. While waiting for the service to complete, the MSG instruction is enabled.

If the requested service does not complete before the Timeout period expires, the Socket Object returns a response to the service request. See the service descriptions for the content of the response.

IMPORTANT Make the value of the service Timeout parameter shorter than the MSG instruction timeout, otherwise application data could be lost

MSG Instruction Timeouts

The default MSG instruction timeout is 30 seconds; the maximum MSG timeout is approximately 35 minutes. Specify the MSG instruction timeout by setting the appropriate member of the MSG tag:

- .UnconnectedTimeout member if the MSG is unconnected.
- .ConnectionRate and .TimeoutMultiplier if the MSG is connected.

The MSG timeout is determined by multiplying the ConnectionRate by the TimeoutMultiplier. (Note: TimeoutMultiplier of 0 corresponds to multiplier of 4, 1 corresponds to multiplier of 8, and so on).

Socket Instance Timeouts

Each socket instance has an Inactivity Timeout (default of 5 minutes). If a socket instance receives no service requests for the amount of time specified by the Inactivity Timeout, the socket instance is deleted. If you then try to use the socket instance, the MSG instruction receive the error Class or instance not supported.

You can change the timeout by setting the Inactivity Timeout attribute via the Set Attribute service.

See Socket Object Instance Attributes on page 159.

If you put the controller in Program mode and then back into Run mode before existing socket instances time out, you can receive errors when the program tries to create socket instances. Eventually the socket instances time out and you can create more instances.

IMPORTANT Make sure the Inactivity Timeout is longer than the longest interval between socket operations. If the Inactivity Timeout is too short, socket instances may time out, resulting in MSG instruction errors.

Programming Considerations

Observe these programming considerations.

TCP Connection Loss

Your application program may encounter conditions that result in TCP connection loss. For example, a network cable can be unplugged, or a target device can be turned off.

Your application program should detect the loss of TCP connections and handle those events appropriately. You can detect connection loss when a:

- Read service returns with an error.
- Write service returns with an error other than EWOULDBLOCK. See Possible Error Codes for Socket Services on page 155.

Depending on the application, you might want to:

- fault the controller.
- try to re-establish the connection (in the case of a client connection).
- wait for another incoming connection to be established (in the case of a server connection).

If you want to re-establish communications with the other device, you must:

- delete the socket instance for the lost connection.
- if the connection is a client connection, create a new socket instance and issue an OpenConnection service to the target device.
- if the connection is a server connection, issue an AcceptConnection service to wait for another connection from the remote device.

Web Server Module Reset

If the web server module is reset, for example, by cycling power or with Removal and Insertion Under Power (RIUP), all socket instances are lost.

If you create new socket instances while MSG instructions are still using the the old instance numbers, there is a possibility that the new instance numbers will match the old instance numbers. In this

situation, your (old) MSG instructions may succeed but may not be communicating with the correct remote device.

Handle this situation by monitoring the status of the web server via a GSV instruction. If you lose communication with the web server, the Logix5000 program should re-initialize its socket communications.

For more information on the GSV instruction, see the Logix5000 Controllers Common Procedures Programming Manual publication number 1756-PM001.

Change Controller Mode Between Run and Program

If the Logix controller transitions from Run to Program mode while socket requests are active, the transition does not complete until all the outstanding MSG requests complete or time out. If you have long timeout values, you can experience an unexpectedly long time for the Run-to-Program transition to complete.

Alleviate long transition times by appropriately setting the Timeout parameter for the socket services. In the Logix5000 program, you can also set the .TO bit for any outstanding socket-related MSG instruction. This causes the MSG instruction to timeout and set the .ER bit.

If the controller transitions from Run mode to Program mode, then back to Run mode again, previous socket instances may still exist on the web server. The previous socket instances time out eventually. Depending on the number of sockets you need, your program may encounter errors during Run-Program-Run transitions because all of the available socket instances are in use. To alleviate this situation:

1. Wait for all socket instances to time out before putting the controller in Run mode.

2. When the Logix5000 program starts, use the DeleteAllSockets service to delete any previous instances.

The DeleteAllSockets service deletes ALL socket instances, not just those created by the controller calling the service.

Application Messages and TCP

A TCP connection is a byte stream between two application entities. The application protocol determines the message formats. Messages can be fixed size or variable size.

If an application sends variable size messages, a common strategy is to first send a fixed size 'header' containing the size of the message, followed by the message. The receiving device can first issue a Read of the fixed size header to determine the remaining size, and then issue a subsequent Read to receive the remaining data.

Partial Reads

It is possible for a Read service to return a BufLen that is less that the requested amount of data. For example, your program may request 100 bytes of data. Because TCP is a byte stream and not a datagram protocol, you can receive less than 100 bytes when the Read service returns.

Depending on the application protocol, issue additional Read requests to receive all the data. If the application protocol dictates that all messages are 100 bytes, then you must issue additional Reads until you receive 100 bytes. If the application protocol uses variable size messages, your program needs additional logic to handle variable message sizes as defined by the application protocol.

When issuing multiple Read services, be careful to adjust the destination tag that receives the data so that data is not overwritten. This fragment of Structured Text logic shows an example of handling a partial Read service.

```
/* copy the message we just read */

COP ( ReadResponse.Buf[0], ReadBuf[CurrentLen], ReadResponse.BufLen );

CurrentLen := CurrentLen + ReadResponse.BufLen;

/* do we need to read more data get a complete message? */

if ( CurrentLen < ApplicationMsgLen ) then

/* issue another read */

ReadParams.BufLen := ApplicationMsgLen - CurrentLen;

MSG ( ReadMSG0 );

       end_if;
```

Partial Writes

Your program may need to handle the situation, although uncommon, where the Write service is unable to send all the specified bytes. Such a situation can occur if the Write service is called multiple times before the target application can receive the data.

