



# 3-Series<sup>®</sup> Control Systems

Reference Guide

Creston Electronics, Inc.

The original language version of this document is U.S. English.  
All other languages are a translation of the original document.

Crestron product development software is licensed to Crestron dealers and Crestron Service Providers (CSPs) under a limited nonexclusive, nontransferable Software Development Tools License Agreement. Crestron product operating system software is licensed to Crestron dealers, CSPs, and end-users under a separate End-User License Agreement. Both of these Agreements can be found on the Crestron website at [www.crestron.com/legal/software\\_license\\_agreement](http://www.crestron.com/legal/software_license_agreement).

The product warranty can be found at [www.crestron.com/warranty](http://www.crestron.com/warranty).

The specific patents that cover Crestron products are listed online at [www.crestron.com/legal/patents](http://www.crestron.com/legal/patents).

Certain Crestron products contain open source software. For specific information, please visit [www.crestron.com/opensource](http://www.crestron.com/opensource).

Crestron, the Crestron logo, 3-Series, 3-Series Control System, Crestron Studio, Crestron Toolbox, DM NVX, SIMPL+, SmartObjects, VT-Pro e, and XiO Cloud are either trademarks or registered trademarks of Crestron Electronics, Inc. in the United States and/or other countries. Active Directory is either a trademark or a registered trademark of Microsoft Corporation in the United States and/or other countries. Wi-Fi is either a trademark or a registered trademark of Wi-Fi Alliance in the United States and/or other countries. Other trademarks, registered trademarks, and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Crestron disclaims any proprietary interest in the marks and names of others. Crestron is not responsible for errors in typography or photography.

©2023 Crestron Electronics, Inc.

# Contents

<b>Introduction</b> .....	<b>1</b>
<b>Tools and Utilities</b> .....	<b>2</b>
<b>3-Series Architecture</b> .....	<b>3</b>
Memory and Directory Structure .....	3
Flash Memory .....	3
SDRAM (Volatile) .....	4
NVRAM (Nonvolatile) .....	5
Console Commands .....	5
<b>Establish Communications</b> .....	<b>6</b>
USB Connection .....	6
TCP/IP Connection .....	8
Set a Logon Banner .....	9
<b>Time and Date Settings</b> .....	<b>10</b>
<b>Authentication</b> .....	<b>11</b>
Enable Authentication .....	11
Account Recovery .....	11
Factory Restore .....	11
Password Recovery Command .....	12
User and Group Management .....	12
Add Local User .....	12
Delete Local User .....	13
Add Local Group .....	13
Delete Local Group .....	14
Add Active Directory Group .....	14
Remove Active Directory Group .....	14
Add User to Group .....	15
Remove User from Group .....	15
User Group Rights .....	15
Password Management .....	16
Set Password Policy .....	16
Update Local Password .....	17
Reset User Password .....	17
Login Behavior .....	17
Local User Login .....	18
Active Directory Login .....	18
Session Timeout Functions .....	19

Change Session Timeout Duration .....	19
Change Login Failure Count .....	19
Locked User Functions .....	20
Add User to Locked List .....	20
Remove User from Locked List .....	20
Blocked IP Address Functions .....	20
Change Lock out Time .....	20
Add IP Address to Blocked List .....	21
Remove IP Address from Blocked List .....	21
<b>Certificate Management .....</b>	<b>22</b>
Certificate Requirements .....	22
Add a Certificate .....	23
TLS/SSL .....	23
Server Certificates .....	24
Self-Signed Certificates .....	24
CA-Signed Certificates .....	24
Externally-Signed Certificates .....	26
<b>802.1X .....</b>	<b>28</b>
<b>Firmware Updates .....</b>	<b>30</b>
<b>Message Logging .....</b>	<b>31</b>
Persistent Log .....	31
Message Levels .....	32
Message Format .....	32
Remote System Logging .....	33
Audit Logging .....	34
Configure Audit Logging .....	34
View Audit Logs .....	35
Clear Audit Log .....	35
<b>Control Subnet .....</b>	<b>36</b>
Prepare the Control Subnet .....	37
Control Subnet Architecture .....	38
Firewall Rules in Normal Operation .....	39
Firewall Rules in Isolation Mode .....	40
<b>Program Management .....</b>	<b>41</b>
Load Programs to the Control System .....	41
IP Table Configuration .....	41
View and Configure IP Table .....	42
Add Peer Entry .....	42
Remove Peer Entry .....	43

Load IP Table .....	43
Run Multiple Programs .....	44
Device Registration Considerations .....	44
Run Programs from External Storage .....	45
<b>Primary-Secondary Mode .....</b>	<b>46</b>
Definitions .....	46
Cresnet Server .....	46
Ethernet Primary .....	46
Ethernet Secondary .....	47
Primary-Secondary Configuration .....	47
Add Primary Entry .....	47
Remove Primary Entry .....	47
View Primary IP Table .....	47
View Secondary Status .....	47
Response Reject Count for Secondary Connection .....	48
Secondary Connection Timeout .....	48
Functional Behavior .....	49
<b>Auto Update Mechanism .....</b>	<b>51</b>
Configure the Auto Update Mechanism .....	51
Manifest File .....	52
Overview of Manifest Parameters .....	52
Description of Manifest Parameters .....	54
Sample Manifest File .....	57
Manifest File Code Flow .....	59
Results File .....	60
Description of Results File Parameters .....	60
Error Handling .....	62
<b>Connect to XiO Cloud Service .....</b>	<b>63</b>
<b>Appendix A: Restore to Factory Defaults .....</b>	<b>64</b>
<b>Appendix B: Port Forwarding .....</b>	<b>65</b>
<b>Appendix C: CP3-GV Feature Set .....</b>	<b>66</b>

# Introduction

Crestron® 3-Series Control System® automated processors unify disparate technologies in buildings so they can communicate and work together intelligently, which lowers costs and boosts efficiency. Their unique distributed architecture enables multitasking essential to a complete building control solution. Up to 10 programs can operate independently and simultaneously while communicating with each other. Code is organized into a few smaller programs rather than one large one, so programming, troubleshooting, and uploading are much faster and easier.

3-Series® Control Systems offer a wide range of features, including:

- Scalable hardware that supports a broad range of space types and architectures
- One system running multiple programs
- SNMP and BACnet/IP support to seamlessly communicate and integrate with IT, HVAC, BMS, and security systems
- XiO Cloud® service connected
- Full network security protocols, including 802.1X, SSH, TLS, and Active Directory® service

**NOTE:** The features and functions described in this document apply to 3-Series® Control Systems with firmware version 1.600 or newer. Crestron recommends updating to the latest firmware version to ensure the control system receives the most recent features and security patches. For more information, refer to [Firmware Updates on page 30](#).

# Tools and Utilities

Standard IT tools, such as SFTP (secure file transfer protocol) clients and SSH (secure shell) clients, can be used to initiate various control system tasks and functions.

Additionally, the following Crestron tools and utilities may be used to program and configure the control system:

- Crestron Studio® software
- Crestron Toolbox™ software
- SIMPL Debugger tool
- SIMPL software
- SIMPL# Pro software
- VT-Pro e® software

All tools and utilities may be downloaded from [www.crestron.com/Support](http://www.crestron.com/Support). For more information on the features and functions of each tool, refer to its embedded help file.

**NOTE:** Access to software downloads and other files is reserved for Authorized Crestron dealers, Crestron Service Providers (CSPs), and Crestron partners only. New users must register for an account to access certain areas of the Crestron website. For more information on registering, refer to [www.crestron.com/register](http://www.crestron.com/register).

# 3-Series Architecture

The following sections provide an overview of the architecture of a 3-Series control system.

## Memory and Directory Structure

A 3-Series processor has between 256 MB and 1 GB of built-in SDRAM (synchronous DRAM) volatile memory. For more information regarding the memory specifications for each 3-Series control system model, refer to the appropriate product page at [www.crestron.com](http://www.crestron.com).

The file system inside the 3-Series control engine is contained within Flash memory. The 3-Series processor also has 128kB of NVRAM (nonvolatile RAM). NVRAM contains program variables that are retained after the loss of electrical power, while volatile memory is lost.

## Flash Memory

Flash memory for a 3-Series control system contains the following components:

- Operating system (.puf file)
- SIMPL and SIMPL# Pro programs
- SIMPL+® programming modules

The files that reside in flash memory conform to a flat directory structure. The following table details the overall file system as shown over SNTP. The internal file path is also provided where applicable.

### 3-Series Control System Directory Structure

SFTP File Path	Internal File Path	Description
\auditlog	(N/Ap)	Contains the output of the audit log files
\autoupdatelogs	\sys\AutoUpdate\Logos	Contains the output of the autoupdate log files
\cert	\romdisk\user\cert	Contains any user certificates loaded to the control system
\firmware	\romdisk\user\system	Contains firmware files (PUF or ZIP) loaded to the control system
\html	\html	Contains web pages
\nvram	\nvram	The NVRAM legacy directory
\plog	\logs	Contains the output of the persistent log (PLOG) files
\rm, \rm2	\rm, \rm2	A mounting point for external removal media
\programxx	\simpl\programxx	The control system program files (where xx is the program number)

SFTP File Path	Internal File Path	Description
\sshbanner	\sshbanner	Contains a custom SSH banner text file loaded to the control system
\temp	(N/Ap)	Contains any temp files used by the control system
\user	\user	Contains any user-defined files loaded to the control system

The directory structure of a 3-Series control system can reside on the internal flash memory and on optional external memory (SD/SDHC). Programs, data files, and data can be stored on either internal or external memory. The files that reside in the internal flash conform to a flat directory structure, while the external memory system conforms to a FAT32-compatible file system.

