



Secure Configuration User Guide



Trademark Notices

Control, DeviceMaster, and PortVision are registered trademarks of Control Corporation.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective owners.

Second Edition, July 30, 2013

Copyright © 2010 - 2013. Control Corporation.

All Rights Reserved.

Control Corporation makes no representations or warranties with regard to the contents of this document or to the suitability of the Control product for any particular purpose. Specifications subject to change without notice. Some software or features may not be available at the time of publication. Contact your reseller for current product information.

Table of Contents

DeviceMaster Security	5
Overview	5
Understanding Security Methods and Terminology	5
TCP and UDP Socket Ports Used by the DeviceMaster	11
DeviceMaster Security Features	12
Security Modes	12
Secure Data Mode and Secure Config Mode Comparison	13
Security Comparison	13
SSH Server	14
SSL Overview	14
SSL Authentication	14
Server Authentication	14
Client Authentication	15
Certificates and Keys	15
SSL Performance	17
SSL Cipher Suites	17
DeviceMaster Supported Cipher Suites	18
SSL Resources	18
Key and Certificate Management	19
Password Authentication	21
Requirements	21
PortVision DX Method	21
Login Authentication	21
Configuring Passwords	23
Telnet Commands	24
Telnet Method	25
Login Authentication	25
Configuring Passwords	26
Telnet Help	27
Web Page Password Access	28
Using PortVision DX	29
Overview	29
PortVision Plus	29
PortVision DX Overview	29
Locating DeviceMasters on the Network	30
Adding a Secure DeviceMaster to PortVision DX	32
Enabling Web Page Security (HTTPS)	35
Configuring Security on a DeviceMaster UP	35
Configuring Security on a DeviceMaster	38
Example 1	42
Example 2	42
Example 3	42
Key and Certificate Management	43

Technical Support.....47

DeviceMaster Security



This subsection provides a basic understanding of the DeviceMaster security options, and the repercussions of setting these options. See the *DeviceMaster Installation and Configuration Guide*, if you need to reset DeviceMaster security options or want to return the DeviceMaster settings to their default values.

This *User Guide* discusses secure web configuration for the following:

- DeviceMaster PRO
- DeviceMaster RTS
- DeviceMaster Serial Hub
- DeviceMaster UP
- DeviceMaster 500

Note: *The remainder of this guide simply refers to DeviceMaster unless the information is specific to a model.*

Overview

The Secure Configuration User Guide provides the following information:

- This section provides background information about DeviceMaster security.
- Use [Password Authentication](#) on Page 21 to set up password authentication and a DeviceMaster password.
- [Using PortVision DX](#) on Page 29 provides information about DeviceMaster security and PortVision DX.
- [Enabling Web Page Security \(HTTPS\)](#) on Page 35 provides the procedures to configure security for DeviceMasters.

Understanding Security Methods and Terminology

The following table provides background information and definitions.

Term or Issue	Explanation
CA (Client Authentication certificate) †	<p>If configured with a CA certificate, the DeviceMaster requires all SSL/TLS clients to present an RSA identity certificate that has been signed by the configured CA certificate. As shipped, the DeviceMaster is not configured with a CA certificate and all SSL/TLS clients are allowed.</p> <p>This uploaded CA certificate that is used to validate a client's identity is sometimes referred to as a <i>trusted root certificate</i>, a <i>trusted authority certificate</i>, or a <i>trusted CA certificate</i>. This CA certificate might be that of a trusted commercial certificate authority or it may be a privately generated certificate that an organization creates internally to provide a mechanism to control access to resources that are protected by the SSL/TLS protocols.</p> <p>See Key and Certificate Management on Page 19 for more information. This section does not discuss the creation of CA Certificates.</p>

Term or Issue	Explanation
Client Authentication	A process using paired keys and identity certificates to prevent unauthorized access to the DeviceMaster. Client authentication is discussed in Client Authentication on Page 15 and Key and Certificate Management on Page 43.
DH Key Pair Used by SSL Servers †	<p>This is a private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of the key pair allows an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking.</p> <p>The DH (Diffie-Hellman) key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming.</p> <p>The most serious limitation of Diffie-Hellman (DH key) in its basic or <i>pure</i> form is the lack of authentication. Communications using Diffie-Hellman all by itself are vulnerable to man in the middle attacks. Ideally, Diffie-Hellman should be used in conjunction with a recognized authentication method such as digital signatures to verify the identities of the users over the public communications medium.</p> <p>See Certificates and Keys on Page 15 and Key and Certificate Management on Page 19 for more information.</p>
<p>† All DeviceMaster units are shipped from the factory with identical configurations. They all have the identical, self-signed, Control Server RSA Certificates, Server RSA Keys, Server DH Keys, and no Client Authentication Certificates. For maximum data and access security, you should configure all DeviceMaster units with custom certificates and keys.</p>	
Digital Certificate	<p>A digital certificate is an electronic <i>credit card</i> that establishes your credentials when doing business or other transactions on the Web. It is issued by a certification authority (CA). It contains your name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real. Some digital certificates conform to a standard, X.509. Digital certificates can be kept in registries so that authenticating users can look up other users' public keys.</p> <p>See Key and Certificate Management on Page 19 for more information.</p>

Term or Issue	Explanation
PKI (public key infrastructure)	<p>A public key infrastructure (PKI) enables users of a basically unsecure public network such as the Internet to securely and privately exchange data and money through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for a digital certificate that can identify an individual or an organization and directory services that can store and, when necessary, revoke the certificates. Although the components of a PKI are generally understood, a number of different vendor approaches and services are emerging. Meanwhile, an Internet standard for PKI is being worked on.</p> <p>The public key infrastructure assumes the use of public key cryptography, which is the most common method on the Internet for authenticating a message sender or encrypting a message. Traditional cryptography has usually involved the creation and sharing of a secret key for the encryption and decryption of messages. This secret or private key system has the significant flaw that if the key is discovered or intercepted by someone else, messages can easily be decrypted. For this reason, public key cryptography and the public key infrastructure is the preferred approach on the Internet. (The private key system is sometimes known as symmetric cryptography and the public key system as asymmetric cryptography.)</p> <p>A public key infrastructure consists of:</p> <ul style="list-style-type: none"> • A certificate authority (CA) that issues and verifies digital certificate. A certificate includes the public key or information about the public key • A registration authority (RA) that acts as the verifier for the certificate authority before a digital certificate is issued to a requestor • One or more directories where the certificates (with their public keys) are held • A certificate management system <p>For more information, see SSL Authentication on Page 14, SSL Performance on Page 17, SSL Cipher Suites on Page 17, and DeviceMaster Supported Cipher Suites on Page 18.</p>

Term or Issue	Explanation
RSA Key Pair†	<p>This is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption. RSA is widely used in electronic commerce protocols, and is believed to be sufficiently secure given sufficiently long keys and the use of up-to-date implementations. The system includes a communications channel coupled to at least one terminal having an encoding device, and to at least one terminal having a decoding device.</p> <ul style="list-style-type: none"> • Public key is a value provided by some designated authority as an encryption key that, combined with a private key derived from the public key, can be used to effectively encrypt messages and digital signatures. • Private Key <ul style="list-style-type: none"> - One half of the <i>key pair</i> used in conjunction with a public key - Both the public and the private keys are needed for encryption / decryption but only the owner of a private key ever needs to know it. Using the RSA system, the private key never needs to be sent across the Internet. - The private key is used to decrypt text that has been encrypted with the public key. <p>Thus, if <i>User A</i> sends <i>User B</i> a message, <i>User A</i> can find out <i>User B's</i> public key (but not <i>User B's</i> private key) from a central administrator and encrypt a message to <i>User B</i> using <i>User B's</i> public key. When <i>User B</i> receives it, <i>User B</i> decrypts it with <i>User B's</i> private key. In addition to encrypting messages (which ensures privacy), <i>User B</i> can authenticate <i>User B</i> to <i>User A</i> (so that <i>User A</i> knows that it is really <i>User B</i> who sent the message) by using <i>User B's</i> private key to encrypt a digital certificate.</p> <p>See Key and Certificate Management on Page 19 for more information.</p>
SSH (Secure Shell)	<p>Secure Shell (SSH) allows data to be exchanged using a secure channel between two networked devices. Replaces telnet which has no security. SSH requires password authentication – even if password is empty.</p> <p>See SSH Server on Page 14 for more information.</p>
SSL (Secure Sockets Layer)	<p>The Secure Sockets Layer (SSL) is the predecessor of (TLS) Transport Layer Security.</p> <p>SSL is a commonly-used protocol for managing the security of a message transmission on the Internet. SSL has recently been succeeded by Transport Layer Security (TLS), which is based on SSL. SSL uses a program layer located between the Internet's Hypertext Transfer Protocol (HTTP) and Transport Control Protocol (TCP) layers.</p> <p>SSL is included as part of both the Microsoft and Netscape browsers and most Web server products. Developed by Netscape, SSL also gained the support of Microsoft and other Internet client/server developers as well and became the de facto standard until evolving into Transport Layer Security.</p> <p>SSL uses the public-and-private key encryption system from RSA, which also includes the use of a digital certificate.</p> <p>See Pages 14 through 18 for detailed information about SSL.</p> <p>Note: <i>Two slightly different SSL protocols are supported by the DeviceMaster: SSLv3 and TLSv1.</i></p>

Term or Issue	Explanation
TLS (Transport Layer Security)	<p>Transport Layer Security (TLS) is a protocol that ensures privacy between communicating applications and their users on the Internet. When a server and client communicate, TLS ensures that no third party may eavesdrop or tamper with any message. TLS is the successor to the Secure Sockets Layer (SSL).</p> <p>TLS and SSL are not interoperable. The TLS protocol does contain a mechanism that allows TLS implementation to back down to SSL 3.0.</p>
Secure Data Mode	<p><i>Secure Data Mode is not supported on the DeviceMaster UP platform.</i></p> <p>TCP connections that carry data to/from the DeviceMaster serial ports are encrypted using SSL or TLS security protocols. See Security Modes on Page 12 and Configuring Security on a DeviceMaster on Page 38 for more information.</p>
Secure Config Mode	<p>Unencrypted access to administrative and diagnostic functions are disabled. See Security Modes on Page 12 and Enabling Web Page Security (HTTPS) on Page 35 for more information for your DeviceMaster model.</p>
Secure Monitor Data Mode via Telnet	<p><i>Secure Monitor Data Mode via Telnet is not supported on the DeviceMaster UP platform.</i></p> <p>Allows monitoring of a single serial port on the DeviceMaster while the port is configured for Secure Data Mode. For more information see, the Enable Monitoring Secure Data via Telnet option on Page 40.</p>
<i>Man in the Middle attack</i>	<p>A man in the middle attack is one in which the attacker intercepts messages in a public key exchange and then retransmits them, substituting his own public key for the requested one, so that the two original parties still appear to be communicating with each other.</p> <p>The attack gets its name from the ball game where two people try to throw a ball directly to each other while one person in between them attempts to catch it. In a man in the middle attack, the intruder uses a program that appears to be the server to the client and appears to be the client to the server. The attack may be used simply to gain access to the message, or enable the attacker to modify the message before retransmitting it.</p>
<i>How Public and Private Key Cryptography Works</i>	<p>In public key cryptography, a public and private key are created simultaneously using the same algorithm (a popular one is known as RSA) by a certificate authority (CA).</p> <p>The private key is given only to the requesting party and the public key is made publicly available (as part of a digital certificate) in a directory that all parties can access.</p> <p>The private key is never shared with anyone or sent across the Internet. You use the private key to decrypt text that has been encrypted with your public key by someone else (who can find out what your public key is from a public directory).</p> <p>Thus, if <i>User A</i> sends <i>User B</i> a message, <i>User A</i> can find out <i>User B</i>'s public key (but not <i>User B</i>'s private key) from a central administrator and encrypt a message to <i>User B</i> using <i>User B</i>'s public key. When <i>User B</i> receives it, <i>User B</i> decrypts it with <i>User B</i>'s private key. In addition to encrypting messages (which ensures privacy), <i>User B</i> can authenticate <i>User B</i> to <i>User A</i> (so <i>User A</i> knows that it is really <i>User B</i> who sent the message) by using <i>User B</i>'s private key to encrypt a digital certificate. When <i>User A</i> receives it, <i>User A</i> can use <i>User B</i>'s public key to decrypt it.</p>

Term or Issue	Explanation
<i>Who Provides the Infrastructure?</i>	<p>A number of products are offered that enable a company or group of companies to implement a PKI. The acceleration of e-commerce and business-to-business commerce over the Internet has increased the demand for PKI solutions. Related ideas are the virtual private network (VPN) and the IP Security (IPsec) standard. Among PKI leaders are:</p> <ul style="list-style-type: none"> • RSA, which has developed the main algorithms used by PKI vendors. • Verisign, which acts as a certificate authority and sells software that allows a company to create its own certificate authorities. • GTE CyberTrust, which provides a PKI implementation methodology and consultation service that it plans to vend to other companies for a fixed price. • Xcert, whose Web Sentry product that checks the revocation status of certificates on a server, using the Online Certificate Status Protocol (OCSP). • Netscape, whose Directory Server product is said to support 50 million objects and process 5,000 queries a second; Secure E-Commerce, which allows a company or extranet manager to manage digital certificates; and Meta-Directory, which can connect all corporate directories into a single directory for security management.
<p>The following topic references are from: http://searchsecurity.techtarget.com/</p> <ul style="list-style-type: none"> • PKI (public key infrastructure) • How Public/Private Key Cryptography Works • Who Provides the Infrastructure • Digital Certificate • DH Key • Man in the Middle attack <p>The RSA Key pair topic reference is from: http://en.wikipedia.org/wiki/RSA</p>	

TCP and UDP Socket Ports Used by the DeviceMaster

Following list is all of the logical TCP and UDP socket ports implemented in DeviceMasters.