If the Write service is not able to send all of the requested data, your program should issue subsequent Writes to send the remaining data. Your program should also adjust the source tag, so that old data is not sent.

This fragment Structured Text logic shows an example of handling a partial Write service.

```
if ( WriteMSG0.ER ) then

/* write failed. if error was EWOULDBLOCK then it means less than the requested byte were sent.
*/

if ( WriteMSG0.EXERR = 70 ) then

/* need to issue another write, with the data that was not sent */

       SentLen := WriteResponse;  /* here's what was sent */

       /* adjust the size  */

       WriteParams.BufLen := WriteParams.BufLen - SentLen;

       /* copy remaining data to send to MSG buffer */

       COP ( WriteBuf[SentLen], WriteParams.Buf[0], WriteParams.BufLen );

/* BufLen = Timeout + Sockaddr + data length */

WriteMSG0.REQ_LEN := 4 + 12 + WriteParams.BufLen;

       MSG ( WriteMSG0 );

end_if;

       end_if;
```

Socket Object Services

The Socket Object supports these services.

For This Socket Service	See Page
CreateSocket	141
OpenConnection	143
AcceptConnection	146
Read	148
Write	151
DeleteSocket	153
DeleteAllSockets	154

From the perspective of the Socket Object, application data has no inherent byte order. The service receives data in the same byte order as it is sent. However, the Logix controllers stores data on CIP byte order (little endian). For example, if you issue a Write service with 1 DINT, that DINT is sent over a TCP connection or in a UDP datagram in CIP byte order. If you issue a Read service and your destination tag (for the response) contains a DINT, the Logix controller assumes the incoming data is in CIP byte order. Depending on the native byte order of the application you are communicating with, you may have to convert the byte order in your Logix5000 program and/or in the application.

To check your MSG configuration in RSLogix software version 15 or later, select a service type from the drop down when you configure the MSG instruction (Configuration tab). The software automatically fills in the appropriate Service Code and Class values. With RSLogix 5000 software prior to version 15, select Custom. You have to specify the Service Code and Class values.

IMPORTANT

With RSLogix software version 15, you can select a socket service type from the drop down. The software automatically fills in the appropriate Service Code and Class values.

With RSLogix 5000 software prior to version 15, you must select Custom for the service type. Then specify the Service Code and Class value for the appropriate service.

CreateSocket

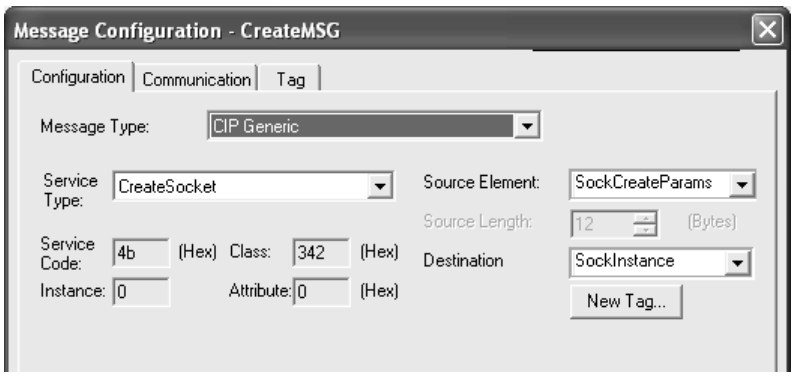
The CreateSocket service creates an instance of the Socket object. The service returns an instance number that you use in the subsequent socket operations.

Call the CreateSocket service with instance 0 (Socket object class).

MSG Configuration Parameters

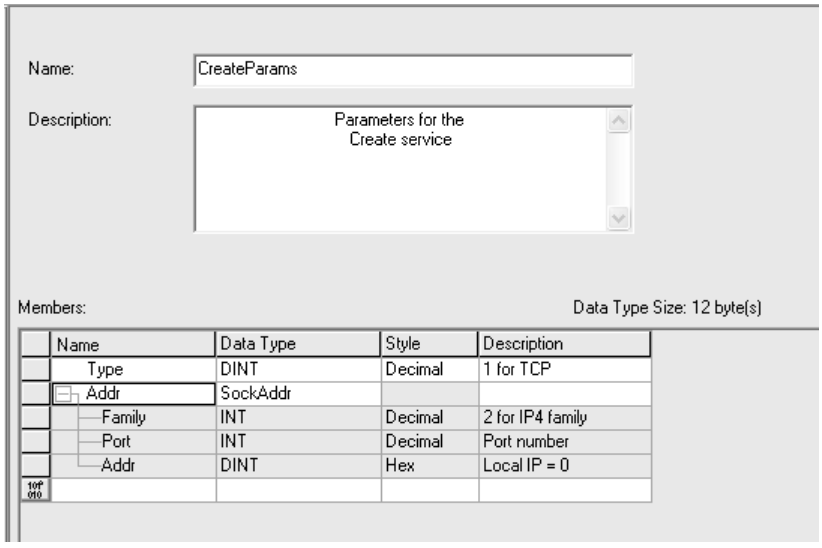
To call this service, click ... in the MSG instruction and enter the values.

Parameter	Value
Service Type	CreateSocket
Service Code	4b
Class	342
Instance	0
Attribute	0



MSG Source Element

Create a user-defined data type.