**NOTE:** Although file system names are case insensitive, the case is preserved to maintain file checksums.

Observe the following about external media directories:

- The **rm** and **rm2** directories appear only when external removable media is inserted into the control system. To reference files in external memory, prefix "\rm\" or "\rm2\" to any fully qualified path from the computer OS environment.
- When a SIMPL or SIMPL# Pro program is stored in external memory, the files reside in the `\rm[2]\programxx\` directory, where "xx" is the program number.
- When web pages are stored in external memory, the pages reside in the `\rm[2]\html\` directory.
- Storing programs and web pages in external memory gives them precedence over files stored in internal flash memory. For example, if different programs are stored in both internal flash and external memory, the program in external memory will run when the system is booted.

## SDRAM (Volatile)

Volatile SDRAM is used by the operating system to dynamically store the following components:

- Digital and analog signal values
- SIMPL+ variables (default is no options are specified, or if using "volatile" qualifier or #DEFAULT\_VOLATILE)

The actual amount of SDRAM used at any given time depends on the program that is running; therefore, usage is variable (dynamic) during normal operation.

## NVRAM (Nonvolatile)

Nonvolatile NVRAM contain the following components:

- SIMPL+ variables (if "volatile" qualifier or #DEFAULT\_VOLATILE is removed)
- Signals explicitly written to NVRAM (such as Analog RAM, Analog RAM from database, Serial RAM, Serial RAM from database, Analog nonvolatile ramp, Digital RAM, and so forth)

**NOTE:** SIMPL# Pro has no access to the NVRAM. Programmers should write files for persistent variables instead.

## Console Commands

The 3-Series processor is capable of understanding and responding to a set of phrases known as console commands. Console commands are used to configure the control system.

Observe the following about console commands:

- Console commands can be sent to the device using an SSH client once the IP address or hostname of the device is known. Console commands may also be sent to the device using the **Text Console** tool in Crestron Toolbox via one of the supported communication protocols.
- Console commands are grouped logically. Issuing the `help` command from the console responds with various command categories.
- Issuing the `help all` command responds with a list of all exposed console commands for the device.
- The same command may be listed in more than one category. Commands are case insensitive and can be entered from the appropriate prompt.
- To view the available options and parameters for a command, issue the command followed by a space and "?" (for example, `iptables ?`).
- Support for new console commands is added via firmware updates. Refer to the firmware release notes for more information.

# Establish Communications

The control system must establish communications with a computer in order to upload programs, troubleshoot, or perform diagnostics.

Depending on the control system capabilities, the following communication protocols may be used to connect with a 3-Series control system:

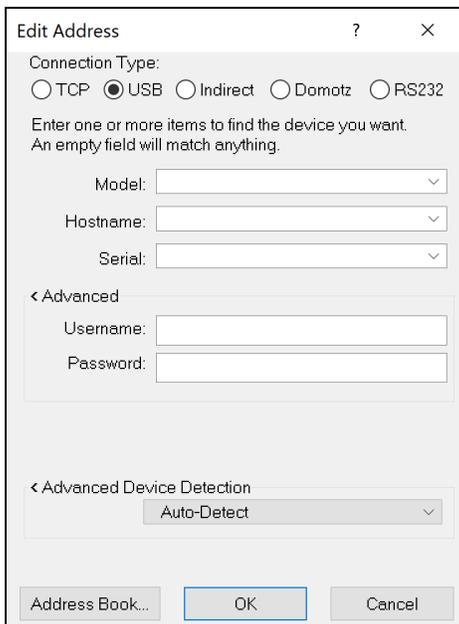
- USB communication with a PC via the **COMPUTER** port on the control system (requires Crestron Toolbox software)
- Ethernet communication via SSH or SSL/TLS

## USB Connection

To connect to the control system via USB:

1. Connect the **COMPUTER** USB port of the control system to the USB port of a computer.
2. Open Crestron Toolbox software.
3. Click the pencil icon  at the bottom left of any tool in Crestron Toolbox. A dialog box for editing the connection type is displayed.
4. Click the **USB** radio button.

### Connection Type Dialog Box - USB

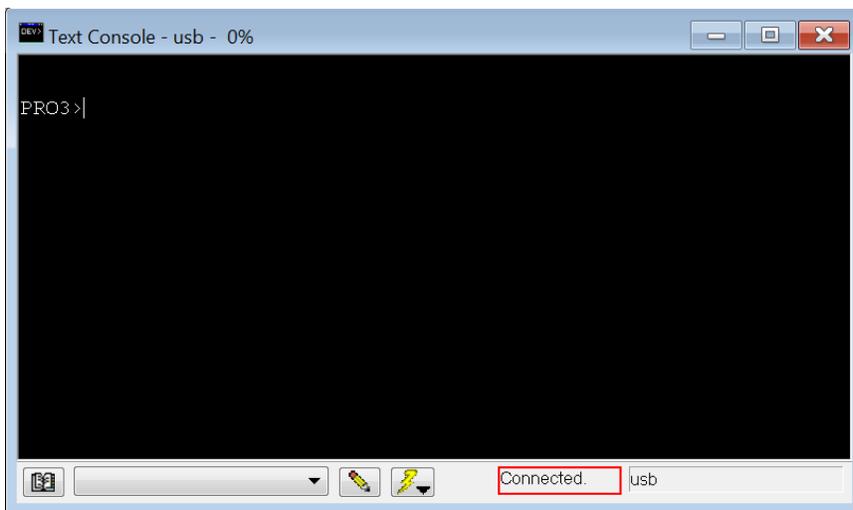


Dialog box titled "Edit Address" showing connection type options (TCP, USB, Indirect, Domotz, RS232) and fields for Model, Hostname, Serial, Username, Password, and Advanced Device Detection (Auto-Detect).

5. Enter the following search parameters for the device (required only if multiple USB connections are active):
  - **Model:** Enter the model name of the Crestron device (for example, PRO3). The model is not user-defined; it is the actual model name of the device as displayed in the **EasyConfig** tool.
  - **Hostname:** Enter the current device hostname
  - **Serial:** Enter the serial number (not the model number or TSID) of the Crestron device. The serial number is a 12-digit number (not beginning with 60 or 65) that may contain letters. The serial number is printed on a sticker affixed to the device and can also be viewed in the **EasyConfig** tool.
6. Enter the following advanced authentication parameters for the device (if required):
  - **Username:** Enter the username required to authenticate device communications
  - **Password:** Enter the password required to authenticate device communications
7. Click **OK**.

The device status reports as "Connected" at the bottom of the selected tool if communications have been established.

#### Text Console - Connection Status



The USB connection information for the control system may be saved using the **Address Book** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

# TCP/IP Connection

DHCP (dynamic host configuration protocol) is enabled by default for 3-Series control systems.

**NOTE:** Crestron Toolbox autodiscovery can be used if the control system has access to the DHCP server. The **Device Discovery** tool may also be used to discover the device and its IP address on the network.

If DHCP is available on the local network, no additional configuration changes are required. If DHCP is not available or if the administrator wishes to configure a static IP, the IP address, default gateway, and DNS server settings must be set.

These settings may be configured via USB using the **Text Console** tool or the **Ethernet Addressing** function in Crestron Toolbox.

To configure Ethernet settings using the **Text Console** tool:

1. Connect the **LAN** port of the control system to the LAN with an Ethernet cable.
2. Open Crestron Toolbox software.
3. Select the **Text Console** tool (**Tools > Text Console**).
4. Establish a USB connection to the control system as described in [USB Connection on page 6](#).
5. Issue the following commands:
  - `dhcp 0 off` - Turns off DHCP to use manually configured network information
  - `ipaddress 0 xxx.xxx.xxx.xxx` - Sets the IP address of the control system to the specific address, where "xxx.xxx.xxx.xxx" is the four octets of the the IP address
  - `ipmask 0 xxx.xxx.xxx.xxx` - Sets the IP mask of the control system to the specified mask, where "xxx.xxx.xxx.xxx" is the four octets of the mask address
  - `defrouter 0 xxx.xxx.xxx.xxx` - Sets the default network gateway to the specified IP address, where "xxx.xxx.xxx.xxx" is the four octets of the default router address
  - `adddns xxx.xxx.xxx.xxx` - Sets the DNS (domain name server) to use for DNS name lookups, where "xxx.xxx.xxx.xxx" is the four octets of the DNS address.

**NOTE:** Although the control system will prompt that a reboot is required, the commands above can be performed prior to executing the reboot.

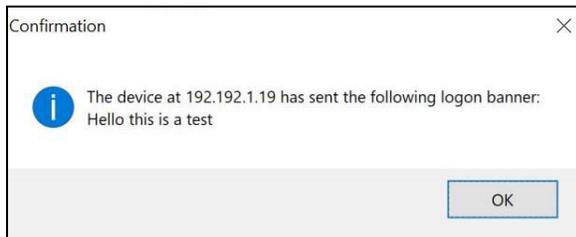
To configure Ethernet settings using the **Ethernet Addressing** function, refer to the appropriate section of the Crestron Toolbox help file.