Socket Port Number	Description
22 SSH 23 Telnet	TCP Ports 22 (ssh) and 23 (telnet) are used for administrative and diagnostic purposes and aren't required for normal use and are enabled by default and Port 23 may be disabled.
80 HTTP 443 SSL or HTTPS	TCP Ports 80 (http) and 443 (https) are used by the web server for administration and configuration and are enabled by default and cannot be disabled.
102 RFC1006	TCP Port 102 is used for RFC1006 (ISO over TCP) serial port access. Not used for normal NS-Link SocketServer access. The RFC1006 server can be disabled by setting the server port number to -1 and is enabled by default.
161 SNMP	UDP Port 161 is used by the SNMP agent if SNMP is enabled which is the default.
4606	TCP Port 4606 is required if you want to use NS-Link or PortVision DX if you want to update firmware without setting up a TFTP server and this port cannot be disabled.
4607	TCP Port 4607 is only used for diagnostic purposes and isn't required for normal operation and this port cannot be disabled. If SocketServer is to be used, then the user may enable usage of TCP or UDP ports for access to the serial ports. These ports are not enabled by default and are also user configurable to different values. Defaults for TCP would begin at 8000 and for UDP would begin at 7000.
TCP 8000 - 8xxx	Incremented per serial port on the DeviceMaster. For example: A DeviceMaster 16- port would have Ports 8000 through 8015.
UDP 7000 - 7xxx	Incremented per serial port on the DeviceMaster. For example: A DeviceMaster 16- port would have Ports 7000 through 7015.

DeviceMaster Security Features

The following subsections provide information about DeviceMaster security features.

Security Modes

The DeviceMaster supports two security modes.

Security Mode	Description
Secure Data <i>(Not supported in the DeviceMaster UP Platform.)</i>	SSL encryption for serial port data streams for both NS-Link and SocketServer. Secure Data mode: <ul style="list-style-type: none"> • Requires SSL encryption of TCP connections to SocketServer (Ports 8000, 8001, 8002, and so forth). • Disables UDP access to SocketServer. • Disables RFC1006 (ISO-over-TCP) access to SocketServer. • Disables MAC-mode access to serial ports. MAC mode admin and ID commands are still allowed. • Requires SSL encryption of NS-Link TCP connections (Port 4606). Not directly supported by NS-Link drivers for Windows and Linux. The Linux driver has been tested using stunnel, but manual setup is required. • Requires SSH instead of telnet connection to the diagnostic log (TCP Port 4607). • Two values for http READ and WRITE commands: A2: Enable.
Secure Config	Encrypts/authenticates configuration and administration operations (web server, IP settings, load SW, and so forth.). Secure Config mode: <ul style="list-style-type: none"> • Disables MAC mode admin commands except for ID request†. • Disables TCP/IP admin commands except for ID request†. • Disables telnet console access (Port 23)†. • Disables unencrypted http:// access via Port 80. • Disables e-mail notification and SNMP features. • Two values for http READ and WRITE commands: A3: Enable.
† Affects both RedBoot and SocketServer / NS-Link applications.	

Secure Data Mode and Secure Config Mode Comparison

This table provides information that compares **Secure Data** and **Secure Config** modes. **Secure Data Mode** is not supporting by the DeviceMaster UP platform.

Feature	Secure Data	Secure Config	Secure Data/ Secure Config
MAC (admin)	enabled	disabled †	disabled †
MAC (async)	disabled	enabled	disabled
TCP 4606 (admin)	SSL, enabled	clear, disabled †	SSL, disabled †
TCP 4606 (async)	SSL	clear	SSL
UDP	disabled	user-configured	disabled
telnet/RFC2217	user-configured	user-configured	user-configured
RFC1006	disabled	user-configured	disabled
4607 (diag log)	SSH	telnet	SSH
8000 (serial port)	SSL	clear	SSL
console (config)	telnet on Port 23 SSH on Port 22	SSH on Port 22	SSH on Port 22
web	clear on Port 80 SSL on Port 443	SSL on Port 443	SSL on Port 443
SMTP, SNMP	user-configured	disabled	disabled
RedBoot MAC	enabled	disabled †	disabled †
RedBoot 4606	enabled	disabled †	disabled †
RedBoot telnet	user-configured	disabled	disabled

Security Comparison

This table displays addition information about security feature comparisons.

Supported by	Weakest				Strongest	
	0	1	2	3	3	4
	None	Password	Authenti- cation	Secure Config	Secure Data	Key & Certificate
RedBoot	yes	yes	yes	no	yes	no
SocketServer	yes	yes	yes	yes	yes	yes
NS-Link Driver/MAC	yes	yes	yes	no	no	no
NS-Link Driver/IP	yes	yes	yes	yes		
Serial Monitoring	yes	yes	yes	no	yes †	no
TCP to Serial Ports	yes	yes	yes	no	no	no
SSH to Serial Ports	no	no	no	yes	yes	yes
UDP to Serial Ports	yes	yes	yes	disabled	disabled	disabled
Telnet/Port23	yes	yes	yes	disabled	yes †	disabled
SSH Telnet/Port 22	yes	yes	yes	yes	yes	yes
Telnet Port 4607	yes	yes	yes	disabled	yes	yes
SSH (PuTTY) 4607	no	no	no	yes	disabled	disabled
HTTP (Port 80)	yes	yes	yes	disabled	disabled	disabled
HTTPS (Port 443)	no	no	no	yes	yes	yes
Email	yes	yes	yes	disabled	disabled	disabled

	Weakest			Strongest		
SNMP	yes	yes	yes	disabled	disabled	disabled
RFC1006	yes	yes	yes	disabled	disabled	disabled

† *Enable Monitoring Secure Data via Telnet must be enabled. SSH does not support port monitoring. You can set the **securemon enable** option.*

admin commands are disabled except for read-only ID command required by NS-Link to identify the device.

The intention is to allow NS-Link to operate through an SSL connection to Port 4606 while is in **Secure Data Mode**, and to allow NS-Link to operate through a MAC connection with **Secure Config Mode** enabled and **Secure Data Mode** disabled.

SSH Server

The DeviceMaster SSH server has the following characteristics:

- Requires password authentication – even if password is empty.
- Enabled/disabled along with telnet access independently of **Secure Data** and **Secure Config Modes**.
- The DeviceMaster uses third-party MatrixSSH library from PeerSec Networks: <http://www.peersec.com/>.

SSL Overview

DeviceMaster SSL provides the following features:

- Provides both encryption and authentication.
 - Encryption prevents a third-party eavesdropper from viewing data that is being transferred.
 - Authentication allows both the client (that is, web browser) and server (that is, DeviceMaster) to ensure that only desired parties are allowed to establish connections. This prevents both unauthorized access and *man-in-the-middle* attacks on the communications channel.
- Two slightly different SSL protocols are supported by the DeviceMaster, SSLv3 and TLSv1.
- The DeviceMaster uses third-party MatrixSSL library from PeerSec Networks: <http://www.peersec.com/matrixssl.html>.

SSL Authentication

DeviceMaster SSL authentication has the following features:

- Authentication means being able to verify the identity of the party at the other end of a communications channel. A username/password is a common example of authentication.
- SSL/TLS protocols allow authentication using either RSA certificates or DSS certificates. DeviceMaster supports only RSA certificates.
- Each party (client and server) can present an ID certificate to the other.
- Each ID certificate is signed by another *authority* certificate or key.
- Each party can then verify the validity of the other's ID certificate by verifying that it was signed by a trusted authority. This verification requires that each party have access to the certificate/key that was used to sign the other party's ID certificate.

Server Authentication

Server Authentication is the mechanism by which the DeviceMaster proves its identity.

- The DeviceMaster (generally an SSL server) can be configured by uploading an ID certificate that is to be presented to clients when they connect to the DeviceMaster.

- The private key used to sign the certificate must also be uploaded to the DeviceMaster.
Note: Possession of that private key will allow eavesdroppers to decrypt all traffic to and from the DeviceMaster.
- The corresponding public key can be used to verify the ID certificate but not to decrypt traffic.
- All DeviceMaster are shipped from the factory with identical self-signed ID certificates and private keys. This means that somebody could (with a little effort) extract the factory default private key from the DeviceMaster firmware and use that private key to eavesdrop on traffic to/from any other DeviceMaster that is being used with the default private key.
- The public/private key pairs and the ID certificates can be generated using **openssl** command-line tools.
- If the server authentication certificate in the DeviceMaster is not signed by an authority known to the client (as shipped, they are not), then interactive SSL clients such as web browsers will generally warn the user.
- If the name in server authentication certificate does not match the *hostname* that was used to access the server, then interactive SSL clients such as web browsers will generally warn the user.

Client Authentication

Client Authentication is the mechanism by which the DeviceMaster verifies the identity of clients (that is, web browsers and so forth).

- Clients can generally be configured to accept a particular unknown server certificate so that the user is not subsequently warned.
- The DeviceMaster (generally an SSL server) can be configured by uploading a trusted *authority* certificate that will be used to verify the ID certificates presented to the DeviceMaster by SSL clients. This allows you to restrict access to the DeviceMaster to a limited set of clients which have been configured with corresponding ID certificates.
- DeviceMaster units will be shipped without an authority certificate and will not require clients to present ID certificates. This allows any and all SSL clients to connect to the DeviceMaster.

Certificates and Keys

To control access to the DeviceMaster's SSL/TLS protected resources you should create your own custom CA certificate and then configure authorized client applications with identity certificates signed by the custom CA certificate.

This uploaded CA certificate that is used to validate a client's identity is sometimes referred to as a *trusted root certificate*, a *trusted authority certificate*, or a *trusted CA certificate*. This CA certificate might be that of a trusted commercial certificate authority or it may be a privately generated certificate that an organization creates internally to provide a mechanism to control access to resources that are protected by the SSL/TLS protocols.

The following is a list that contains additional information about certificates and keys:

- By default, the DeviceMaster is shipped without a CA (Certificate Authority) and therefore allowing connections from any SSL/TLS client. If desired, controlled access to SSL/TLS protected features can be configured by uploading a client authentication certificate to the DeviceMaster.
- Certificates can be obtained from commercial certificate authorities (VeriSign, Thawte, Entrust, and so forth.).
- Certificates can be created by users for their own use by using **openssl** command line tools or other applications.
- Certificates and keys to be uploaded to the DeviceMaster must be in the **.DER** binary file format, not in the **.PEM** ASCII file format. (The **openssl** tools can create files in either format and can convert files back and forth between the two formats.)

- Configuring Certificates and keys are configured by four uploaded files on the bottom *Key and Certificate Management* portion of the *Edit Security Configuration* web page:

- **RSA Key Pair used by SSL and SSH servers**

This is a private/public key pair that is used for two purposes:

- It is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of this key pair allows an eavesdropper to both decrypt traffic on SSL/TLS connections that use RSA encryption during handshaking.
- It is used to sign the Server RSA Certificate in order to verify that the DeviceMaster is authorized to use the server RSA identity certificate. Possession of the private portion of this key pair allows somebody to pose as the DeviceMaster.

If the Server RSA Key is replaced, a corresponding RSA server certificate must also be generated and uploaded as a matched set or clients are not able to verify the identity certificate.