Element	Data Type	Description
Type	DINT	Specify: • 1 for TCP • 2 for UDP
Addr	structure	A user-defined structure that specifies the address for the socket.
Family	INT	Specify the address family. Must be 2.
Port	INT	Specify a local port number, or set to 0 (the local port number will be chosen by the web server). For TCP client operations, specify 0 unless you want a specific local port number. For TCP server communications, specify the port number on which to accept incoming connection requests. For UDP, to receive datagrams on a specific port, you must specify a local port number.
Addr	DINT	Specify an IP address. Typically, set to 0 (any address).

Select the data type you created in the Source Element field of the Message Configuration dialog.

MSG Source Length

Specify the size of the user-defined structure for the Source Element. In this example, CreateParams is 12 bytes.

MSG Destination Element

The MSG instruction returns the instance number of the socket it just created to the Destination Element. Specify a DINT tag (in this example, SockInstance).

Considerations

Use the instance returned by CreateSocket on subsequent service requests.

Use a MOV instruction to move the instance to another MSG tag (the .Instance field).

If you use a local port number that is already in use by the web server, you receive an EADDRINUSE error (see Error Codes section). The web server uses these port numbers:

- 20, 21 – FTP
- 25 – SMTP
- 80 – HTTP
- 123 – NTP
- 161 – SNMP
- 2222 – EtherNet/IP
- 44818 – EtherNet/IP

OpenConnection

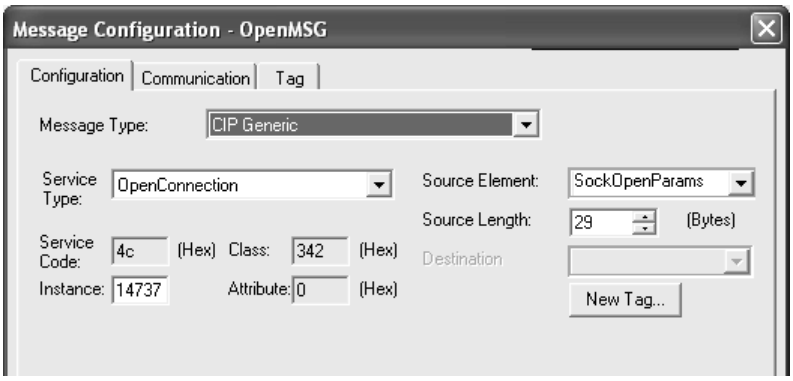
The OpenConnection service does one of the following:

- Opens a TCP connection with the specified destination address
- For UDP, associates a destination IP address and port number with the specified socket

MSG Configuration Parameters

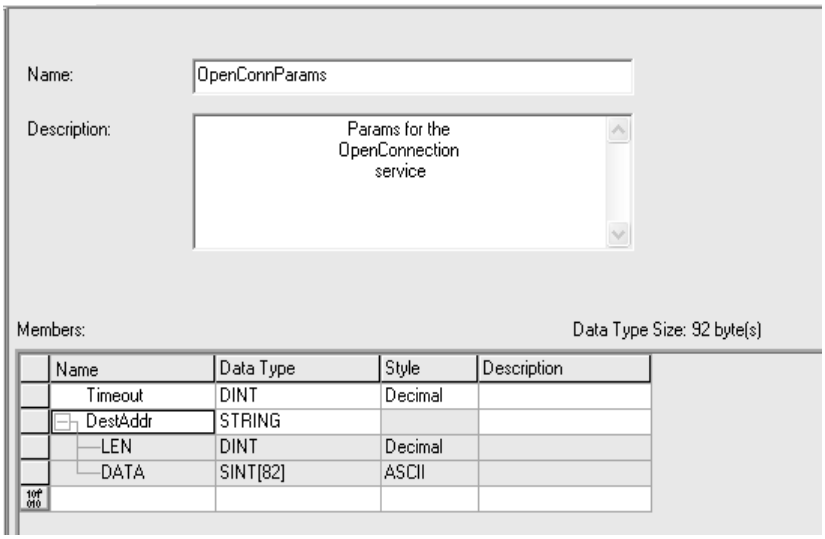
To call this service, click ... in the MSG instruction and enter the values displayed below.

Parameter	Value
Service Type	OpenConnection
Service Code	4c
Class	342
Instance	from CreateSocket
Attribute	0



MSG Source Element

Create a user-defined data type.



where:

Element	Element	Description
Timeout	DINT	Specify the timeout in milliseconds
DestAddr	STRING	Specify an array of characters (maximum of 64) to define the destination of the connection. Specify either of these: • Hostname?port=xxx • IPAddr?port=xxx For example, to specify an IP address, enter 10.88.81.10?port=2813
.LEN	DINT	The length of the destination address.
.DATA	SINT array	The array containing the destination address.

You can use a single string tag for the combination of AddrLen and DestAddr (since the string data type is a DINT followed by the string characters). Note that the quotes displayed on the RSLogix 5000 Data Monitor are not counted in the string length.

The MSG instruction that issues the OpenConnection service should have a Source Length of 8 (Timeout + AddrLen) plus the number of characters in the destination address.

MSG Source Length

Specify 8 bytes (Timeout + AddrLen) + number of characters in the destination address

MSG Destination Element

Not used; the MSG instruction does not return any data.

Considerations

In some cases, the OpenConnection service can return before the timeout period without creating a TCP connection. For example, if the destination device is running, but is not listening for connections on the specified port number, OpenConnection returns with an error before the timeout period.

For UDP, if you use OpenConnection, you do not have to specify the IP address and port number each time you send data. If you do not specify an IP address and port number, you can only receive data from the previously specified IP address and port number until you call OpenConnection to specify a different IP address and port number.