Once a static or dynamic IP address has been set for the control system, the TCP/IP connection information for the control system may be saved using the **Address Book** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

# Set a Logon Banner

A logon banner can be loaded to the control system that is shown when a user connects to the control system successfully over SSH or web server. A sample logon banner is shown below.

## Sample Logon Banner



To load a logon banner to the control system:

1. Create the banner text file using a text editing application. The text file must be a regular ASCII file (not using UTF-16 or any other encoding).
2. Save the text file as **banner.txt**.
3. Use an SFTP client to load the **banner.txt** file to the **\sshbanner** directory on the control system.

# Time and Date Settings

The internal clock of the control system can be configured using console commands.

1. Issue the `timedate` command to manually set the time and date.
  - **Syntax:** `timedate [hh:mm:ss mm-dd-yyyy]`
    - For `hh:mm:ss`, enter the time in hours (using 24-hour format), minutes, and seconds format.
    - For `mm-dd-yyyy` (or `mm/dd/yyyy`), enter the date in month (1–12), day (1–31), and year format.
  - **Example:** `timedate 12:18:30 03-14-2021`
2. Issue the `timezone` command to set the time zone:
  - **Syntax:** `timezone [list | zone]`
    - For `zone`, enter the three-digit code for the time zone.
    - Use the `list` parameter to print a list of all time zones and their codes in the console.
  - **Example:** `timezone 014`
3. Issue the `sntp start` command to synchronize the internal clock with an SNTP server:
  - **Syntax:** `sntp start [server:address]`
    - For `server:`, enter the address (in dot decimal notation) of the SNTP server.
  - **Example:** `sntp start server:255.255.255.255`

The internal clock can also be set using the **System Clock** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

# Authentication

3-Series control systems provide various authentication options that can be used to create user accounts and passwords, to set password policies, and to set access levels for users and groups.

Use the following procedures to enable authentication and configure authentication settings on the control system.

## Enable Authentication

By default, 3-Series control systems do not have user authentication enabled. Once authentication is enabled, authentication settings may be configured for the control system.

To enable authentication, which will create an initial administrator account:

1. Issue the `authentication on` command from the control system console over a secure connection protocol (SSL/TCP, SSH, USB).
2. When prompted, create a username and password for the administrator account. The password must be at least six characters.

**CAUTION:** Do not lose the username and password for the administrator account. The control system cannot be accessed without this information after an administrator account has been created.

3. Issue the `reboot` command to reboot the device with the new authentication settings.

After rebooting, the control system will prompt for the administrator account username and password before a connection is allowed.

Authentication settings can also be configured using the **Authentication** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

**NOTE:** To manually reset authentication, use a small, pointed object (such as the tip of a pen) to press and hold the recessed **SW-R** button on the control system for 15 seconds.

## Account Recovery

If the username or password for the admin account is lost, the following procedures can be used to recover access.

### Factory Restore

To perform a factory restore, follow the procedure described in [Appendix A: Restore to Factory Defaults on page 64](#). All control system settings are restored to their factory defaults, and all user and group accounts are removed.

Upon connecting to the control system after it has been restored, the control system will prompt to create an admin account as described in [Authentication on page 11](#).

## Password Recovery Command

The `pwdrecovermode` command can be issued to configure a password recovery mode on the control system. The password recovery mode is turned off by default and must be turned on by the user.

**NOTE:** The password recovery mode must be turned on prior to attempting to recover a password.

- **Syntax:** `pwdrecovermode [on | off]`
  - `on` - Turns on password recovery mode for the control system
  - `off` - Turns off password recovery mode for the control system
- **Example:** `pwdrecovermode on`

If password recovery mode is turned on, press and hold the **SW-R** button on the control system for 15 seconds to bypass authentication, which allows the user to log in to the control system without entering a password. The user will be prompted to create a new account following the next reboot.

## User and Group Management

Local users and groups can be added to the control system after an administrator account has been created. Additionally, the control system can grant access levels to existing Active Directory users and groups.

The following sections describe how to manage users and groups on the control system.

**NOTE:** Additional console commands for listing users and groups and for showing user information can be viewed by issuing the `help system` command.

### Add Local User

To add a local user to the control system, issue the `adduser` command.

- **Syntax:** `adduser -n:username -p:password`
  - `-n:` - Specifies the name of the local user that will be created
  - `-p:` - Specifies a password for the local user
- **Example:** `adduser -n:jsmith -p:user01`

A local user is created without any access rights. To assign access rights to a local user, the user must be added to at least one local group. For more information, refer to [Add User to Group on page 15](#).

## Delete Local User

To remove a local user from the control system, issue the `deleteuser username` command.

When a local user is removed, the user is also removed from any local groups.

## Add Local Group

To add a local group to the control system, issue the `addgroup` command.

- **Syntax:** `addgroup -n:groupname -l:accesslevel`
  - `-n:` - Specifies the name of the local group that will be created
  - `-l:` - Specifies the access level for the local group
    - `a` - Administrator
    - `p` - Programmer
    - `o` - Operator
    - `u` - User
    - `c` - Connection only
- **Example:** `addgroup -n:CresProgs -l:p`

**NOTE:** A predefined access level must be assigned to a group when it is created.

When a user is added to a group, the user inherits the access level set for the group. Certain control system functions and console commands are accessible only to users with corresponding access levels.

If a user belongs to multiple groups, the user's access level is the combined access level of all groups that contain the user.

## Delete Local Group

To remove a local group from the control system, issue the `deletegroup groupname` command.

When a local user group is removed, users in the group are not removed from the control system. However, the user will lose the access rights associated with the removed group.

## Add Active Directory Group

To add an existing Active Directory group to the control system, issue the `adddomaingroup` command.

**NOTE:** Use the `adlogin` command to log in to the Active Directory server.

- **Syntax:** `adddomaingroup -n:groupname -l:accesslevel`
  - `-n:` - Specifies the name of the Active Directory group to be added
  - `-l:` - Specifies the access level for the Active Directory group
    - `a` - Administrator
    - `p` - Programmer
    - `o` - Operator
    - `u` - User
    - `c` - Connection only
- **Example:** `adddomaingroup -n:ADProgs -l:p`

**NOTE:** The control system cannot create or remove a group from the Active Directory service, but it can grant an access level to an existing Active Directory group.

All users of the Active Directory group inherit the access level set for the group. Certain control system functions and console commands are accessible only to users with corresponding access levels.

## Remove Active Directory Group

To remove a local group from the control system, issue the `deletedomaingroup groupname` command.

When an Active Directory group is removed from the control system, it is not deleted from the Active Directory service. Once the group is removed from the control system, all members of that group lose access to the control system.

## Add User to Group

To add a local or an Active Directory user to a local group, issue the `addusertogroup` command.

- **Syntax:** `addusertogroup -n:username -g:groupname`
  - `-n:` - Specifies the name of the local or Active Directory user
  - `-g:` - Specifies the name of the local group
- **Example:** `addusertogroup -n:jsmith1 -g:CresProgs`

Local users are created on the control system without any access rights. Adding a user to a local group grants the user the access level assigned to the group.

**NOTE:** The control system cannot create or remove a user from the Active Directory service, but it can grant an access level to an existing Active Directory user. This may be accomplished either by adding the Active Directory user to a local group on the control system or by adding the Active Directory group(s) of which the user is a member to the control system.

## Remove User from Group

To remove a local or an Active Directory user from a local group, issue the `removeuserfromgroup` command.

- **Syntax:** `removeuserfromgroup -n:username -g:groupname`
  - `-n:` - Specifies the name of the local or Active Directory user
  - `-g:` - Specifies the name of the local group
- **Example:** `removeuserfromgroup -n:jsmith1 -g:CresProgs`

## User Group Rights

The control system has built-in access levels representing various roles that can be assigned to a group. These access levels apply to all users within that group. Each access level is associated with a set of specific permissions:

1. Access system information and status (read-only).
2. Connect to the device Web XPanel interface.
3. Authenticate CIP and gateway connections.
4. Receive complete administrator access, including managing user accounts and all system settings.
5. Issue programmer commands for user programs, such as loading programs and related files.
6. Issue operator commands for user programs, such as restarting programs.

The following table indicates the permissions that are given to each of the available access levels. The numbers in the table header row correlate with the numbered list items above.

#### Default Rights of Local Groups

Group	1	2	4	5	6	7
Administrator	Yes	Yes	Yes	Yes	Yes	Yes
Programmer	Yes	Yes	Yes	No	Yes	Yes
Operator	Yes	Yes	Yes	No	No	Yes
User	No	Yes	No	No	No	No
Connection Only	No	Yes	Yes	No	No	No

By default, the control system has five groups available (one for each access level): Administrator, Programmer, Operator, User, and Connection Only. The initial user is added to the Administrator group. The default groups may be used, or custom groups can be created with the appropriate access level permissions as needed

## Password Management

The following sections explain how to manage passwords for local users on the control system.

### Set Password Policy

To set the password policy for the control system, issue the `setpasswordrule` command.

- **Syntax:** `setpasswordrule {-all | -none} | { -length:minpasswordlength} {-mixed} {-digit} {-special}`
  - `-all` - All password rules are applied.
  - `-none` - No password rules are applied.
  - `-length:` - Specifies the minimum password length. By default, the minimum password length is six characters.
  - `-mixed` - Specifies that the password must contain a lower and upper case character.
  - `-digit` - Specifies that the password must contain a number character.
  - `-special` - Specifies that the password must contain a special character.