- **RSA Server Certificate used by SSL servers**

- This is the RSA identity certificate that the DeviceMaster uses during SSL/TLS handshaking to identify itself. It is used most frequently by SSL server code in the DeviceMaster when clients open connections to the DeviceMaster's secure web server or other secure TCP ports. If a DeviceMaster serial port configuration is set up to open (as a client), a TCP connection to another server device, the DeviceMaster also uses this certificate to identify itself as an SSL client if requested by the server.
- In order to function properly, this certificate must be signed using the Server RSA Key. This means that the server RSA certificate and server RSA key must be replaced as a pair.

- **DH Key pair used by SSL servers**

This is a private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages.

Possession of the private portion of the key pair allows an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking.

- **Client Authentication Certificate used by SSL servers**

If configured with a CA certificate, the DeviceMaster requires all SSL/TLS clients to present an RSA identity certificate that has been signed by the configured CA certificate. As shipped, the DeviceMaster is not configured with a CA certificate and all SSL/TLS clients are allowed.

SSL Performance

The DeviceMaster has these SSL performance characteristics:

- Encryption/decryption is a CPU-intensive process, and using encrypted data streams will limit the number of ports that can be maintained at a given serial throughput. For example, the table below shows the number of ports that can be maintained by SocketServer at 100% throughput for various cipher suites and baud rates.

	9600	38400	57600	115200
RC4-MD5	32	16	10	5
RC4-SHA	32	13	9	4
AES128-SHA	28	7	5	2
AES256-SHA	26	7	4	2
DES3-SHA	15	3	2	1

Note: *These throughputs required 100% CPU usage, so other features such as the web server are very unresponsive at the throughputs shown above. To maintain a usable web interface, one would want to stay well below the maximum throughput/port numbers above.*

- The overhead required to set up an SSL connection is also significant. The time required to open a connection to SocketServer varies depending on the public-key encryption scheme used for the initial handshaking. Typical setup times for the three public-key encryption schemes supported by the DeviceMaster are shown below:
 - RSA 0.66 seconds
 - DHE 3.84 seconds
 - DHA 3.28 seconds
- Since there is a certain amount of overhead for each block of data sent/received on an SSL connection, the SocketServer polling rate and size of blocks that are written to the SocketServer also has a noticeable effect on CPU usage. Writing larger blocks of data and a slower SocketServer polling rate will decrease CPU usage and allow somewhat higher throughputs.

SSL Cipher Suites

This subsection provides information about SSL cipher suites.

- An SSL connection uses four different facilities, each of which can use one of several different ciphers or algorithms. A particular combination of four ciphers/algorithms is called a “cipher suite”.
- A Cipher Suite consists of
 - Public Key Encryption Algorithm
 - Used to protect the initial handshaking and connection setup.
 - Typical options are RSA, DH, DHA, DHE, EDH, SRP, PSK
 - DeviceMaster supports RSA, DHA, DHE
 - Authentication Algorithm
 - Used to verify the identities of the two parties to each other.
 - Typical options are RSA, DSA, ECDSA
 - DeviceMaster supports only RSA
 - Stream Cipher
 - Used to encrypt the user-data exchanged between the two parties.
 - Typical options: RC4, DES, 3DES, AES, IDEA, Camellia, NULL
 - DeviceMaster supports RC4, 3DES, AES
 - Message Authentication Code

- hash function (checksum) used to verify that each message frame has not be corrupted or changed while in transit.
- typical options include MD5, SHA, MD2, MD4
- DeviceMaster supports MD5, SHA
- In the design of the SSL/TLS protocols the choices of four of the above are not independent of each other: only certain combinations are defined by the standards. The standard combinations of protocol (SSL or TLS) and cipher suites support by DeviceMaster are shown in the attached table.

DeviceMaster Supported Cipher Suites

The DeviceMaster supports the cipher suites:

Protocol	Public Key	Authentication	Cipher	MAC
SSL	RSA	RSA	3DES	SHA
SSL	RSA	RSA	RC4	SHA
SSL	RSA	RSA	RC4	MD5
SSL	DHE	RSA	3DES	SHA
SSL	DHA	RSA	RC4	MD5
SSL	RSA	RSA	NULL	MD5
SSL	RSA	RSA	NULL	SHA
TLS	RSA	RSA	AES128	SHA
TLS	RSA	RSA	AES256	SHA
TLS	DHE	RSA	AES128	SHA
TLS	DHE	RSA	AES256	SHA
TLS	DHA	RSA	AES128	SHA
TLS	DHA	RSA	AES256	SHA

SSL Resources

You can refer to the following SSL resources for more information:

- Standard reference book is SSL and TLS by Eric Rescorla
- Wikipedia page on SSL/TLS provides a good overview: <http://en.wikipedia.org/wiki/TLS>
- **openssl** contains command-line tools to do the following. More information is available at: <http://www.openssl.org/>
 - Create/examine keys/certificates
 - Act as client or server
- **ssldump** is a -command line tool that displays a human-readable dump of an SSL connection's handshaking and traffic:. More information can be found at: <http://www.rtfm.com/ssldump/>
 - If provided with server's private key, can decrypt data stream
 - Can display decoded data stream in ASCII/hex
 - Can display contents of handshaking packets (including ID certificates)

Key and Certificate Management

Key And Certificate Management is only available in *Edit Security Configuration* web page.

Key and Certificate Management

RSA Key pair used by SSL and SSH servers	factory	<input type="button" value="Set"/>	<input type="button" value="Delete"/>
RSA Server Certificate used by SSL servers	factory	<input type="button" value="Set"/>	<input type="button" value="Delete"/>
DH Key pair used by SSL servers	factory	<input type="button" value="Set"/>	<input type="button" value="Delete"/>
Client Authentication Certificate used by SSL servers	none	<input type="button" value="Set"/>	<input type="button" value="Delete"/>

Key and Certificate Management Options	
RSA Key pair used by SSL and SSH servers	<p>This is a private/public key pair that is used for two purposes:</p> <p>It is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of this key pair allows an eavesdropper to both decrypt traffic on SSL/TLS connections that use RSA encryption during handshaking.</p> <p>It is used to sign the Server RSA Certificate in order to verify that the &dm; is authorized to use the server RSA identity certificate. Possession of the private portion of this key pair allows somebody to pose as the &dm;.</p> <p>If the Server RSA Key is to be replaced, a corresponding RSA identity certificate must also be generated and uploaded or clients are not able to verify the identity certificate.</p>
RSA Server Certificate used by SSL servers	<p>This is the RSA identity certificate that the DeviceMaster uses during SSL/TLS handshaking to identify itself. It is used most frequently by SSL server code in the DeviceMaster when clients open connections to the DeviceMaster's secure web server or other secure TCP ports. If a DeviceMaster serial port configuration is set up to open (as a client) a TCP connection to another server device, the DeviceMaster also uses this certificate to identify itself as an SSL client if requested by the server.</p> <p>In order to function properly, this certificate must be signed using the Server RSA Key. This means that the server RSA certificate and server RSA key must be replaced as a pair.</p>
DH Key pair used by SSL servers	<p>This is a private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages.</p> <p>Note: Possession of the private portion of the key pair allows an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking.</p>

Key and Certificate Management Options	
Client Authentication Certificate used by SSL servers	<p>If configured with a CA certificate, the DeviceMaster requires all SSL/TLS clients to present an RSA identity certificate that has been signed by the configured CA certificate. As shipped, the DeviceMaster is not configured with a CA certificate and all SSL/TLS clients are allowed.</p> <p>See Client Authentication on Page 15 for more detailed information</p>
<ul style="list-style-type: none">• <i>All DeviceMaster units are shipped from the factory with identical configurations. They all have the identical, self-signed, Control Server RSA Certificates, Server RSA Keys, Server DH Keys, and no Client Authentication Certificates.</i>• <i>For maximum data and access security, you should configure all DeviceMaster units with custom certificates and keys.</i>	

Password Authentication

This section discusses two methods of configuring password authentication. Review the [Requirements](#) discussion to select the appropriate methods.

Requirements

The procedures in this section require telnet.

- If you have a Windows operating system without telnet support, you can use the [PortVision DX Method](#) (below). In addition, if you use PortVision DX to configure an authenticated password, PortVision DX is able to locate those DeviceMasters after you configure security using [Enabling Web Page Security \(HTTPS\)](#) on Page 35.
- If you have telnet on your system, you can use [Telnet Method](#) on Page 25 as long as the **Telnet/ssh** option has not been disabled in the *Edit Security Configuration* page.

PortVision DX Method

If you have not done so, install PortVision DX, which is a Windows application. If necessary, you can [download the latest version](#) of PortVision DX or locate the product CD and install that version.

This subsection discusses the following topics:

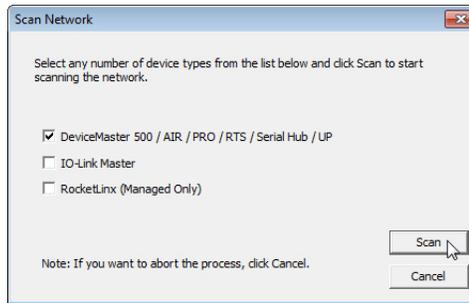
- [Login Authentication](#)
- [Configuring Passwords](#) on Page 23
- [Telnet Commands](#) on Page 24

Login Authentication

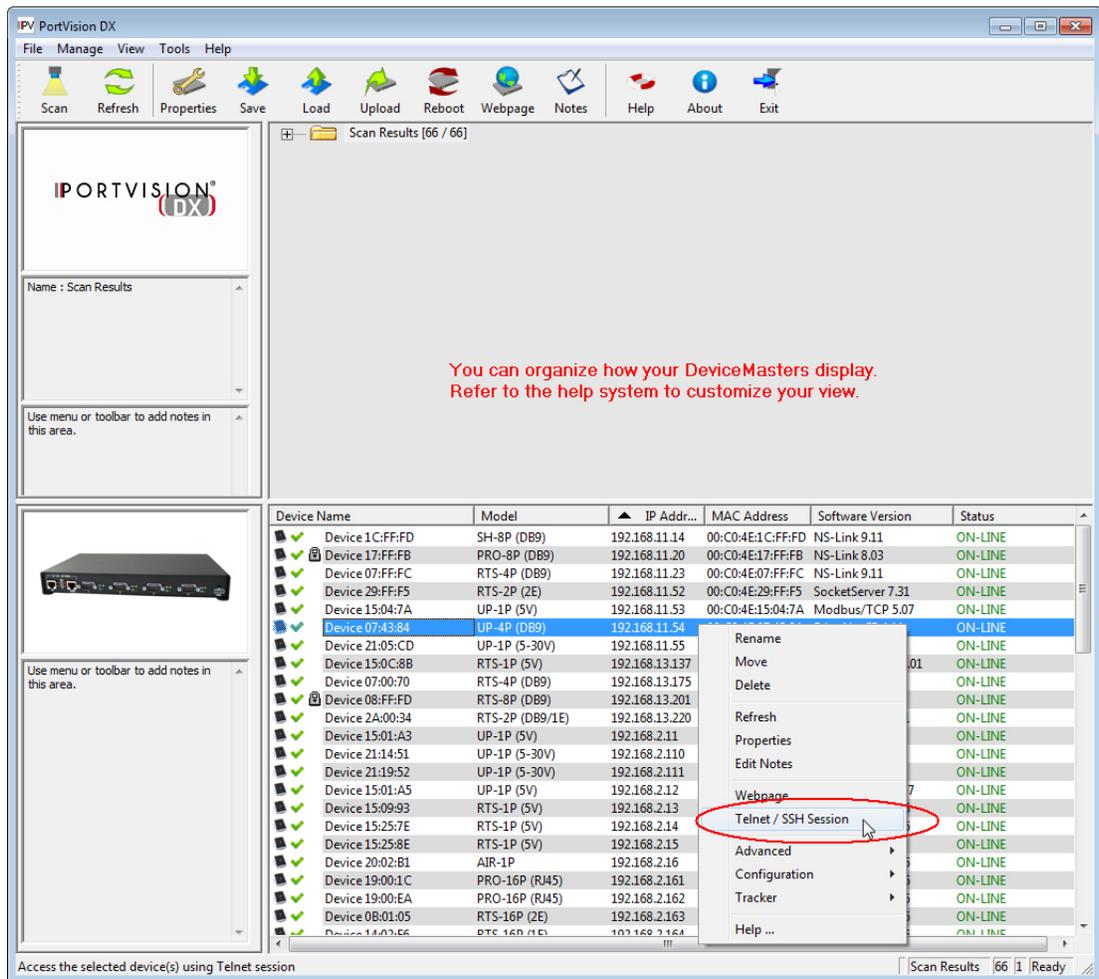
Use the following steps to access a telnet session in PortVision DX so that you can set the log-in authentication.