For UDP, if you do not use OpenConnection, you must specify the destination address each time you call the Write service to send data. When you call the Read service, in addition to the data, you receive the address of the sender. You can then use the address of the sender to send a response via the Write service.

If you call OpenConnection on a UDP socket with an AddrLen of 0, this removes the association with the destination address.

AcceptConnection

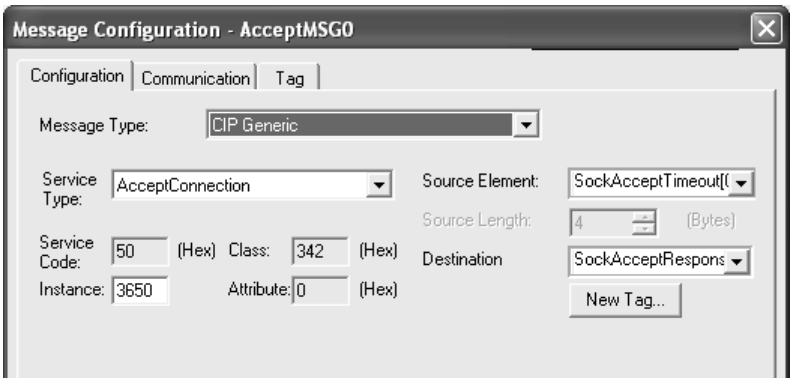
The AcceptConnection service accepts a TCP connection request from a remote destination. Before calling AcceptConnection, call CreateSocket and specify the local port number that will accept the connection. When AcceptConnection completes, it returns a socket instance that you use for sending and receiving data on the newly-created connection.

The AcceptConnection service is not valid for UDP sockets.

MSG Configuration Parameters

To call this service, click ... in the MSG instruction and enter the values displayed below.

Parameter	Value
Service Type	AcceptConnection
Service Code	50
Class	342
Instance	from CreateSocket
Attribute	0



MSG Source Element

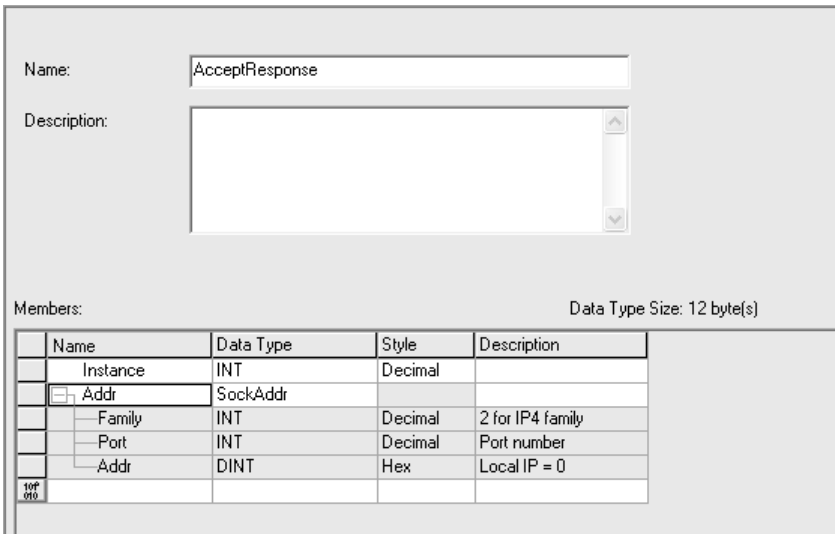
Specify a DINT tag to contain the timeout in milliseconds.

MSG Source Length

Specify 4 bytes (Timeout).

MSG Destination Element

Create a user-defined data type.



where:

Element	Element	Description
Instance	DINT	Contains the instance for this service. Use this Instance on subsequent Read and Write services for this connection.
Addr	structure	A user-defined structure that contains the address for the socket.
Family	INT	Contains the address family. Must be 2.
Port	INT	Contains a local port number.
Addr	DINT	Contains an IP address.

Considerations

Create a separate socket instance (CreateSocket) for each port number that will accept connections. After you create a socket instances, call AcceptConnection to wait for an incoming connection request. You can accept connections on the same port number. Each call to AcceptConnection returns a different instance number to use when subsequently reading and writing data.

Read

The Read service reads data on a socket. You specify a number of bytes to receive. The Read service returns the number of bytes received.

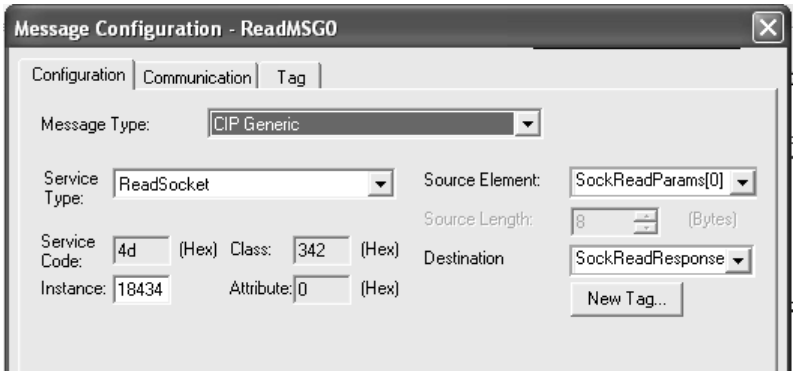
For TCP, the Read service returns when any data is received, up to the requested number of bytes. If no data is received before the timeout period, the Read service returns a status of success (0) and a BufLen of 0. The Read service can return fewer bytes than were requested. Your application might need to issue multiple Read requests to receive an entire application message.