**NOTE:** The `-length`, `-mixed`, `-digit`, and `-special` parameters cannot be combined with `-none`.

- **Example:** `setpasswordrule -length:9 -mixed -digit -special`

**NOTE:** The following special characters are allowed: `` ~ ! @ $ % ^ & * ( ) _ + = { } [ ] | ; " < > , .`

All passwords that are created, updated, or reset for local users must follow the password rules set by this command to be considered valid.

## Update Local Password

To update the current user's password, issue the `updatepassword` command.

Users may update their password. The user is prompted to enter the current password once and the new password twice. If the old password does not match the current password, the operation fails and the password is not changed.

## Reset User Password

To reset a user's password, issue the `resetpassword` command.

- **Syntax:** `resetpassword -n:username -p:defaultpassword`
  - `-n:` - Specifies the user whose password will be reset
  - `-p:` - Specifies a default password that can be provided to the user following the reset
- **Example:** `resetpassword -n:jsmith1 -p:Default321!`

## Login Behavior

Users must enter a username and password to connect to the control system.

A user is given a maximum of three login attempts for each type of transport protocol (Ethernet and USB). If a user fails to authenticate against console within the maximum attempts allowed, the transport protocol used to attempt the connection is blocked.

- For USB transport, the transport is blocked for 5 minutes after the maximum logon attempt is reached. If the user tries again after 5 minutes and continues to fail, the block time is doubled. The block time continues to be doubled until a successful logon or control system reboot happens. Once a user successfully authenticates against console, the failure count is reset to zero, and the block time is reset to 5 minutes.
- For Ethernet transport, after a remote IP fails to logon within maximum attempts allowed, the console forces the transport to close and blocks further logon attempts from that remote IP for 24 hours.

**NOTE:** To regain access to the control system from a blocked IP, successfully log in to the console over USB or from a different IP address, and then use the `remblockedip` console command to remove the blocked IP. For more information, refer to [Blocked IP Address Functions on page 20](#).

## Local User Login

If a user opens a connection to the console, the console prompts the user for a username and password as shown in the example below.

```
PR04 Console
Login: jsmith1
Password: *****
PR04>
```

Local users are created with no access rights. Even if a user has an account in the control system, the user cannot connect to the control system console unless the user been added to a group. To grant access to the user, an administrator must ensure that the user has been first added to a group.

## Active Directory Login

To log onto the console as an Active Directory user, both the domain name and username must be provided (separated by a "\" or "/") when prompted by the console.

```
PR04 Console
Login: csusers\jsmith1
Password: *****
PR04>
```

After an administrator adds an Active Directory user or group to the control system, the name and SID of the user or group is stored in the control system.

When an Active Directory user attempts to authenticate against the console, the console in turn uses the user credentials to authenticate against the Active Directory service. If the Active Directory authentication is successful, the console queries the Active Directory service for the user's SID:

- If the user has been added to the control system, the console compares the SID from the Active Directory service with the stored SID. Access is granted to the user only if the SIDs match.
- If the user has not been added to the control system, the console queries the Active Directory service for all groups containing the user and retrieves the group SIDs. The console then iterates these SIDs to compare them to the stored group SIDs. Access is granted to the user only if at least one match is found.

# Session Timeout Functions

By default, a user is never logged off automatically unless the value for the logon session timeout is manually changed.

If the value for logon session timeout has been changed, the console starts a timer after a user logs in and monitors the user's activities. If a user is idle for more than a set duration, the console logs the user out automatically.

## Change Session Timeout Duration

To change the duration for the logon session timeout, issue the `setlogoffidletime` command.

- **Syntax:** `setlogoffidletime [minutes]`
  - `minutes` - The duration (in minutes) that must elapse before the console logs off an idle user. Entering 0 disables the user from being logged off automatically. The default value is "20."
  - No parameter - Displays the current timeout setting
- **Example:** `setlogoffidletime 30`

## Change Login Failure Count

To change the value for the logon failure count, issue the `setloginattempts` command.

- **Syntax:** `setloginattempts [number]`
  - `number` - The number of login attempts allowed before the console is blocked. Entering 0 allows unlimited attempts. The default value is "3."
  - No parameter - Displays the current setting
- **Example:** `setloginattempts 3`

# Locked User Functions

Administrators are able to lock user accounts, which prevents access to the control system via the console and Crestron Toolbox.

## Add User to Locked List

To add a user to the locked list, issue the `addlockeduser` command.

- **Syntax:** `addlockeduser [name]`
  - `name` - Enter the user account that will be locked.
  - No parameter - Lists all locked user accounts
- **Example:** `addlockeduser jsmith1`

## Remove User from Locked List

To remove a users from the locked list, issue the `remlockeduser` command.

- **Syntax:** `remlockeduser [name]`
  - `name` - Enter the user account that will be removed from the locked list
  - No parameter - Lists all locked user accounts
- **Example:** `remlockeduser jsmith1`

# Blocked IP Address Functions

When a user reaches the maximum number of login attempts over an Ethernet connection, the client's IP address is blocked.

## Change Lock out Time

To change the duration (in hours) that an IP address is blocked by the console, issue the `setlockouttime` command.

- **Syntax:** `setlockouttime [number]`
  - `number` - The number of hours to block an IP address. Entering 0 blocks the IP address indefinitely. 255 is the maximum value. The default value is "24."
  - No parameter - Displays the current setting
- **Example:** `setlockouttime 24`

## Add IP Address to Blocked List

To add an IP address to the blocked list manually, issue the `addblockedip` command.

- **Syntax:** `addblockedip [ipaddress]`
  - `ipaddress` - Enter the IP address that will be blocked.
  - No parameter - Lists all blocked IP addresses
- **Example:** `addblockedip 255.255.255.255`

## Remove IP Address from Blocked List

To remove an IP address from the blocked list manually, issue the `remblockedip` command.

- **Syntax:** `remblockedip [ipaddress]`
  - `ipaddress` - Enter the IP address that will be removed from the blocked list.
  - No parameter - Lists all blocked IP addresses
- **Example:** `remblockedip 255.255.255.255`

# Certificate Management

Security certificates for 802.1X and other security protocols can be added, removed, and managed from the console.

The control system supports five types of certificates:

- **Root:** The Root certificate is used by the control system to validate the network's authentication server. 3-Series control systems have a variety of Root certificates, self-signed by trusted CAs (Certificate Authorities), that are preloaded into the device. Root certificates must be self-signed.
- **Intermediate:** The Intermediate store holds non self-signed certificates that are used to validate the authentication server. These certificates are provided by the network administrator if the network does not use self-signed Root certificates.
- **Machine:** The machine certificate is an encrypted PFX file that is used by the authentication server to validate the identity of the control system. The machine certificate will be provided by the network administrator, along with the certificate password.

**NOTE:** Only one machine certificate may be stored on the control system for 802.1X.

- **WebSocket:** A WebSocket certificate is used to validate the network's authentication server via the WebSocket (WSS) protocol.
- **User:** The User store holds additional certificates not used in the 802.1X standard.

Certificates can also be managed using the **Security Certificates** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

## Certificate Requirements

3-Series control systems support all standard X.509v3 certificates that use the following:

- RSA key with length 2048, 3072, or 4096 bits
- Hash Algorithms using SHA-1, SHA-256, SHA-384, or SHA-512

# Add a Certificate

To add a certificate to a certificate store on the control system:

1. Use an SFTP or SCP client to upload the certificate file (in .cer or .pem format) to the "\user" directory.
2. Use an SSH console or Crestron Toolbox to copy the certificate file to the "\romdisk\user\cert" directory.
3. Issue the `certificate add certificate_store <certificate_name> <certificate_uid> <password>` command.
  - `certificate_store`: The certificate store where the certificate file will reside [root|machine|user|intermediate|websocket]
  - `certificate_name`: The name of the certificate file
  - `certificate_uid`: The unique identifier of the certificate file
  - `password`: The password for the certificate file (machine certificates only)

## TLS/SSL

3-Series control systems provide support for Transport Layer Security (TLS) and Secure Sockets Layer (SSL). TLS/SSL is a protocol that provides a secure channel for communication between two machines. The secure channel is transparent and passes data through unchanged. The data is encrypted between the client and the server, but the data the one end writes is exactly what the other end reads.

**NOTE:** 3-Series control systems only support TLS/SSL over TCP/IP. TLS/SSL is set to "off" by default and is set to "self" after authentication is turned on.

To enable TLS/SSL, issue the `ssl` command:

- **Syntax:** `ssl [off | self | ca [-p:privatekeypassword]]`
  - `off`: Turns off SSL if it is on
  - `self`: Turns on SSL using a self-signed certificate
  - `ca`: Turns on SSL using a CA-signed certificate. The `-p` argument must be provided with the private key password for the CA-signed certificate
  - No parameter: Displays the current setting
- **Example:** `ssl ca -p:myprivatekeypassword`

**NOTE:** When TLS/SSL is enabled, the control system uses a server certificate. For more information, refer to [Server Certificates on page 24](#).

To configure TLS settings without reconfiguring SSL, issue the `TLSVERSION` command:

- **Syntax:** `TLSVERSION [TLS1.0 | TLS1.2]`
  - `TLS1.0`: Implies that TLS 1.0 is the minimum version for TLS connections
  - `TLS1.2`: Implies that TLS 1.2 is the minimum version for TLS connections
  - No parameter: Displays the current setting
- **Example:** `TLSVERSION TLS1.2`

TLS/SSL certificates may also be managed using the **SSL Management** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

## Server Certificates

When authentication is enabled, the control system uses a server-side certificate to authenticate various control system components, including the web server.