1. Start PortVision DX.
2. If this is the first time you have started PortVision DX:
 - a. Click the **Scan** button on the *Toolbar* to locate the DeviceMaster for which you want to configure password authentication.
 - a. Click the DeviceMaster option or other appropriate models.

b. Click the **Scan** button.



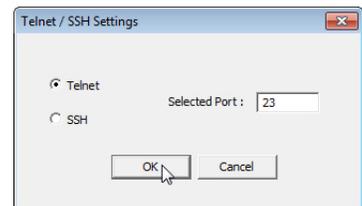
3. Highlight the DeviceMaster in the *Device List* pane (lower) that you want to configure for password authentication and click **Telnet / SSH Session**.



4. Click the **Telnet** option, leave the **Selected Port** number as **23**, and click **Ok**.

5. If the DeviceMaster has a password configured, type the password and press **Enter**.

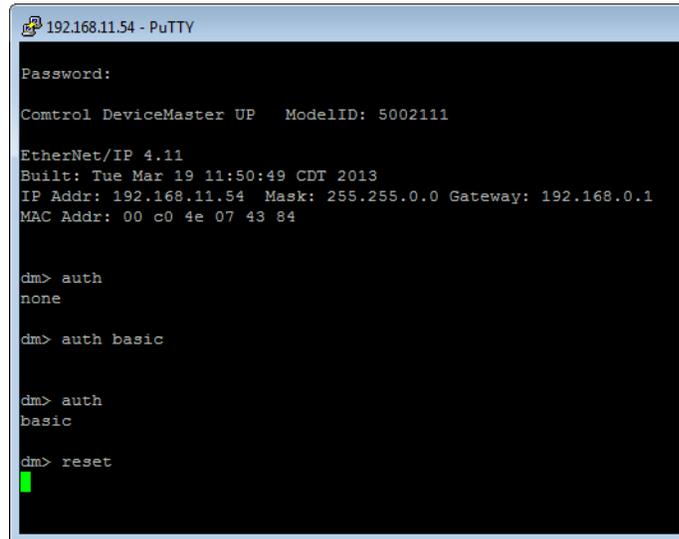
*Note: If no password has been configured, press **Enter**.*



6. Type **auth** and press **Enter** to see the authentication status, **none** indicates that there is no authentication set.
7. Type **auth basic** and press **Enter** to enable enforcing log-in functionality.
8. Type **reset** and press **Enter**.
9. Close the *PuTTY* window.

PortVision DX temporarily displays that DeviceMaster as *OFF-LINE* until the next polling cycle because the DeviceMaster is rebooting.

To disable enforcing log-in functionality, type **auth none**.



```

192.168.11.54 - PuTTY
Password:
Control DeviceMaster UP   ModelID: 5002111
EtherNet/IP 4.11
Built: Tue Mar 19 11:50:49 CDT 2013
IP Addr: 192.168.11.54  Mask: 255.255.0.0 Gateway: 192.168.0.1
MAC Addr: 00 c0 4e 07 43 84

dm> auth
none

dm> auth basic

dm> auth
basic

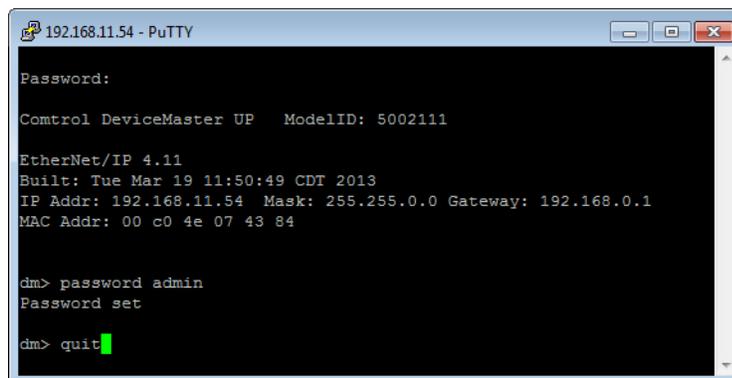
dm> reset

```

Configuring Passwords

Use the following procedure to configure a DeviceMaster password.

1. Highlight the DeviceMaster in the *Device List* pane (lower) that you want to configure for a password and click **Telnet / SSH Session**.
 2. Click the **Telnet** option, leave the **Selected Port** number as **23**, and click **Ok**.
 3. If the DeviceMaster has a password configured, type the password and press **Enter**.
- Note:** *If no password has been configured, press Enter.*
4. Type **password** and the password that you want to set. The example below shows setting the password to **admin**.



```

192.168.11.54 - PuTTY
Password:
Control DeviceMaster UP   ModelID: 5002111
EtherNet/IP 4.11
Built: Tue Mar 19 11:50:49 CDT 2013
IP Addr: 192.168.11.54  Mask: 255.255.0.0 Gateway: 192.168.0.1
MAC Addr: 00 c0 4e 07 43 84

dm> password admin
Password set

dm> quit

```



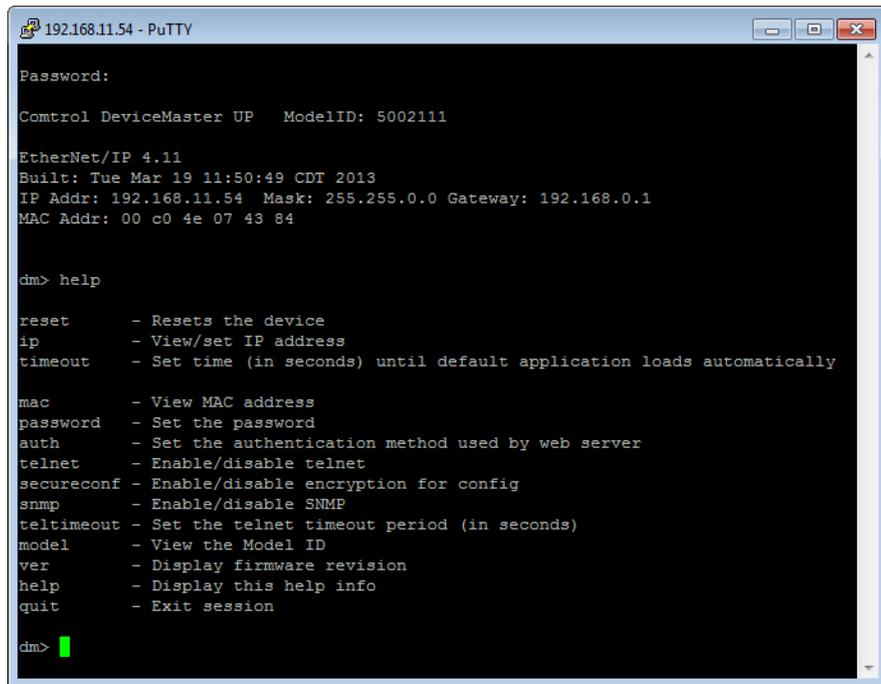
Caution

Make sure that you do not forget the password because after you configure the DeviceMaster with Secure Config Mode, you will not be able to recover the password and will need to return it to the factory to have the default setting loaded.

5. Type **quit** and press **Enter**.

Telnet Commands

To access telnet help, type **help**.



The screenshot shows a PuTTY terminal window titled "192.168.11.54 - PuTTY". The terminal output is as follows:

```
192.168.11.54 - PuTTY
Password:
Control DeviceMaster UP   ModelID: 5002111
EtherNet/IP 4.11
Built: Tue Mar 19 11:50:49 CDT 2013
IP Addr: 192.168.11.54  Mask: 255.255.0.0 Gateway: 192.168.0.1
MAC Addr: 00 c0 4e 07 43 84

dm> help

reset      - Resets the device
ip         - View/set IP address
timeout    - Set time (in seconds) until default application loads automatically

mac        - View MAC address
password   - Set the password
auth       - Set the authentication method used by web server
telnet     - Enable/disable telnet
secureconf - Enable/disable encryption for config
snmp       - Enable/disable SNMP
teltimeout - Set the telnet timeout period (in seconds)
model      - View the Model ID
ver        - Display firmware revision
help       - Display this help info
quit       - Exit session

dm> █
```

Telnet Method

There are several procedures that must be performed for the password access to be enforced.

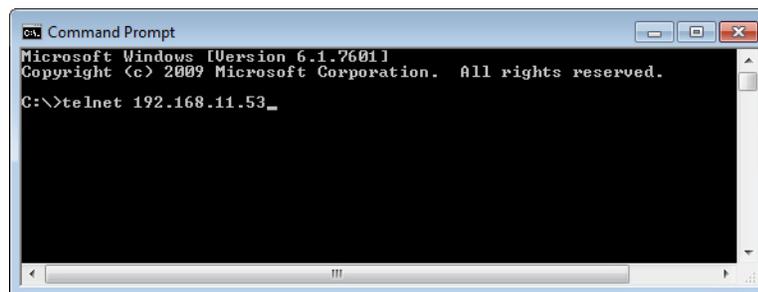
- [Login Authentication](#)
- [Configuring Passwords](#) on Page 26
- [Telnet Help](#) on Page 27

Login Authentication

Before the web page password access method can be enforced, the log-in authentication must be set.

The following steps must be performed for the password access to be enforced:

1. Telnet to the DeviceMaster by typing: **telnet <ip_address>** and press **Enter**.

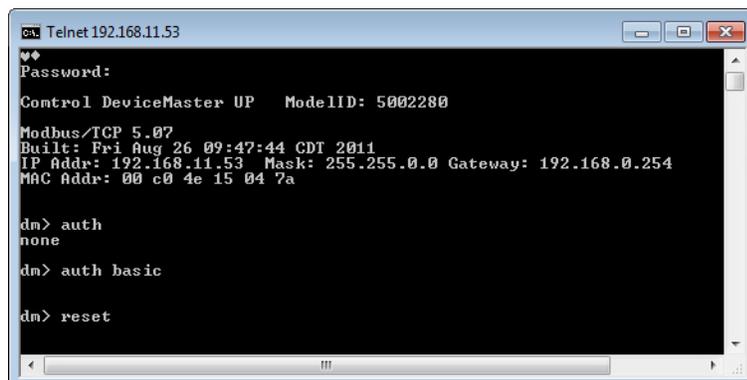


```

C:\>telnet 192.168.11.53_

```

2. If the DeviceMaster has a password configured, type the password and press **Enter**.
Note: If no password has been configured, press Enter.
3. Type **auth** and press **Enter** to see the authentication status, **none** indicates that there is no authentication set.
4. Type **auth basic** and press **Enter** to enable enforcing log-in functionality.



```

Telnet 192.168.11.53
Password:
Control DeviceMaster UP ModelID: 5002280
Modbus/TCP 5.07
Built: Fri Aug 26 09:47:44 CDT 2011
IP Addr: 192.168.11.53 Mask: 255.255.0.0 Gateway: 192.168.0.254
MAC Addr: 00 c0 4e 15 04 7a

dm> auth
none

dm> auth basic

dm> reset

```

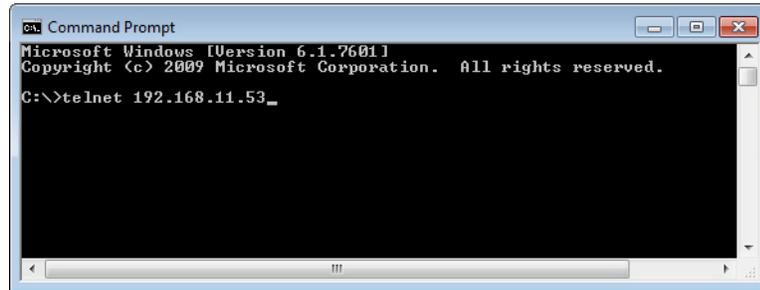
5. Type **reset** and press **Enter**. Allow the system to start-up. By default, this typically takes about 15 seconds.

Note: To disable enforcing log-in functionality, set the authentication to none by typing auth none.

Configuring Passwords

The password can be set or cleared with Telnet. Perform the following procedure to set or clear the password.