For UDP, the Read service completes when a datagram is available.

MSG Configuration Parameters

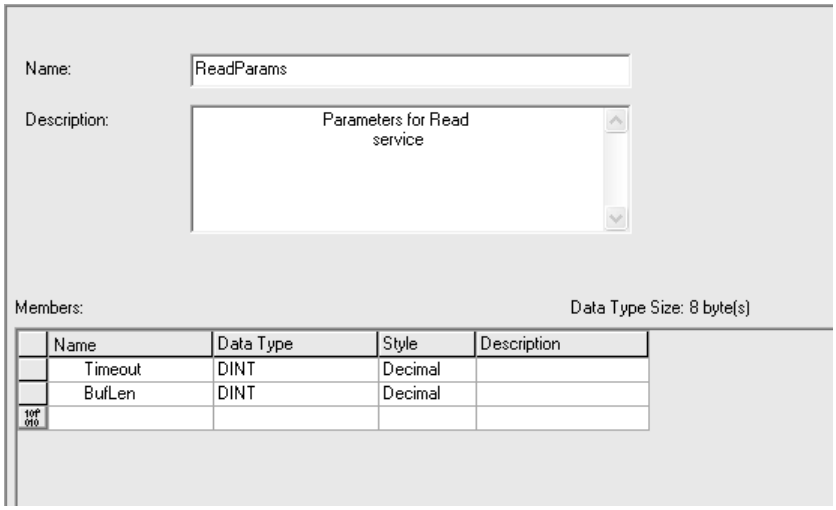
To call this service, click ... in the MSG instruction and enter the values displayed below.

Parameter	Value
Service Type	Read
Service Code	4d
Class	342
Instance	from CreateSocket
Attribute	0



MSG Source Element:

Create a user-defined data type.



where:

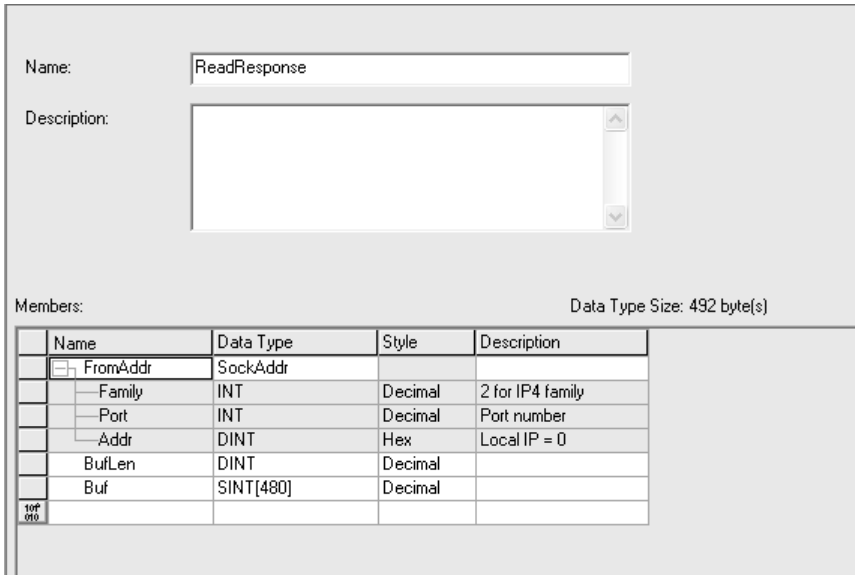
Element	Element	Description
Timeout	DINT	Specify the timeout in milliseconds
BufLen	DINT	Specify the number of bytes of data to receive.

MSG Source Length

Specify 8 bytes (Timeout + BufLen)

MSG Destination Element

Create a user-defined data type.



where:

Element	Element	Description
FromAddr	structure	A user-defined structure that contains the address to read.
Family	INT	Contains the address family. Must be 2.
Port	INT	Contains a local port number.
Addr	DINT	Contains an IP address.
BufLen	DINT	Contains the number of bytes of data received.
Buf	SINT array	Contains the data.

Considerations

See Programming Considerations on page 135 for more information.

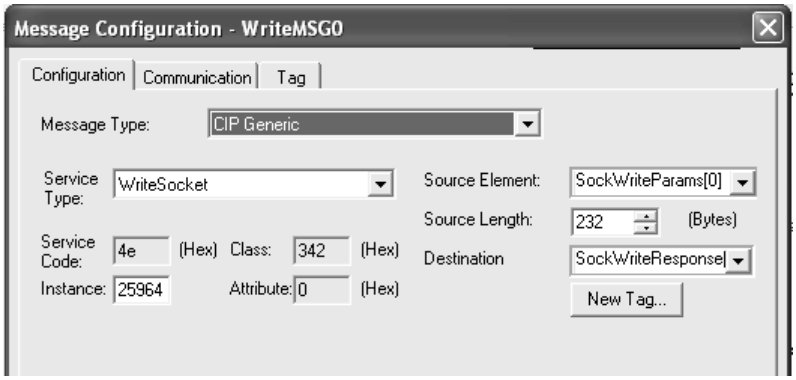
Write

The Write service sends data on a socket. You specify the number of bytes to send. The Write service attempts to send the requested number of bytes and returns the number of bytes sent.