One of the following three server-side certificate types may be used:

- A self-signed certificate that is generated by the control system
- A CA (Certificate Authority)-signed certificate and signing chain that are loaded onto the control system
- An externally requested and signed certificate, signing chain, and private key that are loaded onto the control system

Instructions for configuring each server certificate type are provided in the sections that follow.

### Self-Signed Certificates

To use a self-signed certificate with TLS/SSL:

1. Issue the `ssl self` command in the control system console.
2. Issue the `reboot` command to reboot the control system.

### CA-Signed Certificates

The following procedures are used to obtain and load a CA-signed certificate to the control system.

## Generate a Certificate Signing Request (CSR)

To generate a certificate signing request, issue the `createcsr c:st:l:o:ou:cn:e [-i:option]` command, where the following parameters are replaced with the appropriate data that should appear in the certificate:

**NOTE:** Any parameter that is not required can be left blank as needed.

- `c`: The two letter country code (corresponding to ISO 3166)
- `st`: State or province name
- `l`: Locality or city name
- `o`: Organization name (required)
- `ou`: Organizational or unit name
- `cn`: Common name (required)

**NOTE:** The common name is not transferable, and thus must be one that is used by clients. The common name must be officially registered to the company; otherwise, the certificate request is rejected.

- `e`: Email address
- `-i`: Ignores blank parameters in the CSR request. Valid values are `true` and `false`.

By default, a certificate request for a certificate with a 2048-bit RSA signature is requested. The CSR "request.csr" file is saved automatically to the "\sys\" directory of the control system.

## Obtain the Certificate

The exact procedures required to obtain a CA-signed certificate differ depending on the CA, but in all cases, it is necessary to submit the request.csr file along with any other verification that the CA requires.

To obtain the request.csr file:

1. Using SSH, issue the `move \sys\request.csr \user` command.
2. Use an SCP client to copy the request.csr file from \user directory of the control system.

In some cases, it may be necessary to open the .csr file in a text editing program and to copy and paste the text between the "Begin certificate request" and "End certificate request" delimiters before sending the file to the CA.

**NOTE:** All certificate files must be in .pem format.

## Load the Certificate Files

Once the CA validates the request.csr file, the CA issues the validated certificate to the requester. The following certificate files are required for deployment on the control system:

- Signed CA-signed certificate in .cer format (base-64 encoded)
- Certification chain (concatenation of the issuing CA and its CA) in .cer format (base-64 encoded)

To upload the CA-signed certificate to the control system:

1. Rename the three certificate files as follows:
  - Rename the signed certificate file to "srv\_cert.cer".
  - Rename the certification chain file to "rootCA\_cert.cer".
2. Use an SCP or SFTP client to copy the two certificate files to the \user directory on the control system.
3. Connect to the control system via SSH or Crestron Toolbox.
4. Issue the `delete \sys\rootCA_cert.cer` and `delete \sys\srv_cert.cer`, commands to delete any existing certificate files.
5. Issue the `move \user\rootCA_cert.cer \sys` and `move \user\srv_cert.cer \Sys` commands to move the new certificate files to the \sys directory.

## Enable TLS/SSL with the CA-Signed Certificate

To enable TLS/SSL with the CA-signed certificate:

1. Issue the `ssl ca -p:[privatekeypassword]` command in the control system console.
2. Issue the `reboot` command to reboot the control system.

## Externally-Signed Certificates

The following procedures are used to load an externally-signed certificate to the control system.

The following certificate files are required for deployment on the control system. These files are generally provided by the IT administrator:

- Private key in .pem format
- Signed CA-signed certificate in .cer format (base-64 encoded)
- Certification chain (concatenation of the issuing CA and its CA) in .cer format (base-64 encoded)

## Load the Certificate Files

To upload the externally-signed certificate to the control system:

1. Rename the three certificate files as follows:
  - Rename the private key file to "srv\_key.pem".
  - Rename the signed certificate file to "srv\_cert.cer".
  - Rename the certification chain file to "rootCA\_cert.cer".
2. Use an SCP or SFTP client to copy the three certificate files to the \User directory on the control system.
3. Connect to the control system via SSH or Crestron Toolbox.
4. Issue the `delete \sys\rootCA_cert.cer`, `delete \sys\srv_cert.cer`, and `delete \sys\srv_key.pem` commands to delete any existing certificate files.
5. Issue the `move \user\rootCA_cert.cer \sys`, `move \user\srv_cert.cer \sys`, and `move \user\srv_key.pem \sys` commands to move the new certificate files to the \sys directory.

## Enable TLS/SSL with the Externally-Signed Certificate

To enable TLS/SSL with the externally-signed certificate:

1. Issue the `ssl ca -p:[privatekeypassword]` command in the control system console.
2. Issue the `reboot` command to reboot the control system.
3. Issue the `del \sys\*.cer` and `del \sys\*.pem` commands.

# 802.1X

802.1X is an IEEE network standard designed to enhance the security of wireless and Ethernet LANs. It is widely used in corporate networks to provide an authentication mechanism for devices wishing to connect to the network. The standard relies on the exchange of messages between the device and the network's host, or authentication server.

To enable and configure 802.1X for the control system:

1. Issue the `8021xauthenticate` on command to enable 802.1X.
2. Issue the `8021xvalidateserver` on command to allow the control system to verify the identity of the network's authentication server. Issuing this command allows certificates signed by trusted CAs (Certificate Authorities) to be selected, which is used during server validation.

**NOTE:** Using the `8021xvalidateserver` command is optional based on the recommendation of the network administrator, but the option should be enabled for most applications.

3. Issue the `8021xmethod [password | certificate]` command to select the secure password method or the certificate method depending on the network administrator's requirement.
4. If the certificate method was selected, issue the `certificate add machine {certificate_name} {certificate_uid} {password}` command to add the machine certificate supplied by the network administrator to the certificate store. Refer to [Add a Certificate on page 23](#).

**NOTE:** The machine certificate is an encrypted PFX file that will be supplied by the network administrator, along with the certificate password. The machine certificate is used to verify the identity of the control system.

5. If the password method was selected, enter the username and password supplied by the network administrator:
  - a. Issue the `8021xusername [username]` command to enter the username supplied by the network administrator.
  - b. Issue the `8021xpassword [password]` command to enter the password supplied by the network administrator.

**NOTE:** A machine certificate is not required for the password method.

6. If the validate server option is enabled, select the certificates that will be used for server validation:
  - a. Issue the `8021xtrustedcas list` command to list all trusted certificates stored on the control system.
  - b. Issue the `8021xtrustedcas use {certificate_name} {certificate_uid}` command to enable a certificate for validation. Multiple certificates may be enabled.

**NOTE:** If the network does not use any of the listed certificates, the network administrator will provide a certificate that must be uploaded to the control system manually using the `certificate add [certificate_store] {certificate_name} {Certificate_uid}` command. If the certificate is self-signed, enter `root` for `certificate_store`. If the certificate is not self-signed, enter `intermediate` for `certificate_store`.

7. If required, issue the `8021xdomain [domain_name]` command to set the domain name of the network.
8. Issue the `reboot` command to reboot the control system with the new 802.1X settings.

802.1X configuration can also be performed using the **802.1X** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

# Firmware Updates

To take advantage of all the latest device features, the control system should always contain the latest firmware, which is available for download at [www.crestron.com/Support](http://www.crestron.com/Support).

**NOTE:** Access to firmware is reserved for Authorized Crestron dealers, Crestron Service Providers (CSPs), and Crestron partners only. New users must register for an account to access certain areas of the Crestron website. For more information on registering, navigate to [www.crestron.com/register](http://www.crestron.com/register).

To perform a firmware update:

1. Download the latest device firmware (.puf file) at [www.crestron.com/Support](http://www.crestron.com/Support).
2. Use an SFTP client to transfer the firmware .puf file to the control system's **/firmware** directory.
3. Issue the `puf <filename>` command in the control system console, where `<filename>` is the complete filename of the .puf file, including the filename extension.

**NOTE:** Firmware updates can be scheduled using the control system's auto update mechanism. For more information, refer to [Auto Update Mechanism on page 51](#).

The **Package Update Tool** in Crestron Toolbox can also be used to send firmware to the control system and to manage the firmware update. For more information, refer to the Crestron Toolbox help file. The web user interface for the control system can also be used to load firmware.

# Message Logging

3-Series control systems provide messages when the control system performs a task or when it encounters an issue, such as a hardware or software failure, hardware incompatibility with software definitions, or a programming error.

**NOTE:** A **MSG** LED on the control system or controlled device may light to indicate that an error has occurred.

Messages created by the control system are written to a persistent log that is stored in the internal flash memory. The persistent log can be saved to external storage on supported models and can be viewed through the console or through Crestron Toolbox.

## Persistent Log

The persistent log (PLOG) is a log of messages defined by the control system. The current PLOG can be viewed in the control system console and is available following a device reboot. The control system stores the contents of the current PLOG file at `\plog\CurrentBoot`. The log file is locked and cannot be opened for transfer.

To print the contents of the current PLOG in the console, issue the `err plogcurrent` command.