1. Telnet to the DeviceMaster by typing **telnet** and the IP address.



2. If the DeviceMaster has a password configured, type the password and press **Enter**.

Note: *If no password has been configured, press **Enter**.*

3. Type **password** and the password that you want to set. The example below shows setting the password to **admin**.



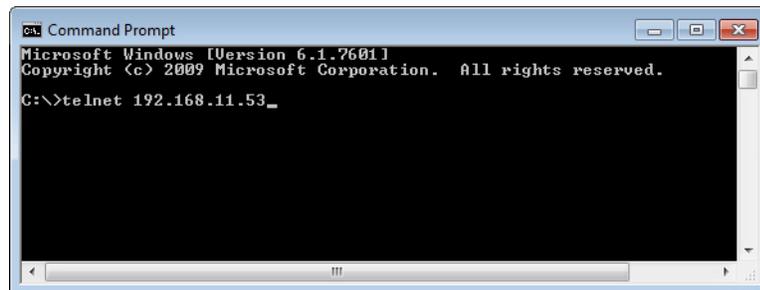
Caution

Make sure that you do not forget the password because after you configure the DeviceMaster with Secure Config Mode, you will not be able to recover the password and will need to return it to the factory to have the default setting loaded.

4. Type **quit** to exit.

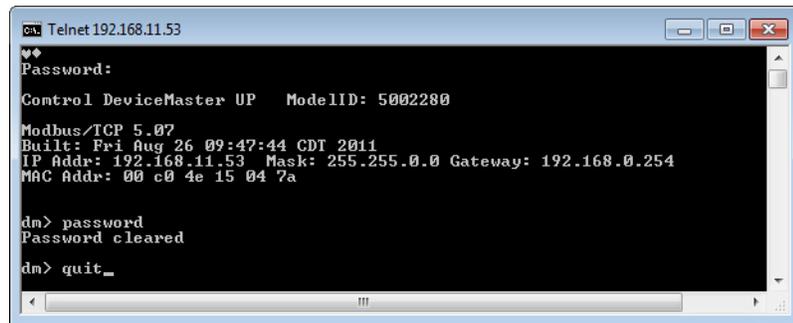
If you want to clear a password, you can do the following:

1. Telnet to the DeviceMaster by typing **telnet** and the IP address.



2. Type the DeviceMaster password and press **Enter**.

3. Type password and press **Enter**, which clears the existing password.



```
Ca: Telnet 192.168.11.53
Password:
Control DeviceMaster UP  ModelID: 5002280
Modbus/TCP 5.07
Built: Fri Aug 26 09:47:44 CDT 2011
IP Addr: 192.168.11.53  Mask: 255.255.0.0  Gateway: 192.168.0.254
MAC Addr: 00 c0 4e 15 04 7a

dm> password
Password cleared

dm> quit_
```

Note: To set a new password, type *password*, the new password, and press **Enter**.

4. Type **quit** to exit.

Telnet Help

To access the Telnet help for the DeviceMaster, type **help**.



```
Ca: Telnet 192.168.11.53
Password:
Control DeviceMaster UP  ModelID: 5002280
Modbus/TCP 5.07
Built: Fri Aug 26 09:47:44 CDT 2011
IP Addr: 192.168.11.53  Mask: 255.255.0.0  Gateway: 192.168.0.254
MAC Addr: 00 c0 4e 15 04 7a

dm> help
reset      - Resets the device
ip         - View/set IP address
timeout    - Set time (in seconds) until default application loads automatically
mac        - View MAC address
password   - Set the password
auth       - Set the authentication method used by web server
telnet     - Enable/disable telnet
secureconf - Enable/disable encryption for config
snmp       - Enable/disable SNMP
teltimeout - Set the telnet timeout period (in seconds)
model      - View the Model ID
ver        - Display firmware revision
help       - Display this help info
quit       - Exit session

dm>
```

Type **quit** to exit.

Web Page Password Access

When the authentication is set to require a password, such as **basic**, you will need to log into each web server session whether you use PortVision DX or a web browser.

Use these steps to log in:

1. Leave the *User name* blank.
2. Type in your password. If there is no password configured, leave the *Password* blank.
3. Click **OK**.

Once logged in, you will have full read/write access to the web pages.



Using PortVision DX

PortVision DX can be used to automatically locate non-secured DeviceMasters. Once located, PortVision DX remembers the DeviceMaster.

PortVision DX is not able to automatically locate a DeviceMaster UP, if **Secure Config Mode** was enabled before scanning with PortVision DX.

If the DeviceMaster is configured with security before PortVision DX has located it, then you can manually add the DeviceMaster to PortVision DX.

Use this section to:

- Locate your DeviceMasters before using the next section to set configure security.
- Add a secure DeviceMaster to PortVision DX that PortVision DX is unable to locate.

Overview

This subsection provides a brief overview of PortVision DX. You can install PortVision DX using the CD shipped with your DeviceMaster or [download](#) the latest version.

PortVision Plus

PortVision DX replaces PortVision Plus. PortVision Plus does not support:

- SocketServer versions v9.00 or above (non-DeviceMaster UP models)
- NS-Link device driver v10.xx or above (non-DeviceMaster UP models)

PortVision DX Overview

PortVision DX automatically detects Control Ethernet-attached products physically attached to the local network segment so that you can configure the network address, upload firmware, and manage the following products:

- DeviceMaster family
 - DeviceMaster PRO
 - DeviceMaster RTS
 - DeviceMaster Serial Hub
 - DeviceMaster UP
 - DeviceMaster 500
- IO-Link Master
- RocketLinx switches

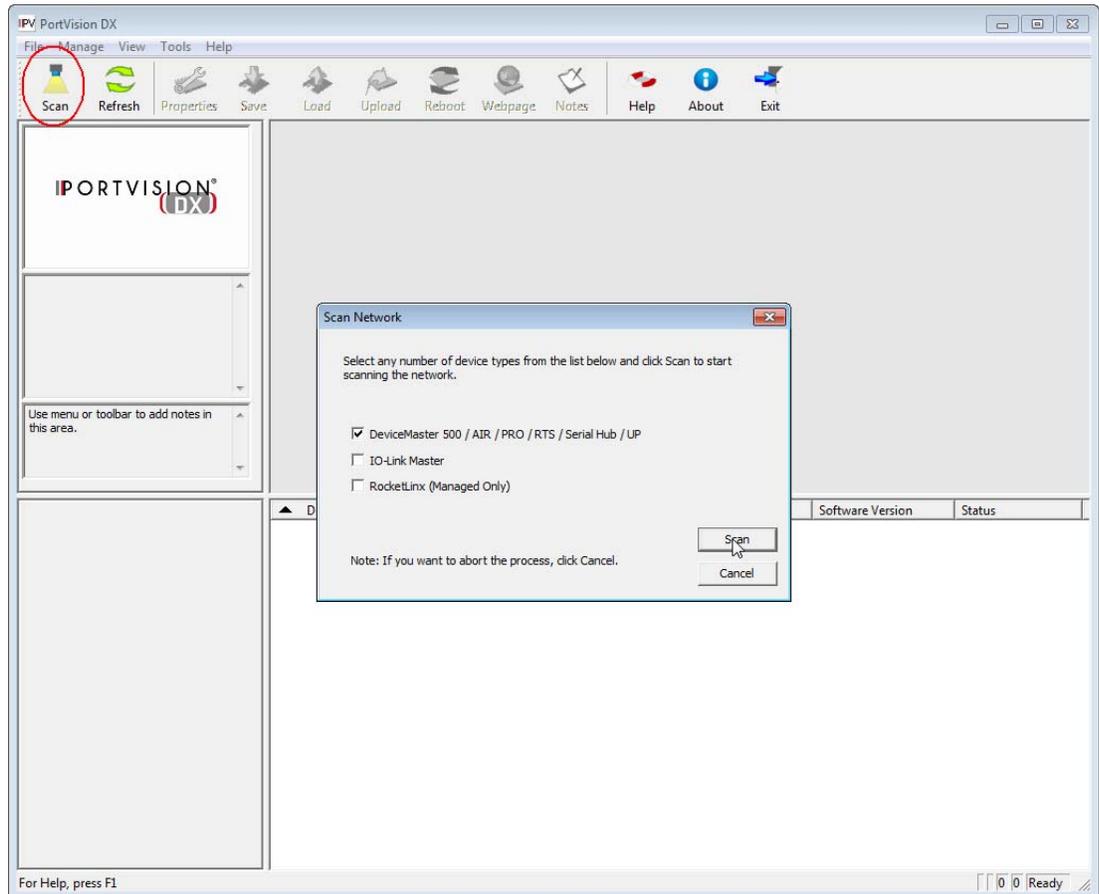
In addition to identifying Control Ethernet-attached products, you can use PortVision DX to display any third-party switch and hardware that may be connected directly to those devices. All non-Control products and unmanaged RocketLinx switches are treated as non-intelligent devices and have limited feature support. For example, you cannot configure or update firmware on a third-party switch.

Locating DeviceMasters on the Network

PortVision DX automatically locates non-secured DeviceMaster when you scan the network for new devices.

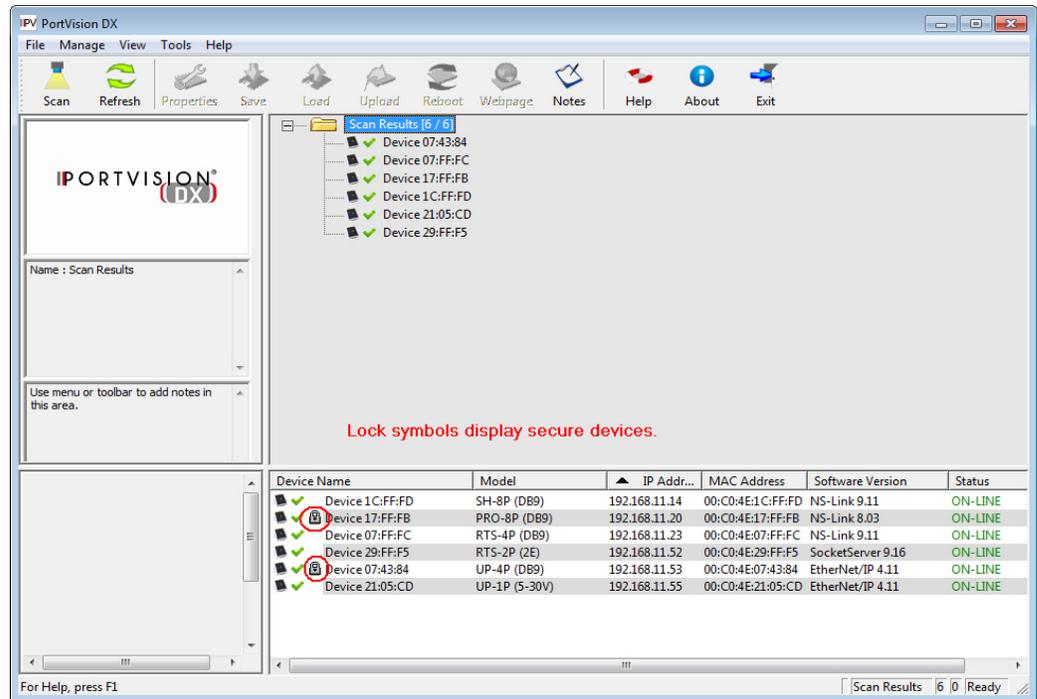
1. Click the **Scan** button in the *Toolbar* and then select the device types for which you want to locate.

Note: *Optionally, you can locate all Control Ethernet-attached products.*



2. PortVision DX displays DeviceMasters (or other Control Ethernet-attached products) located on the network.
 - DeviceMasters with a lock symbol have been configured for security, if PortVision DX located the DeviceMaster before security was configured, it keeps the DeviceMaster UP in its device list.
 - If you configure security before using PortVision DX to locate that DeviceMaster, PortVision DX is unable to locate it on the network.

You can add that DeviceMaster to PortVision DX using the IP or MAC address. See [Adding a Secure DeviceMaster to PortVision DX](#) on Page 32 if you want to display that DeviceMaster in PortVision DX.