MSG Configuration Parameters

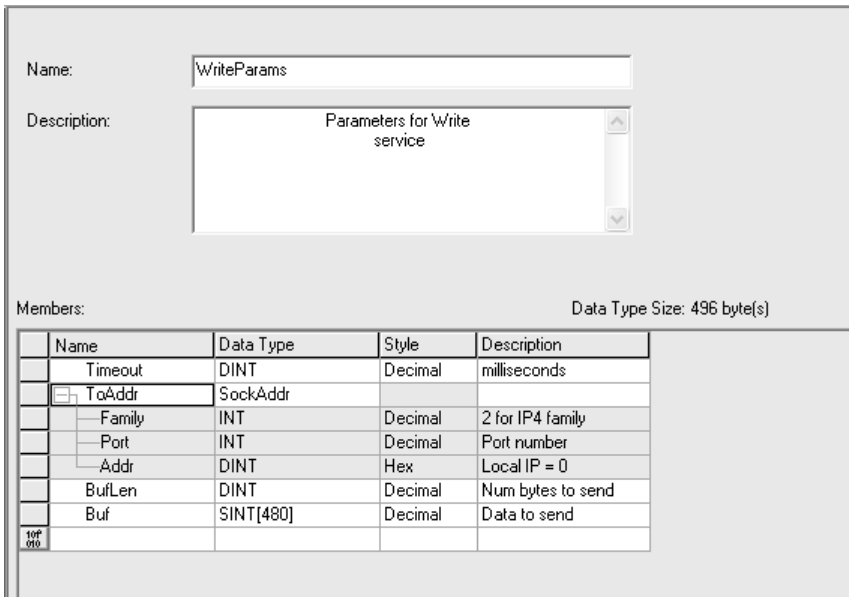
To call this service, click ... in the MSG instruction and enter the values displayed below.

Parameter	Value
Service Type	Write
Service Code	4e
Class	342
Instance	from CreateSocket
Attribute	0



MSG Source Element

Create a user-defined data type.



where:

Element	Data Type	Description
Timeout	DINT	Specify the timeout in milliseconds
ToAddr	structure	A user-defined structure that specifies the address to write.
Family	INT	Specify the address family. Must be 2.
Port	INT	Specify a local port number, or set to 0 (the local port number will be chosen by the web server). For TCP client operations, specify 0 unless you want a specific local port number. For TCP server communications, specify the port number on which to accept incoming connection requests. For UDP, to receive datagrams on a specific port, you must specify a local port number.
Addr	DINT	Specify an IP address. Typically, set to 0 (any address).
BufLen	DINT	Specify the number of bytes of data to write.
Buf	SINT array	Specify the data to write.

MSG Source Length

Specify 16 bytes (Timeout + Addr + BufLen) + number of bytes to write

MSG Destination Element

The MSG instruction returns the number of bytes that were written. Specify a DINT tag.

Considerations

See Programming Considerations on page 135 for more information.

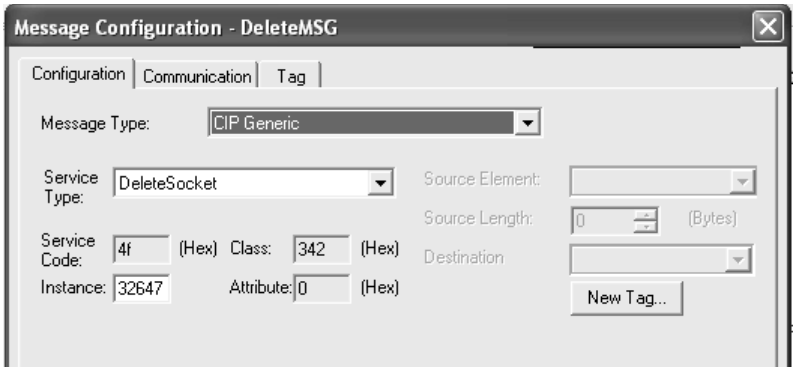
DeleteSocket

The DeleteSocket service deletes a socket instance. For a TCP connection, the DeleteSocket service also closes the connection prior to deleting the instance.

MSG Configuration Parameters

To call this service, click ... in the MSG instruction and enter the values displayed below.

Parameter	Value
Service Type	DeleteSocket
Service Code	4f
Class	342
Instance	from CreateSocket
Attribute	0



MSG Source Element

Not used.

MSG Source Length

0

MSG Destination Element

Not used.

Considerations

Delete a socket instance if it is no longer needed. If unused instances are not deleted and you continue to create additional instances, you can exceed the maximum number of instances.

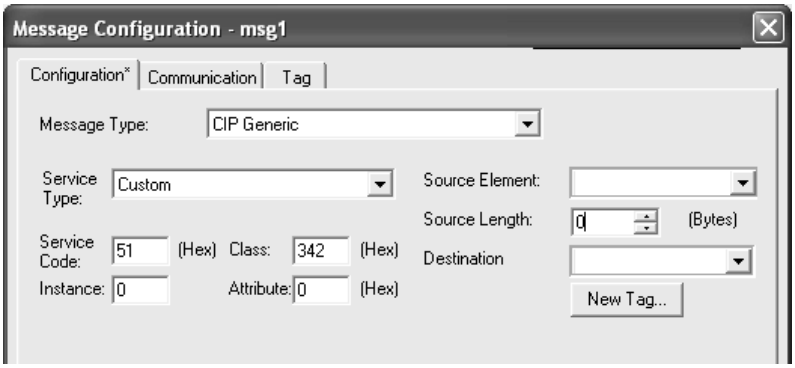
DeleteAllSockets

The DeleteAllSockets service deletes all currently created socket instances. For TCP, the DeleteAllSockets service also closes all connections prior to deleting the instances.

MSG Configuration Parameters

To call this service, click ... in the MSG instruction and enter the values displayed below.

Parameter	Value
Service Type	Custom
Service Code	51
Class	342
Instance	0
Attribute	0



You must specify Custom for the Service Type. There is no drop-down selection for Delete All Sockets.