Observe the following about the PLOG:

- If a soft reboot is performed, any pending messages are written to the latest log file and zipped into one file. On reboot, the zipped file at `\plog\CurrentBoot` is moved to `\plog\PreviousBoot`. During subsequent reboots, the zipped file from `\plog\PreviousBoot` is moved to `\plog\ZippedLogs` for storage.
- The control system logs errors as long as there are not over 250 messages per two-minute period for two consecutive two-minute logging periods:
  - For each error period, the console displays a `PersistentLog: Error state threshold met` message if error logging is suspended.
  - When logging is suspended, the log file displays a `PersistentLog: Consecutive error states detected; logging is suspended` message, and the user receives a `PersistentLog is suspended, please contact dealer` message in the console.
  - Message logging resumes when there has been less than 10 error messages logged for 10 consecutive two-minute logging periods. When logging resumes, all messages from the 10 previous logging periods are logged to the PLOG file. The log in the console also displays a `PersistentLog: Consecutive quiet states detected; logging is resumed` message.

The PLOG can also be viewed in Crestron Toolbox using the **Error Log** function. For more information, refer to the Crestron Toolbox help file.

# Message Levels

The following table defines the six levels of messages that can appear in the persistent log.

## 3-Series Control System Message Levels

Type	Description
OK	General information about an event that has occurred
Info	General information about an event that has occurred
Notice	An event has occurred that is noteworthy but that does not affect program operation
Warning	An event has occurred that could affect program operation, but the program still runs normally
Error	An event has occurred indicating the program is not running as expected
Fatal	An event has occurred that prevents the program from running

# Message Format

Each error message has the following format: `Level: Message.`

- `Level:` - The message level
- `Message` - A description of the message

Some error messages have a suffix with additional information in parenthesis:

`Level: {Application} [App#] # [Date/Time] # Message`

**Example:** `Ok: TLDM # 2021-03-12 16:23:43 # Router got Connected`

When reporting an error message to a [Crestron True Blue support](#) representative, report the exact message as it appears in the error log. The `Application` field indicates the program that produced the error.

**NOTE:** The `App#` field is displayed when there are multiple instances of a single application (such as `SimplSharpPro.exe`) running and indicates the program slot (1–10) to which the program is associated.

# Remote System Logging

Control system messages can be captured and stored on a remote server using the remote system logging function.

**NOTE:** The remote server host must have system log server with the applicable security certificates and sufficient disk space to store the active system log. The host should also be configured to archive older system logs and to off load them over time. If TLS is enabled, a TLS-enabled server with the appropriate certificates is required.

To configure remote system logging, issue the `remotesyslog` command.

- **Syntax:** `remotesyslog [-s:] {-e:} {-a} [-i:address] [-p:port] {-t:protocol} {-v:on|off}`
    - `-s: [on|off]` - Enables or disables remote system logging
    - `-e: [level]` - Logs all messages at the provided error level and above. The default setting is "Notice."
      - `ok`
      - `info`
      - `notice`
      - `warning`
      - `error`
      - `fatal`
    - `-a` - Appends the contents of the audit log to the system log
- NOTE:** Audit logging must be turned on to use this feature. For more information, refer to [Audit Logging on page 34](#).
- `-i:address` - Sets the remote server IP address for the system log in dot decimal notation or as an ASCII string containing the server host name (maximum 255 characters)
  - `-p:port` - The remote server port number in decimal notation
  - `-t:protocol` - The protocol used to connect to the remote server
    - `tcp`
    - `udp`
    - `ssl`
  - `-v:` - If SSL is enabled, set to `on` to require server verification or set to `off` to not require server verification.
  - No parameter: Displays the current setting
- **Example:** `remotesyslog -s:on -e:notice -a -i:255.255.255.255 -p:12345 -t:ssl -v:off`

Remote system logging can also be configured in Crestron Toolbox using the **Syslog** function. For more information, refer to the Crestron Toolbox help file.

## Audit Logging

Audit logging can be enabled on the control system to track logons, logoffs, account management changes, and console commands.

**NOTE:** Logons, logoffs, and authentication management is always logged by the control system regardless of audit log settings.

## Configure Audit Logging

To configure audit logging on the control system, issue the `auditlogging` command. Audit logging is turned off by default.

- **Syntax:** `auditlogging [on|off] {[all]|[none]}{admin} [prog] [oper] [user]} [remotesyslog]}`

- `on` - Turns on audit logging
- `off` - Turns off audit logging
- No parameter: Displays the current audit logging setting

The following access level parameters are optional and are used to log commands by access level:

- `admin` - Logs administrator-level commands
  - `prog` - Logs programmer-level commands
  - `oper` - Logs operator-level commands
  - `user` - Logs user-level commands
  - `all` - Logs all commands
  - `none` - Logs no commands
  - `remotesyslog` - Writes to the remote syslog server only
- **Example:** `auditlogging on admin oper`

**Example log output:** `[2021-11-30T07:02:44-08:00]: EVENT: COMMAND(SHELL 172.30.255.255) USER: admin # AUDITLogging on all`

## View Audit Logs

To display the audit log in the console, issue the `getauditlog` command.

To print the last 50 audit log entries in the console, issue the `printauditlog` command. The optional `all` parameter can be appended to print the entire log.

**NOTE:** Use the **Audit Logs** function in Crestron Toolbox to change the audit log storage location and file name. For more information, refer to the Crestron Toolbox help file.

## Clear Audit Log

To clear all entries in the audit log, issue the `clearauditlog` command.

# Control Subnet

The AV3, PRO3, and CP3N have a dedicated Control Subnet, which allows for dedicated communication between the control system and Crestron Ethernet devices without interference from other network traffic on the LAN.

When using the Control Subnet, observe the following:

**CAUTION:** Do not connect the **CONTROL SUBNET** port to the LAN. The **CONTROL SUBNET** port must only be connected to Crestron Ethernet devices.

- The control system acts as a DHCP server to all devices connected to the Control Subnet and assigns IP addresses as needed.
- A DNS server is built in to the control system to resolve host names.
- Only connect Crestron Ethernet devices to the Control Subnet.
- The control system can also operate in isolation mode by issuing the `isolatenetworks on` command. When in isolation mode, the firewall is configured so that no communication can occur between the LAN and devices on the Control Subnet. Using this mechanism, customers can protect their corporate LAN from devices on the Control Subnet.
- Devices on the Control Subnet do not have any resources on the LAN side. For example, if a touch screen with a SmartObjects® technology object requiring network access is installed on the Control Subnet, the object will not work.
- Devices on the LAN do not have access to any devices on the Control Subnet. Crestron Toolbox also does not have access to these devices when it is connected to the LAN. To configure devices on the Control Subnet with Crestron Toolbox (outside of runtime), the computer running Crestron Toolbox must be connected to the Control Subnet.
- Any NAT/port mapping rules that were previously created do not work when the control system is in isolation mode.

**NOTE:** If the control system is running in isolation mode, Crestron Ethernet devices requiring Internet access should not be connected to the **CONTROL SUBNET** port (directly or indirectly) and should be instead connected to the LAN.

# Prepare the Control Subnet

Before enabling the Control Subnet on the control system, note the following assumptions:

- The system is not capable of dual authorization.
- Physical security is assumed to be provided by the environment.
- Administrators are trusted to follow and apply all administrator guidance as needed.

To prepare the Control Subnet:

1. Issue the `AUTH ON` command from the control system console over a secure connection protocol (SSL/TCP, SSH, USB).
2. When prompted, enter a username and password for the administrator account. The password must be at least six characters.

**CAUTION:** Do not lose the username and password for the administrator account. The control system cannot be accessed without this information after an administrator account has been created.

3. Issue the `reboot` command to reboot the device with the new authentication settings.
4. Create other users and assign them to groups as needed. For more information, refer to [User and Group Management on page 12](#).
5. Issue the `CIPHER STRONG` command to update the SSH ciphers setting.
6. Issue the `SSHPORT OFF` command to disable SSH.
7. If the installation requires Banners, copy the Banner to the following device folder: `"\SSHBanner\banner.txt"`.

At this time, FTP and HTTP services will be disabled. HTTPS will continue to be available.

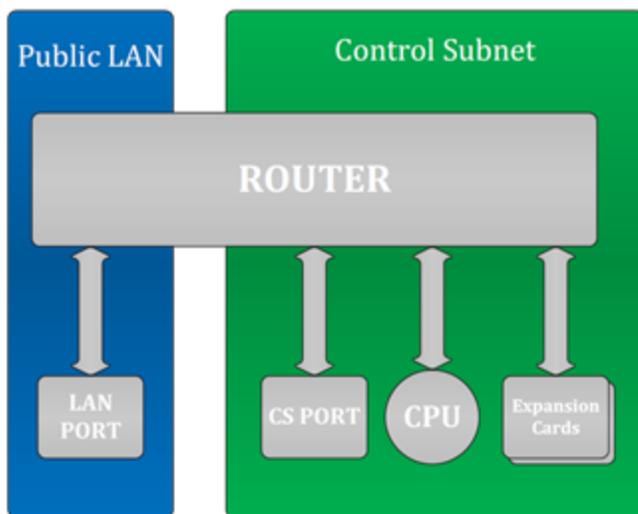
# Control Subnet Architecture

Even if nothing is plugged into the **CONTROL SUBNET** port(s) on the back on the control system, the following are still present on the Control Subnet:

- Control System CPU (Where AV programs run)
- Optional Expansion cards (PRO3 and AV3 only)

This design is in place to ensure that the Crestron CPU and optional expansion cards are protected from malicious packets on the LAN. Refer to the diagram below for more information on how these components work together.