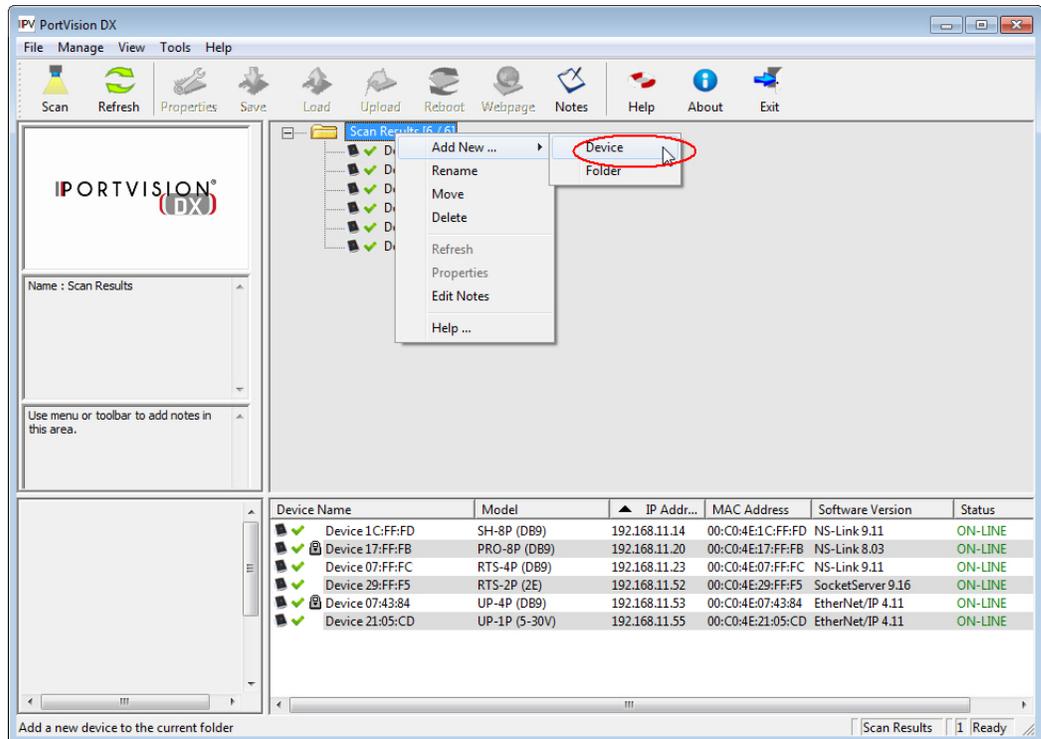
3. Go to [Enabling Web Page Security \(HTTPS\)](#) on Page 35 if you want to configure security.

Adding a Secure DeviceMaster to PortVision DX

If PortVision DX had not located the DeviceMaster before security was enforced, PortVision DX is unable to locate the DeviceMaster.

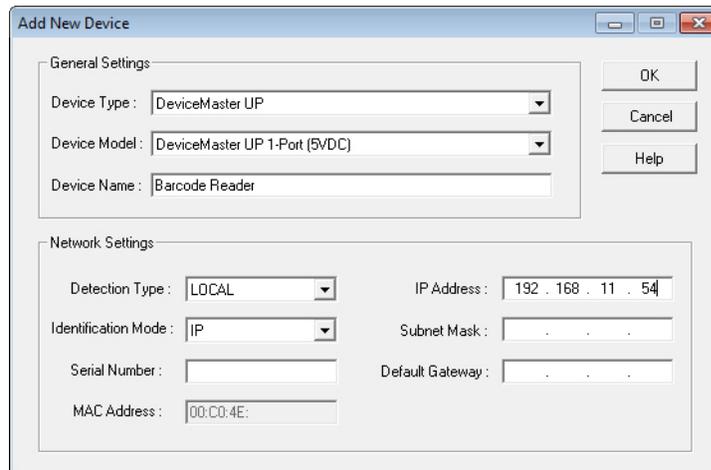
The DeviceMaster will need to be added to the list by using the **Add New Device** option using the following procedure.

1. Right-click the folder for which you want to add the DeviceMaster.
2. Click the **Add New...Device** option.

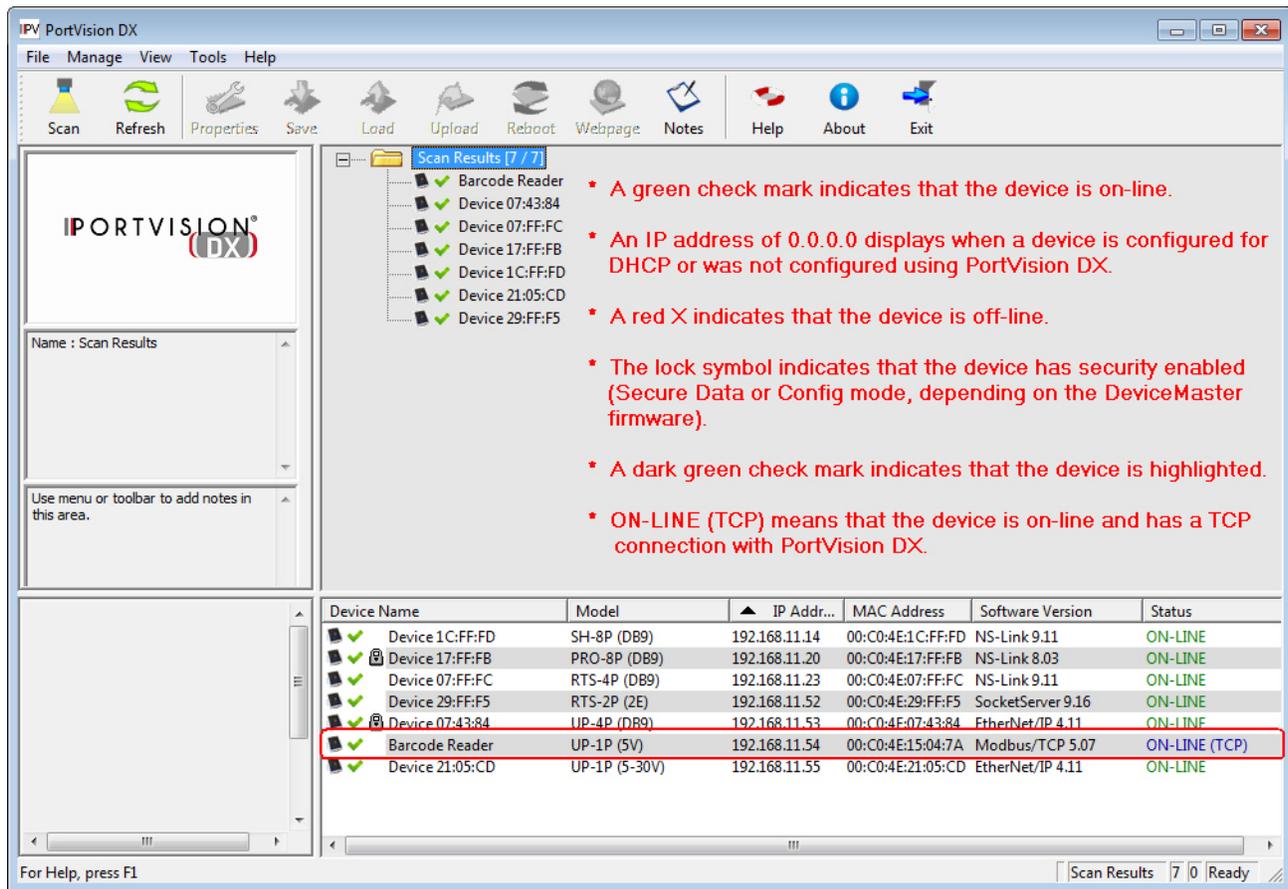


3. Select the appropriate product family from the **Device Type** drop list.
4. Select the specific model type from the **Device Model** drop list.
5. Optionally, provide a **Device Name** that to display in PortVision DX.
6. Select the appropriate **Detection Type** (Local or Remote).
 - You have these options if you select **Local**:
 - If you select **IP** in the **Identification Mode** drop list, you will need to enter the DeviceMaster IP address.
 - If you select **MAC** in the **Identification Mode** drop list, you will need to enter the DeviceMaster MAC address.
 - If you select **Remote**, you will need to enter the DeviceMaster IP address.
7. Optionally, enter the DeviceMaster serial number.

8. Click **Ok** when you have completed the *Add New Device* window.



When you return to the main screen, the DeviceMaster displays in the *Device List* pane (lower).



Enabling Web Page Security (HTTPS)

HTTPS configuration is available but it is up to you to determine whether you want to implement security.

The default settings are:

- Both HTTP (non-secure/unencrypted) and HTTPS (secure/encrypted) configurations are enabled
- Telnet/ssh are enabled
- SNMP is disabled

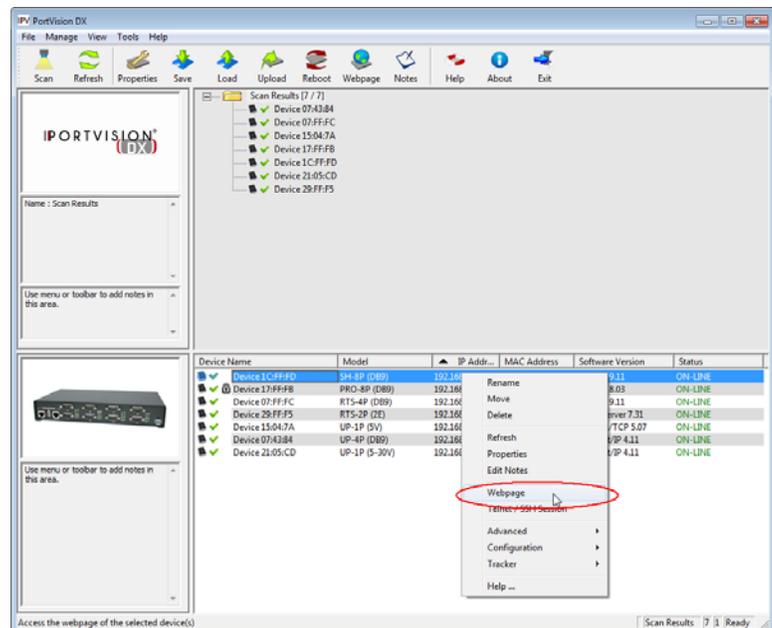
It is up to you to determine whether or not to disable the unencrypted HTTP configuration access. on your DeviceMaster UP (below) or other DeviceMaster models ([Configuring Security on a DeviceMaster](#) on Page 38).

Configuring Security on a DeviceMaster UP

DeviceMaster UP embedded web pages are used to configure DeviceMaster UP security. Enabling security on the DeviceMaster UP disables non-secure configuration functionality.

Use the following procedure to enable DeviceMaster UP security.

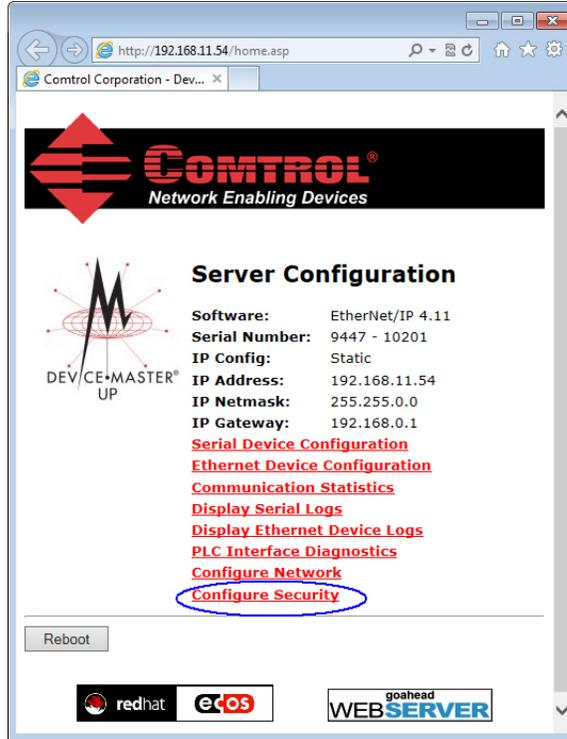
1. Open the DeviceMaster UP *Server Configuration* page using one of these methods:
 - **PortVision DX**
 - Start PortVision DX and click the **Scan** button if the DeviceMaster UP is not displayed.
 - Right-click the DeviceMaster UP in the *Device List* pane (lower) that you want to configure, and then click **Webpage**.



- **Web browser**

Open a web browser and enter the IP address of the DeviceMaster UP for which you want to configure security.

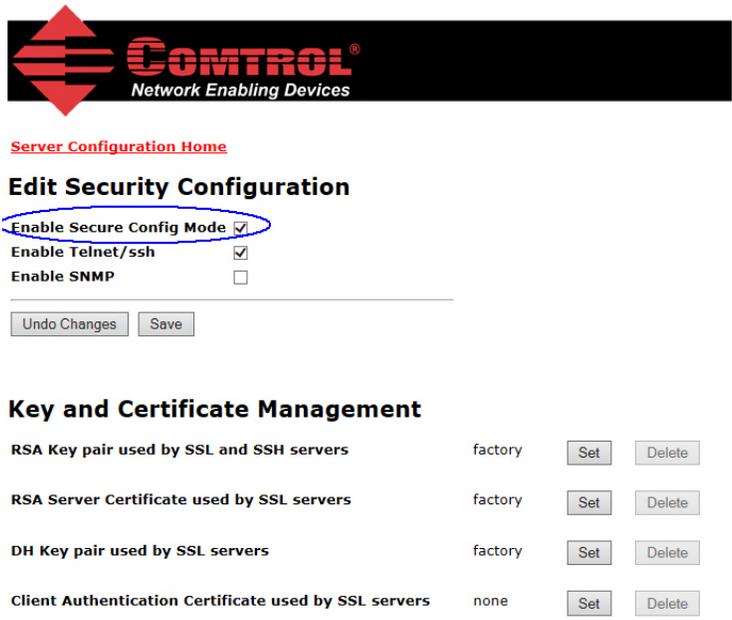
2. Click the **Configure Security** link on the *Server Configuration* page.