MSG Source Element

Not used.

MSG Source Length

0

MSG Destination Element

Not used.

Considerations

Call the DeleteAllSockets service with instance 0.

IMPORTANT Be careful when using DeleteAllSockets when there are multiple controllers using the socket interface of the web sever module. DeleteAllSockets deletes all socket instances created by all controllers, not just the controller calling the service.

A typical use of DeleteAllSockets is when application development and debugging is complete. Use DeleteAllSockets as the first operation when the program first begins to operate.

Possible Error Codes for Socket Services

If the Socket Object encounters an error with a service request, or while processing the service request:

- The Socket Object returns an error code.
- The MSG instruction sets the .ER bit.
- The MSG instruction sets error codes in the .ERR and .EXTERR fields.

CIP error status codes are:

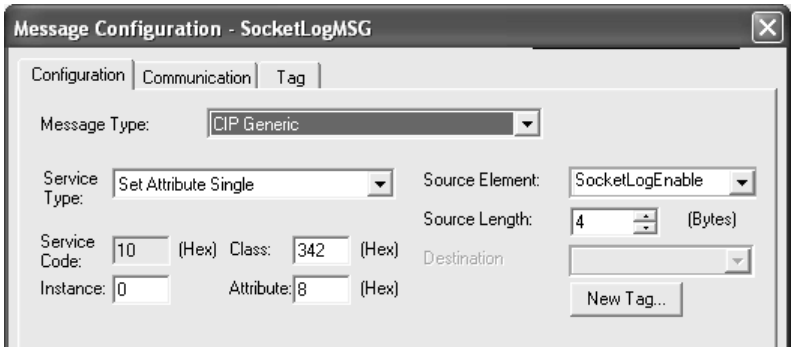
Error Code		Mnemonic	Description
Decimal	**Hexadecimal**		
22	16	EINVAL	Invalid argument
32	20	EPIPE	Broken pipe
40	28	EDESTADDRREQ	Destination address required
41	29	EPROTOTYPE	Protocol wrong type for socket
42	2A	ENOPROTOOPT	Protocol not available
43	2B	EPROTONOSUPPORT	Protocol not supported
44	2C	ESOCKTNOSUPPORT	Socket type not supported
45	2D	EOPNOTSUPP	Operation not supported on socket
46	2E	EPFNOSUPPORT	Protocol family not supported
47	2F	EAFNOSUPPORT	Addr family not supported
48	30	EADDRINUSE	Address already in use
49	31	EADDRNOTAVAIL	Can't assign requested address
50	32	ENOTSOCK	Socket operation on non-socket
51	33	ENETUNREACH	Network is unreachable
52	34	ENETRESET	Network dropped connection on reset
53	35	ECONNABORTED	Software caused connection abort
54	36	ECONNRESET	Connection reset by peer
55	37	ENOBUFS	No buffer space available
56	38	EISCONN	Socket is already connected
57	39	ENOTCONN	Socket is not connected
58	3A	ESHUTDOWN	Can't send after socket shutdown
59	3B	ETOOMANYREFS	Too many references: can't splice
60	3C	ETIMEDOUT	Connection timed out
61	3D	ECONNREFUSED	Connection refused
62	3E	ENETDOWN	Network is down

Error Code		Mnemonic	Description
Decimal	Hexadecimal		
63	3F	ETXTBSY	Text file busy
64	40	ELOOP	Too many levels of symbolic links
65	41	EHOSTUNREACH	No route to host
66	42	ENOTBLK	Block device required
67	43	EHOSTDOWN	Host is down
68	44	EINPROGRESS	Operation now in progress
69	45	EALREADY	Operation already in progress
70	46	EWOULDBLOCK	Operation would block
71	47	ENOSYS	Function not implemented

Errors for any of the Socket Object calls not covered by CIP general status codes contain a general status code of 0xFF (Object Specific General Error) and an extended status code. The extended status codes are standard error codes returned by common socket API implementations. Additional information can be found on the Internet or in socket API programming guides.

Socket Attributes

You access these attributes by configuring the CIP Generic MSG instruction to get or set the specific attribute.

If You Want To	Use This CIP Generic MSG Type:
Change an attribute value for a socket	SetAttribute 
Get a socket value	GetAttribute 

Some socket attributes apply to all sockets and some apply to specific socket instances:

If You Want Attribute Information For	In the CIP Generic MSG Type, Specify	Refer To
All sockets	0 for the Instance (socket class)	Attributes for All Socket Services on page 146
A specific socket instance	The specific socket instance number for the Instance (the instance number is returned by a CreateSocket or AcceptConnection service)	Attributes for Specific Socket Instances on page 147

Socket Object Class Attributes

The Class attributes apply to the Socket Object, not to specific socket instances. When you get or set a Class attribute, set the instance to 0.

Class Attribute	Name	Data Type	Access	Description
1	Revision	INT	Get	Object revision
2	Max Instance	INT	Get	Largest socket instance number currently created
3	Number of Instances	INT	Get	Number of socket instances currently created
8	Log Enable	DINT	Get Set	Enable or disable logging to the Socket Object Log web page. Each socket service has a corresponding bit. If enabled, requests for that service request are logged. If clear, then requests for that service are not logged. Bit 0: CreateSocket requests Bit 1: OpenConnection requests Bit 2: AcceptConnection requests Bit 3: Read requests Bit 4: Write requests Bit 5: DeleteSocket and DeleteAllSockets requests Bit 6: Get / Set Attribute requests Bit 7: Log all service errors

If you use Get Attributes All to get the class attributes, the response contains all of the above attributes, in the order shown above (total size of 10 bytes). Since there is only one attribute (Log Enable) a Set Attributes All request contains only the Log Enable attribute.