**Public LAN/Control Subnet Diagram**



The firewall rules permit entry to only the traffic that is listened to by the CPU. As a result, a port scan will only show ports that are listened to by the CPU. Users can set up manual port forwarding rules to make custom connections to the devices on the Control Subnet. For more information, refer to [Appendix B: Port Forwarding on page 65](#).

Crestron Toolbox can create custom port forwarding rules in the 64000–64299 range to enable management of the devices on the Control Subnet. These port forwarding rules are created when the tool connects and are broken down when the tool disconnects or when the device is rebooted.

# Firewall Rules in Normal Operation

Under normal operating procedures, the firewall on the control system router behaves as follows.

## Control System Firewall Rules - Normal Operation

Direction	Port(s)	Rule	Description
Inbound from LAN	20, 21	To CPU	FTP (if enabled)
Inbound from LAN	22	To CPU	SSH
Inbound from LAN	80, 443	To CPU	Web server (if enabled)
Inbound from LAN	843	To CPU	Flash policy server (if enabled)
Inbound from LAN	41794, 41796	To CPU	Crestron communication protocols
Inbound from LAN	Listen ports used by program	To CPU	Programmatic listeners
Inbound from LAN	49200	To CPU	Crestron HTML5 User Interface
Inbound from LAN	64000–64299	To devices on control system	Allows Crestron management tools to access devices on the Control Subnet; ports are opened and closed as needed
Control Subnet Outbound to LAN	Any port	Allowed	All outbound traffic is allowed
Inbound from LAN	User defined	User defined	Allows manual port forwarding to devices on Control Subnet

# Firewall Rules in Isolation Mode

When running in isolation mode, the firewall on the control system router behaves as follows.

## Control System Firewall Rules - Isolation Mode

Direction	Port(s)	Rule	Description
Inbound from LAN	20, 21	To CPU	FTP (if enabled)
Inbound from LAN	22	To CPU	SSH
Inbound from LAN	80, 443	To CPU	Web server (if enabled)
Inbound from LAN	843	To CPU	Flash policy server (if enabled)
Inbound from LAN	41794, 41796	To CPU	Crestron communication protocols
Inbound from LAN	Listen ports used by program	To CPU	Programmatic listeners
Inbound from LAN	49200	To CPU	Crestron HTML5 User Interface
Inbound from LAN	64000–64299	Blocked	In isolation mode, Crestron management tools cannot connect to any devices on the Control Subnet
Control Subnet Outbound to LAN	Any port	All other devices: Blocked	No outbound traffic is allowed
Inbound from LAN	User defined	Blocked	In isolation mode, no port forwarding can be managed by the user

# Program Management

3-Series control systems are equipped with program slots that are used to store program files. Program files can be created using SIMPL, SIMPL# Pro, and Crestron Studio, and allow the control system to be custom programmed to perform certain tasks or enable certain system functionality.

## Load Programs to the Control System

A program must be compiled and uploaded to the control system before it can be run by the control system.

- For more information on creating, compiling, and uploading a program using SIMPL, SIMPL# Pro, or Crestron Studio, refer to the appropriate tool help file.
- For more information on uploading a program using Crestron Toolbox, refer to the Crestron Toolbox help file.

Program files can also be manually loaded to the control system by using an SFTP client to transfer the program file to the `/simpl/appXX` directory, where "XX" is the respective program slot on the control system.

## IP Table Configuration

Programs running on the control system that use Ethernet to communicate between the control system and network-enabled devices require an IP table. The IP table allows the control system to identify and communicate with Ethernet equipment on an IP network.

Each controlled Ethernet device has an IP table, which is also known as a primary list. The primary list specifies the IP ID of the controlled device and the IP address or fully-qualified domain name (FQDN) of the control system(s) that sends it commands.

The control system IP table lists the IP address/FQDN and the IP ID of every device in the network. The IP ID is a hexadecimal value that must be unique and ranges from 03 to FE.

**NOTE:** IP tables used in Ethernet-based primary-secondary applications have unique IP table requirements. For more information, refer to [Primary-Secondary Mode on page 46](#).

The following commands can be used to view, configure, create, and remove IP table entries for the control system.

**NOTE:** IP table information can also be entered using information given in SIMPL and SIMPL# Pro or via Crestron Toolbox. For more information, refer to the appropriate help file.

## View and Configure IP Table

To view and configure IP table settings for the control system, issue the `iptables` command.

- **Syntax:** `iptables [-p:program] [-t] [-i:id] [-c] [-o]`
  - `-p:program` - Enter `all` to display all programs or enter a number (0–10) to display the program for the respective program slot
  - `-t` - Displays IP table data in table format
  - `-i:id` - Enter the IP ID to display the entry for that ID.
  - `-c` - Clears the IP table for the specified program (requires `-p` parameter)
  - `-o` - Displays offline devices only
  - No parameters - Shows the IP Table for Program 1
- **Example:** `iptables -all -t`

## Add Peer Entry

To add a peer (secondary) entry to the IP table, use the `addpeer` command.

- **Syntax:** `addpeer cipid ip_address/name [-d:device_id] [-c:cipport] [-p:program] [-u:roomid]`
  - `cipid` - The ID of the CIP node (in hexadecimal format)
  - `ip_address/name` - The IP address (in dot decimal notation) or the name of the site for DNS lookup
  - `-d:device_id` - The device ID in the device redirection table (in hex, must be less than 256)
  - `-c:cipport` - The CIP port number for the connection (must be greater than 256)
  - `-p:program` - The program number on the control system that uses the device (default is 1)
  - `-u:roomid` - The room ID used for communication with a Crestron Virtual Control server (max length is 32 characters, valid values are A–Z and 0–9)
- **Example:** `addpeer 13 255.255.255.255 -d:134 -c:458 -p:3 -u:avf469`

**NOTE:** For more information on connecting a device to Crestron Virtual Control, refer to the [Crestron Virtual Control Server Software Product Manual](#).

## Remove Peer Entry

To remove a peer (secondary) entry from the IP table, use the `rempeer` command.

- **Syntax:** `rempeer ciped ip_address/name [-d:device_id] [-c:cipport] [-p:program] [-u:roomid]`
  - `ciped` - The ID of the CIP node (in hexadecimal format)
  - `ip_address/name` - The IP address (in dot decimal notation) or the name of the site for DNS lookup
  - `-d:device_id` - The device ID in the device redirection table (in hex, must be less than 256)
  - `-c:cipport` - The CIP port number for the connection (must be greater than 256)
  - `-p:program` - The program number on the control system that uses the device (default is 1)
  - `-u:roomid` - The room ID used for communication with a Crestron Virtual Control server (max length is 32 characters, valid values are A–Z and 0–9)
- **Example:** `rempeer 13 255.255.255.255 -d:134 -c:458 -p:3 -u:avf469`

## Load IP Table

To load a program-specific DIP file from removable media to the `\sys\` directory of the control system, issue the `loadiptable` command.

- **Syntax:** `loadiptable -p:[appid] [path]`
  - `-p:[appid]`: The specific program identifier
  - `path`: The path of the file on removable media, including "\" (such as `\rm\tmpdir` or `\rm2\dipdir`)
- **Example:** `loadiptable -p:1 \rm\dipdir`

**NOTE:** The program must be restarted for the new IP table to take effect.

# Run Multiple Programs

3-Series processors run multiple programs simultaneously to allow programmers to independently develop and run device specific programs for AV, lighting, HVAC, security, and so forth. As a system grows, processing resources can easily be shifted from one 3-Series processor to another without rewriting any code.

## Device Registration Considerations

To keep the system running seamlessly, consider the following when stopping and starting programs:

- To ensure that devices are registered by the correct programs, note that programs restart in ascending order by program slot when the device is rebooted or when the `progreset` command is entered. For example, Program 1 starts before Program 2.
- Since most devices can only be registered by a single program, the first program to try registering the device succeeds; subsequent attempts to register the device fail. If Program 1 registers a device, then Program 2 is not able to register it. Devices that fail to register can often be found in the log listed for that particular program that attempted to register it. For Ethernet devices, the IP Table will generally have a status of "NOT REG" due to a previous program registering the device first.

**NOTE:** This behavior does not apply when programs are started and stopped individually. For example, if the programmer stops all programs and restarts Program 10 before Program 1, Program 10 registers the device first.

- There are exceptions to this rule described in the bullet above, as some devices, slots, and ports can be registered by multiple programs. Refer to the following table to determine whether a particular control system slot or port is exclusive (can only be registered by one program) or shareable (can be registered by multiple programs).

**NOTE:** Within the following table, "shareable" means that all programs can access the device. "Exclusive" means only one program can ever register the device. Additional details provided in the **Status** column describe exceptions for how this device is handled by multiple programs.

### Port and Slot Sharing

Slot or Port	Status
Expansion card cages	Shareable
BACnet	Exclusive
Built-in audio slot	Sharable

Slot or Port	Status
Built-in COM port	Slots are sharable but ports are exclusive (For example, Program 1 can register COM 2 while Program 5 registers COM 1, but both programs cannot register COM 2 at once.)
Built-in digital inputs	Slots and ports are shareable
Built-in relays	Slots and ports are shareable
Built-in RF gateway	Slots are sharable but individual devices are exclusive
Built-in system monitor	Sharable
Built-in Versiports	Slots are sharable, ports are sharable only if they have the same configuration

## Run Programs from External Storage

Certain 3-Series control systems are equipped with external storage ports. On system boot or a hardware reset, the control system checks for any programs in external memory (if installed) before checking in internal flash.