Note: This image shows the EtherNet/IP Server Configuration page, which is similar to other DeviceMaster UP protocol Server Configuration pages.

3. Click **Enable Secure Config Mode** to provide this level of security, which disables the following features:

- Telnet access to administrative and diagnostic functions is disabled. If enabled, SSH log ins are still allowed.
- Unencrypted access to the web server via port 80 (<http://> URLs) is disabled.



- Encrypted access to the web server via port 443 (https:// URLs) is still allowed.
 - Administrative commands that change configuration or operating state and are received using the Control proprietary TCP driver protocol on TCP port 4606 are ignored.
 - Administrative commands that change configuration or operating state and are received using the Control MAC mode proprietary Ethernet protocol number 0x11FE are ignored.
4. Unless you want to disable Telnet/ssh on the DeviceMaster UP, leave the **Enable Telnet/ssh.** option enabled.
Note: If you disable this option, PortVision DX cannot communicate through the Telnet/SSH option.
 5. If you want to enable SNMP, click the **Enable SNMP** option.
 6. Click the **Save** button.
 7. Click the **Reboot** button to implement the security changes to the DeviceMaster UP.
 8. Optionally, go to [Key and Certificate Management](#) on Page 43 to configure keys or certificates.



Configuration Updated

Changes to security configuration will not take effect until DeviceMaster unit is rebooted.

Configuring Security on a DeviceMaster

DeviceMaster embedded web pages are used to configure DeviceMaster security. Enabling security on the DeviceMaster disables non-secure configuration functionality.

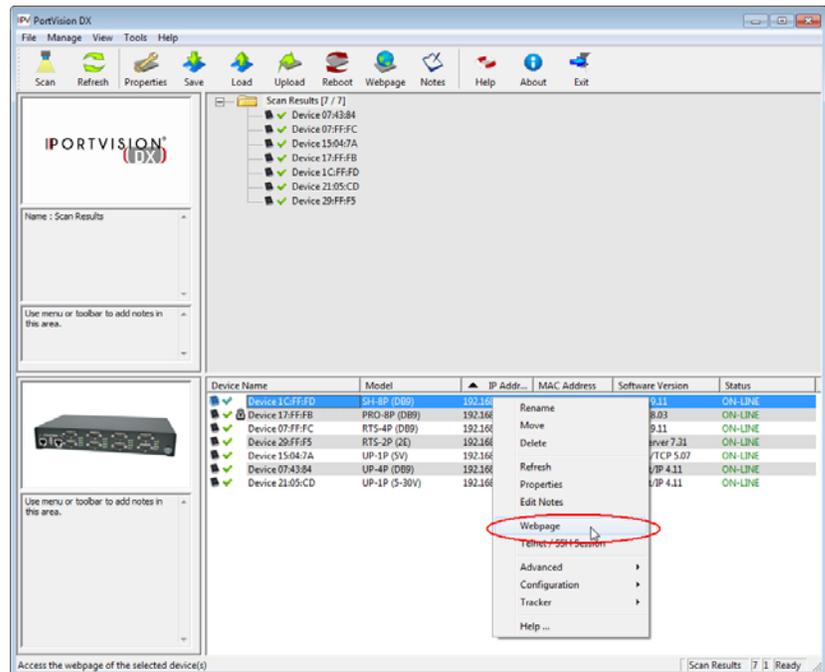
This subsection discusses configuring security with the SocketServer or NS-Link web page for these DeviceMaster models:

- DeviceMaster PRO
- DeviceMaster RTS
- DeviceMaster Serial Hub
- DeviceMaster 500

Note: *The DeviceMaster is loaded with SocketServer from the factory but displays as NS-Link if a device driver is installed and configured.*

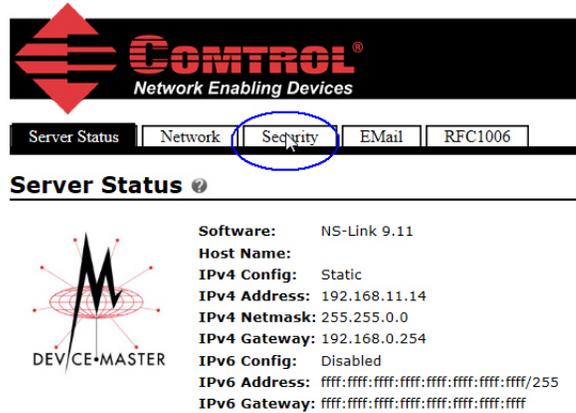
Use the following procedure to enable DeviceMaster security.

1. Open the DeviceMaster *Server Configuration* (or *Server Status*, depending on the DeviceMaster firmware) page using one of these methods:
 - **PortVision DX**
 - Start PortVision DX and click the **Scan** button if the DeviceMaster is not displayed.
 - Right-click the DeviceMaster in the *Device List* pane (lower) that you want to configure, and then click **Webpage**.



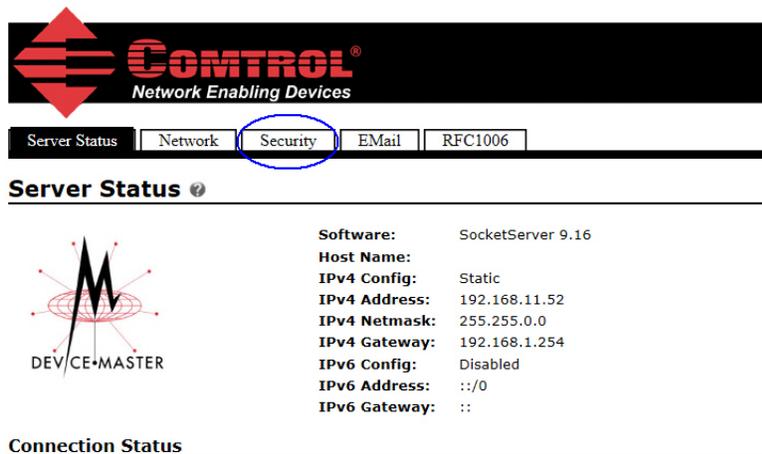
2. Click the **Security** tab.

Note: If the DeviceMaster a SocketServer version before v9.00, the home page is called Server Configuration and you can click the Security link to access the Edit Security Configuration page..



Note: The image above shows the Server Status page for a DeviceMaster loaded with SocketServer v9.11 after an NS-Link device driver was installed and configured.

The image below shows the Server Status page for a DeviceMaster loaded with SocketServer v9.16.



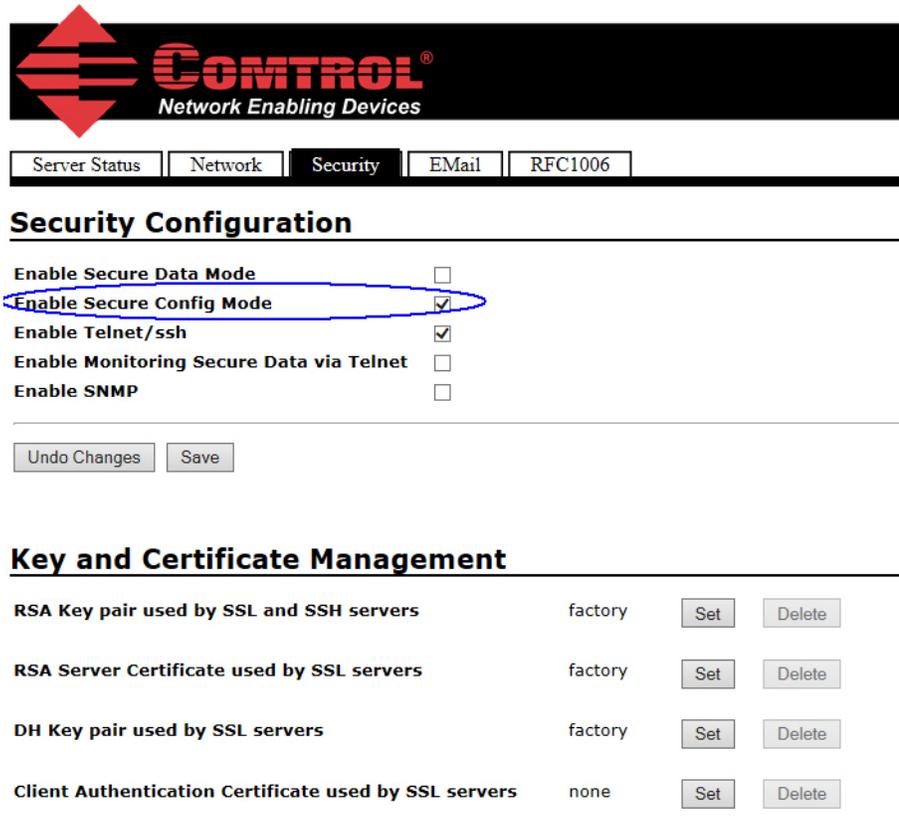
[Port 1](#) [Port 2](#)

3. Optionally, click **Enable Secure Data Mode**.

If **Secure Data Mode** is enabled, TCP connections that carry data to/from the serial ports are encrypted using SSL or TLS security protocols. This includes the following:

- TCP connections to the per-serial-port TCP ports (default is 8000, 8001, 8002, ...) are encrypted using SSL/TLS.
- TCP connections to TCP port 4606 on which the DeviceMaster implements the Control proprietary serial driver protocol are encrypted using SSL/TLS.
- Since SSL/TLS can not be used for either UDP data streams or for the Control proprietary MAC mode Ethernet driver protocol, both UDP and MAC mode serial data transport features are disabled.
- In order to minimize possible security problems e-mail and RFC1006 features are also disabled in **Secure Data Mode**.

In addition to encrypting the data streams, it is possible to configure the DeviceMaster so that only authorized client applications can connect using SSL/TLS. See the [Client Authentication](#) section for details.



4. Click **Enable Secure Config Mode**.

If **Secure Config Mode** is enabled, unencrypted access to administrative and diagnostic functions is disabled. **Secure Config Mode** changes DeviceMaster behavior as follows:

- Telnet access to administrative and diagnostic functions is disabled. SSH access is still allowed.
 - Unencrypted access to the web server via port 80 (http:// URLs) is disabled.
 - Encrypted access to the web server via port 443 (https:// URLs) is allowed.
 - Administrative commands that change configuration or operating state which are received using the Control proprietary TCP driver protocol on TCP port 4606 are ignored.
 - Administrative commands that change configuration or operating state that are received using the Control MAC mode proprietary Ethernet protocol number 0x11FE are ignored.
5. If desired, disable Telnet/ssh. This option enables or disables the telnet security feature after you click **Save** and the DeviceMaster has been rebooted. *This option is enabled by default.*
- If this option is disabled, PortVision DX cannot access the Telnet/SSH option.
6. If desired, click **Enable Monitoring Secure Data via telnet**.
- When checked, **Enable Telnet/ssh** allows the monitor command to be used while **Secure Data Mode** is enabled. When unchecked, the monitor command can only be used if **Secure Data Mode** is not enabled. You must click **Save** and reboot the

DeviceMaster for the change to go into affect. This option is disabled by default.

The **Enable Monitoring Secure Data via Telnet** feature allows you to monitor serial data being sent/received on a serial port (either via NS-Link or SocketServer). The monitoring is done by telnetting to the DeviceMaster and using the following commands:

- **monitor [-ac] portnumber**

Display a live hex dump of Tx/Rx data for the specified serial port. You can only monitor one port at a time. The live dump continues until the **Enter** key is pressed. See the following detailed description and examples. The data is logged when it is written/read to/from the serial port driver's Tx/Rx buffers -- as such, the relative timing between Rx/Tx bytes is not precise, but it should be sufficient to debug most problems (especially frame-oriented, command/response serial protocols).

Monitoring serial data through a telnet connection does generate extra network traffic and may have small effects on the timing of DeviceMaster operations when large amounts of data are being logged at high baud rates. See [Example 1](#) on Page 42 for more information.