Socket Object Instance Attributes

The Socket Object provides a number of instance attributes that apply to specific socket instances. To get or set an instance attribute, specify a valid instance number.

Instance Attribute	Name	Data Type	Access	Description
1	LocalAddr	Struct SockAddr	Get	Local address for the socket
2	RemoteAddr	Struct SockAddr	Get	Remote address for the socket
3	SendBufSize	DINT	Get Set	Size of the socket send buffer (bytes)
4	RecvBufSize	DINT	Get Set	Size of the socket receive buffer (bytes)
5	TCPKeepAlive	DINT	Get Set	Enable (1) or disable (0) TCP Keep Alive for the socket. Enabled by default.
6	TCPNoDelay	DINT	Get Set	Enable (1) or disable (0) the TCP No Delay behavior. Enabled by default.
7	InactivityTimeout	DINT	Get Set	See above section on Inactivity Timeout.

If you use Get Attributes All to get the instance attributes, the response contains all of the above attributes, in the order shown above (total size of 36 bytes). If you use Set Attributes All, the request must include attributes 3, 4, 5, 6 and 7, in that order (total size of 20 bytes).

Troubleshoot Socket Applications

To help debug and troubleshoot applications, the socket interface provides a set of web pages, located in the web server's Advanced Diagnostics folder.

Web Page	Description
Socket Object Diagnostics	Displays information about on each existing instance: • Instance number • Socket type – client, server, or listen • Local and remote ports and IP addresses • Send and receive buffer sizes • Socket up time and inactivity time • Socket state and last error state
Socket Object Statistics	Displays attribute settings for each instance
Socket Object Log	Displays a log of service requests (maximum of 50 log entries): • Service requests made to the socket object • For each request, shows parameters passed and the result of service (success / failure) You can enable/disable logging for some services by setting Class attribute #8 (Log Enable). See Socket Object Class Attributes on page 158.

Debugging Hints and Tips

Category	Consideration
Web server module	Make sure the web server module has a valid IP address. Also, if you communicate with devices on different subnets, configure the web server with a valid subnet mask and gateway address.
CreateSocket service	Make sure the Destination tag is of type of DINT. After creating the socket with the CreateSocket service, make sure you use the instance number that CreateSocket returns in the subsequent socket services you call.
MSG instruction	Make sure the Source Element is of a type that matches the request parameters for the requested service. Also make sure the Source Length is the correct length for the service parameters. There is a limit to the number of active MSG instructions in a Logix controller. If a MSG instruction is enabled and exceeds the maximum number of active MSG instructions, the MSG instruction receives an error (.ER bit set).
OpenConnection service	Make sure the Source Length includes the size of the Timeout parameter + Address Length parameter + the Length of the address itself.

Category	Consideration
Service Timeout parameter	Make sure the Timeout parameter is sufficient for the service. Also make sure the Timeout parameter is less than the MSG instruction timeout. If the timeout set to 0, the service returns immediately.
TCP protocol	Your program should handle the possibility of loss of TCP connections. A TCP connection is a byte stream with no inherent message boundaries. The application defines how to interpret message boundaries. For example, the application might use a fixed length for all messages. For a variable-length message, the application might use a fixed-length header that contains the length of the remainder of the message. Both ends of the TCP connection must agree on the application protocol that is used.
Ethernet sniffer	An Ethernet sniffer is useful to monitor the messages between the web server and other devices. You can capture network traffic and set up filters to isolate messages between particular devices and particular messages between those devices.

Additional Resources

Consult the following publications for more information.

Topic	Publication Title	Publication No.
MSG intructions	Logix5000 Controllers Common Procedures	1756-PM001
	EtherNet/IP Modules in Logix5000 Control Systems User Manual	ENET-UM001
	Logix5000 Controllers General Instructions Reference Manual	1756-RM001

You can view or download publications at http://www.literature.rockwellautomation.com. To order paper copies of technical documentation, contact your local Rockwell Automation distributor or sales representative.

Notes:

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future. Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at RADocumentComments@ra.rockwell.com

Pub. Title/Type EtherNet/IP Web Server Module

Cat. No. 1756-EWEB Pub. No. ENET-UM527D-EN-P Pub. Date May 2006 Part No. 957988-64

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness 1 2 3	How can we make this publication more useful for you?	
Completeness (all necessary information is provided) 1 2 3	Can we add more information to help you?	
	procedure/step	illustration feature
	example	guideline other
	explanation	definition
Technical Accuracy (all provided information is correct) 1 2 3	Can we be more accurate?	
	text	illustration
Clarity (all provided information is easy to understand) 1 2 3	How can we make things clearer?	
Other Comments	You can add additional comments on the back of this form.	

Your Name _____

Your Title/Function _____

Location/Phone _____

Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at _____

___Yes, please contact me via _____

Return this form to: Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

Rockwell Automation Support

Rockwell Automation provides technical information on the web to assist you in using its products. At http://support.rockwellautomation.com, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect Support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://support.rockwellautomation.com.

Installation Assistance

If you experience a problem with a hardware module within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your module up and running.

United States	1.440.646.3223 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

New Product Satisfaction Return

Rockwell tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning, it may need to be returned.

United States	Contact your distributor. You must provide a Customer Support case number (see phone number above to obtain one) to your distributor in order to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for return procedure.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication ENET-UM527E-EN-P - October 2006 PN 953030-29
Supersedes Publication ENET-UM527D-EN-P - May 2006