- The **-a** option enables displaying of ASCII representation of data in a column to the right the hex representation. See [Example 2](#) on Page 42.
- The **-c** option enables the use of color instead of < and > to indicate the data flow direction. Tx is green and Rx is red. See [Example 3](#) on Page 42.

- **securemon [enable|disable]**

By default, monitoring of Tx/Rx data when in **Secure Data Mode** is not allowed through telnet (an insecure protocol). This command allows you to override that default when **securemon** is enabled it allows monitoring of secure data via an insecure protocol like telnet.

Currently, because of issues with the DeviceMaster ssh implementation, monitoring serial port data via the ssh command-line interface is not supported. It is expected that it will be supported in the future. Once it is supported, the **securemon** setting does not affect the ability to monitor secure data via ssh (which is always allowed).

7. If desired, click **Enable SNMP**. This option enables or disables the SNMP security feature after you click **Save** and the DeviceMaster has been rebooted. *This option is enabled by default*



Example 1

The following example shows how to monitor output using a loopback plug and a program that repeatedly sends the string abcABC123 to Port 1:

```
dm> monitor 1
Serial monitoring started for port 1 -- press [Enter] to stop.
> 61 62 63 41 42 43 31 32 33
< 61 62 63 41 42 43 31 32 33
> 61 62 63 41 42 43 31 32 33
< 61 62 63 41 42 43 31 32 33
> 61 62 63 41 42 43 31 32 33
< 61 62 63 41 42 43 31 32 33
> 61 62 63 41 42 43 31 32 33
< 61 62 63 41 42 43 31 32 33
```

Example 2

The following example shows how the -a option enables displaying of ASCII representation of data in a column to the right the hex representation:

```
dm> monitor -a 1
Serial monitoring started for port 1 -- press [Enter] to stop.
> 61 62 63 41 42 43 31 32 33 > abcABC123
< 61 62 63 41 42 43 31 32 33 < abcABC123
> 61 62 63 41 42 43 31 32 33 > abcABC123
< 61 62 63 41 42 43 31 32 33 < abcABC123
> 61 62 63 41 42 43 31 32 33 > abcABC123
< 61 62 63 41 42 43 31 32 33 < abcABC123
> 61 62 63 41 42 43 31 32 33 > abcABC123
< 61 62 63 41 42 43 31 32 33 < abcABC123
> 61 62 63 41 42 43 31 32 33 > abcABC123
< 61 62 63 41 42 43 31 32 33 < abcABC123
> 61 62 63 41 42 43 31 32 33 > abcABC123
< 61 62 63 41 42 43 31 32 33 < abcABC123
```

Example 3

The -c option enables the use of color instead of < and > to indicate the data flow direction. Tx is green and Rx is red.

```
dm> monitor -c 1
Serial monitoring started for port 1 -- press [Enter] to stop.
61 62 63 41 42 43 31 32 33 61 62 63 41 42 43 31
32 33 61 62 63 41 42 43 31 32 33 61 62 63 41 42
43 31 32 33 61 62 63 41 42 43 31 32 33 61 62 63
41 42 43 31 32 33 61 62 63 41 42 43 31 32 33 61
62 63 41 42 43 31 32 33 61 62 63 41 42 43 31 32
33 61 62 63 41 42 43 31 32 33 61 62 63 41 42 43
31 32 33 61 62 63 41 42 43 31 32 33 61 62 63 41
42 43 31 32 33 61 62 63 41 42 43 31 32 33 61 62
63 41 42 43 31 32 33 61 62 63 41 42 43 31 32 33
The -a and -c options can be used together:
dm> monitor -ac 1
Serial monitoring started for port 1 -- press [Enter] to stop.
61 62 63 41 42 43 31 32 33 61 62 63 41 42 43 31 | abcABC123abcABC1
32 33 61 62 63 41 42 43 31 32 33 61 62 63 41 42 | 23abcABC123abcAB
43 31 32 33 61 62 63 41 42 43 31 32 33 61 62 63 | C123abcABC123abc
41 42 43 31 32 33 61 62 63 41 42 43 31 32 33 61 | ABC123abcABC123a
62 63 41 42 43 31 32 33 61 62 63 41 42 43 31 32 | bcABC123abcABC12
33 61 62 63 41 42 43 31 32 33 61 62 63 41 42 43 | 3abcABC123abcABC
```

31 32 33 61 62 63 41 42 43 31 32 33 61 62 63 41		123abcABC123abcA
42 43 31 32 33 61 62 63 41 42 43 31 32 33 61 62		BC123abcABC123ab
63 41 42 43 31 32 33 61 62 63 41 42 43 31 32 33		cABC123abcABC123

Key and Certificate Management

Use the following procedure to set up security key and certificates for the DeviceMaster.

For detailed information about these options, see the table in *Key and Certificate Management* on Page 19. For information about security keys and certificates, see *DeviceMaster Security* on Page 5.

1. If required, click **Set** on the *Edit Security Configuration* page to configure **RSA key pair used by SSL and SSH servers.**

The RSA Key Pair is used to sign the Server RSA Certificate. This verifies that the DeviceMaster UP is authorized to use the server RSA identity certificate. If the Server RSA Key is to be replaced, a corresponding RSA identity certificate must also be generated and uploaded. If this is not done, clients are not able to verify the identity certificate.

Note: Possession of the private portion of this key pair could allow someone to pose as the DeviceMaster UP.

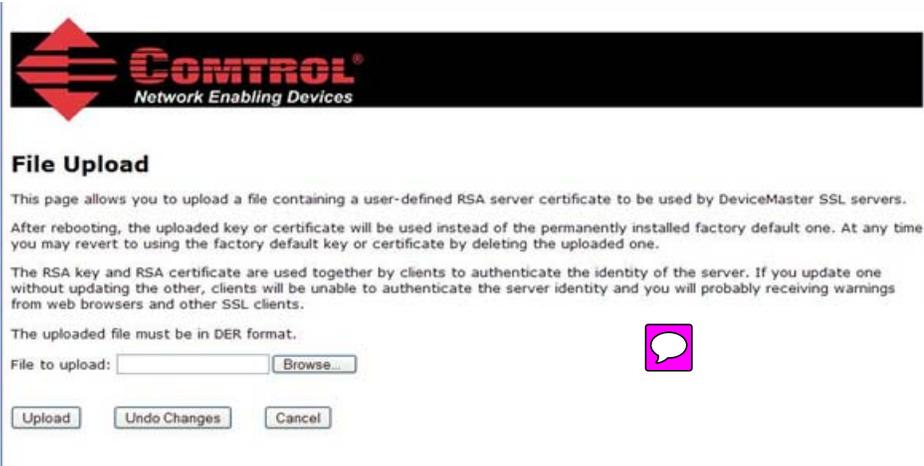
- a. Click **Browse** to locate the server RSA key.
- b. Click **Upload**.

2. If required, click **Set** on the *Edit Security Configuration* page to configure the **RSAServer Certificate used by SSL servers.**

This is the certificate that the DeviceMaster UP uses during SSL/TLS handshaking to identify itself. It is used most frequently by the DeviceMaster UP SSL server firmware when clients open connections to the DeviceMaster UP's secure web server or other secure TCP ports. In order to function

properly, this certificate must be signed using the Server RSA Key. This means that the server RSA certificate and server RSA key must be replaced as a pair.

- a. Click **Browse** to locate the RSA server certificate.
- b. Click **Upload**.

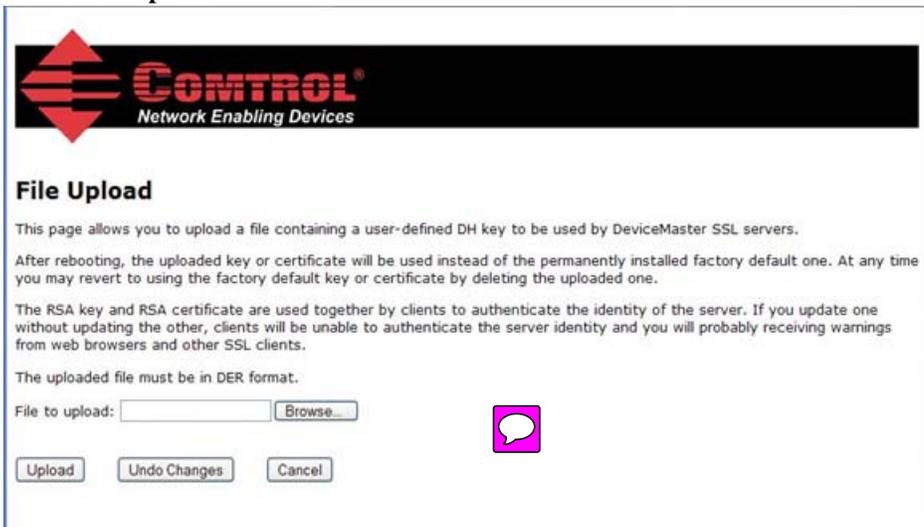


- 3. If required, click **Set** to enter the **DH Key Pair** used by SSL servers on the *Edit Security Configuration* page.

This is the private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages.

Note: Possession of the private portion of the key pair can allow an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking.

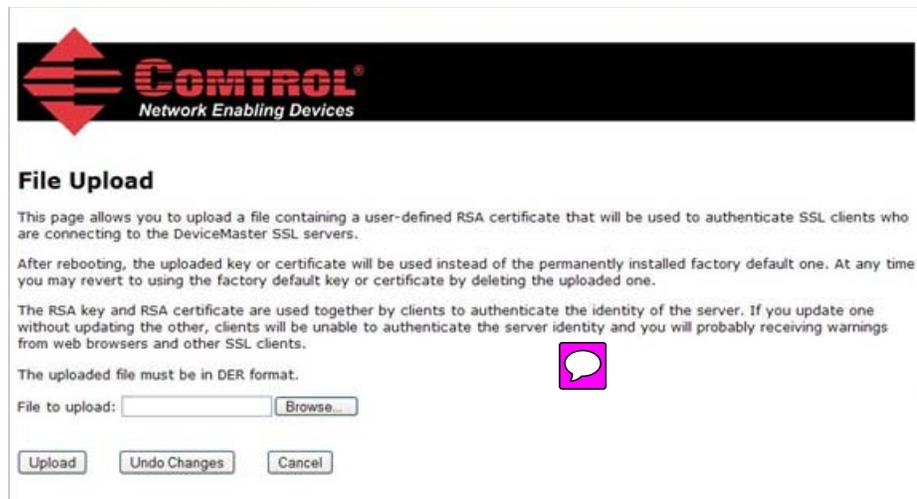
- a. Click **Browse** to locate the private/public key pair.
- b. Click **Upload**.



4. If required, click **Set** on the *Edit Security Configuration* page to upload the **Client Authentication Certificate used by SSL servers**.

If a CA certificate is uploaded, the DeviceMaster UP only allows SSL/TLS connections from client applications that provide to the DeviceMaster UP an identity certificate. This identity certificate must have been signed by the CA certificate that was uploaded to the DeviceMaster UP. The uploaded CA certificate is used to validate a client's identity.

- The uploaded CA certificate is sometimes referred to as a *trusted root certificate*, a *trusted authority certificate*, or a *trusted CA certificate*.
- The uploaded CA certificate might be that of a trusted commercial certificate authority or it may be a privately generated certificate that an organization creates internally to provide a mechanism to control access to resources that are protected by the SSL/TLS protocols.
- To control access to the DeviceMaster UP's SSL/TLS protected resources you should create your own custom CA certificate and then configure authorized client applications with identity certificates signed by the custom CA certificate.



- a. Click **Browse** to locate the Client Authentication Certificate.
 - b. Click **Upload**.
5. After completing the key and certification management, click **Save**.
 6. To allow the changes to become affective, click **Reboot**.



Configuration Updated

Changes to security configuration will not take effect until DeviceMaster unit is rebooted.

Technical Support



You may want to review the troubleshooting procedures in the appropriate document before contacting Technical Support since they will request that you perform, some or all of the procedures before they will be able to help you diagnose your problem.

- [*DeviceMaster Installation and Configuration Guide*](#)
- [*DeviceMaster UP Hardware Installation and Configuration Guide*](#)

If you need technical support, use one of the following methods.

Control Contact Information	
DeviceMaster UP Downloads	ftp://ftp.comtrol.com/html/up_main.htm
DeviceMaster Downloads	ftp://ftp.comtrol.com/html/DM_PRO_RTS_SERIALHUB.htm
Web site	http://www.comtrol.com
FAQs	http://forum.comtrol.com/
Technical Support	http://my.comtrol.com/
Phone	(763) 957-6000