To configure running programs from external storage, use the **Compact Flash** function in Crestron Toolbox. For more information, refer to the Crestron Toolbox help file.

# Primary-Secondary Mode

Primary-secondary mode is a network configuration that allows a 3-Series control system to access ports on other Crestron control systems over Ethernet. By attaching a secondary control system to a primary control system, the primary control system can use ports that it may not normally have (I/O, IR, RF, and so forth).

In a primary-secondary environment, the primary control system contains the program that controls all Cresnet and Ethernet devices attached to it. The secondary control system turns off its processing capabilities and behaves like any other Cresnet or Ethernet device. The secondary control system obeys the program running on the primary control system, making its ports available to the primary for control. Only one primary program needs to be written to control multiple secondary systems.

**NOTE:** If a control system needs to be able to communicate with other control systems while running its own program, use the "Intersystem Communications" symbol for peer-to-peer communications between control systems over Ethernet. For more information, refer to the SIMPL or SIMPL# Pro help files.

## Definitions

Depending on the communications capabilities of a control system, it may function as a Cresnet server, an Ethernet primary, or an Ethernet secondary.

**NOTE:** A 3-Series control system can be made secondary to a 3-Series or 4-Series control system. A 3-Series control system cannot be made secondary to a 2-Series control system.

## Cresnet Server

When the control system is in Cresnet server mode (the default mode for 3-Series control systems), it can control Cresnet and Ethernet devices as well as 2-Series control systems operating in the Cresnet client mode. Control systems can function as a Cresnet server and an Ethernet primary simultaneously.

## Ethernet Primary

When the control system is in Ethernet primary mode, it can control Ethernet and Cresnet devices as well as 2-Series, 3-Series, and 4-Series control systems operating in the Ethernet secondary mode.

Control systems can function as an Ethernet primary and a Cresnet server simultaneously.

## Ethernet Secondary

When the control system is in Ethernet secondary mode, it operates as an Ethernet device and makes its ports (except for the LAN port) available to a primary control system. Any program loaded into the control system does not run when it is in Ethernet secondary mode.

**NOTE:** 3-Series and 4-Series cage cards are not supported when the control system is in Ethernet secondary mode.

## Primary-Secondary Configuration

Use the following console commands to configure primary-secondary mode parameters for the control system.

### Add Primary Entry

To add a primary entry to the IP table, use the `addmaster` command.

- **Syntax:** `addmaster [cipid] -[ip_address/name]`
  - `cipid` - The ID of the CIP node (in hexadecimal format)
  - `ip_address/name` - The IP address (in dot decimal notation) or the name of the site for DNS lookup
- **Example:** `addmaster 1E -PRO3-IH`

### Remove Primary Entry

To remove a primary entry from the IP table, use the `remmaster` command.

- **Syntax:** `remmaster [cipid] -[ip_address/name]`
  - `cipid` - The ID of the CIP node (in hexadecimal format)
  - `ip_address/name` - The IP address (in dot decimal notation) or the name of the site for DNS lookup
- **Example:** `remmaster 1E -PRO3-IH`

### View Primary IP Table

To view the primary IP table, use the `miptable` command. Append `[-t]` to the command to output the results in table format.

### View Secondary Status

To display the status of the secondary processor, use the `slavestatus` command.

## Response Reject Count for Secondary Connection

To set the default response reject count for the secondary processor connections, issue the `ethslvconnfnct` command:

- **Syntax:** `ethslvconnfnct [connectfailedcount]`
  - `connectfailedcount` - Sets the default secondary connect response reject count. The secondary stops connecting after this number of connect response rejections.
  - No parameter - Displays the current response reject count
- **Example:** `ethslvconnfnct 6`

The `ethslvconnfnct` command sets the number of response rejections that must occur following unsuccessful connection attempts before the secondary reverts back to its normal operating mode. The default count is 100, with each count occurring every 10 seconds.

For example, if the count is set to "6", the secondary would revert back to normal roughly a minute after the first response rejection.

This command can be used in a scenario where a particular IP address is active but does not have a program that listens to that ID.

## Secondary Connection Timeout

To set the default timeout setting for secondary connection, use the `ethslvconntimeout` command:

- **Syntax:** `ethslvconntimeout [timeoutinsec]`
  - `timeoutinsec` - Sets the duration (in seconds) before the secondary connection times out and the secondary returns to its normal operating mode
  - No parameter - Displays the current timeout settings
- **Example:** `ethslvconntimeout 120`

The `ethslvconntimeout` command sets the timeout duration before the secondary reverts back to its normal operation mode if a TCP/IP connection to the primary cannot be established.

This command can be used in a scenario where a secondary attempts to connect to a nonexistent IP address or to a primary that is not running.

# Functional Behavior

Observe the following regarding functional behavior for primary-secondary mode:

- When operating in Ethernet secondary mode, the control system can address any installed hardware, but it cannot address Ethernet devices.
- A 3-Series server and 3-Series client each have their own independent Cresnet bus, allowing the server to assign Cresnet IDs 03 to FE and the client to issue Cresnet IDs 03 to FE, ultimately doubling the number of devices in the network.
- A secondary 3-Series control system can only be configured to operate as an Ethernet secondary to another 3-Series or 4-Series control system; Cresnet client mode is not supported. A 2-Series control system can operate as a Cresnet client or Ethernet secondary to a 3-Series control system.

**NOTE:** There can only be one primary IP table entry.

- A 3-Series control system can switch back and forth between secondary mode and normal mode (running registered user programs) without requiring a reboot. Parameters may be set using the `ethslvconnfct` command that determine how long the secondary controller tries to connect to the primary before reverting back to its normal operating mode.

The following behavior is dependent on whether a primary IP table entry exists when booting the system:

- No primary IP table entry is present when booting the system.
  - Adding a primary IP table entry enables the secondary to start connecting to the primary. Once the secondary is connected, all user programs stop executing, and the device enters into secondary mode.
  - Stopping the program on the primary does not force the secondary back into its normal operating mode. The secondary tries to connect for a set duration before reverting back to normal operations. After reverting, the secondary does not go back into secondary mode until the control system reboots or the primary IP table entry is added again.

**NOTE:** An Ethernet Slave - Reached max count of connect responses being rejected by master before a successful connect. Not retrying - Initiating normal behavior error is logged if this condition occurs.

- Removing the primary IP table entry forces the secondary control system to revert back to normal operations.
- A primary IP entry is present when booting the system.
  - The secondary tries to connect to the primary for a set duration.
  - If the secondary connects successfully, it enters secondary mode.

- If the secondary does not connect successfully, it enters its normal operating mode. The secondary does not go back to secondary mode until the control system reboots or the primary IP table entry is added again.

# Auto Update Mechanism

3-Series control systems provide an automatic update mechanism that centralizes updates on a remote server and allows devices to automatically download and update their respective components when updates are available. The centralized location can be a web server, a MyCrestron.com portal, or a private server.

Through the auto update mechanism, a control system can be scheduled to automatically download updates and can also trigger automatic updates for every device that it controls. The auto update mechanism will also periodically provide feedback to the user, including estimated time for updates, any failures, and successful updates.

## Configure the Auto Update Mechanism

To enable the auto update mechanism for the control system:

**NOTE:** This procedure assumes that a manifest file has already been created and is stored in a centralized location. For more information about the manifest file, refer to [Manifest File on page 52](#).

1. Issue the `auenable on` command.
2. Issue the `aumanifesturl [url]` command to set the URL for the remote manifest file, where `url` is the URL of the manifest file in the following format:  

```
[http or sftp]://username:password@[hostname or ip]:port/path/file
```
3. Issue the `aupollinterval [interval_in_minutes]` command to set how often the control system polls the update server for updates.

**NOTE:** Control systems round up to the nearest hour.

4. Issue the `autime [time]` command to set the day of the week and time when the update manifest file should be read, where `TIME` is formatted as `[day_of_week]hh:mm`
  - `day_of_week`: Day of the week when the manifest file should be read
  - `hh:mm`: 24-hour time when the manifest file should be read
  - Specifying only `hh:mm` will make the update run every day at that time. For example, inputting `autime 11:00` will make the update run at 11:00 AM every day.
  - Specifying `day_of_week` with `hh:mm` will make the update run only at that day and time each week. For example, inputting `autime friday 11:00` will make the update run at 11:00 AM every Friday.
5. Issue the `auusername [username]` command to set a username for downloading automatic updates.
6. Issue the `aupassword [password]` command to set a password for downloading automatic updates.

# Manifest File

The manifest file, which is JSON encoded, contains all the information required for scheduling and performing automatic updates on a 3-Series control system. The control system parses the manifest file, locates any required updates, and performs them in the order listed in the file. The manifest is always downloaded and parsed either at a scheduled time or from a forced action, which can be set from console commands.

The following shows an example of a manifest file JSON format:

```
[
  {
    "ControllerHostName": "ControllerHostname",
    "deviceHostname": "DeviceToUpdateHostname",
    "deviceModel": "Prompt",
    "deviceId": "XX",
    "indirectCresnetId": "XX",
    "controlSystemHostname": "ControlSystemHostName",
    "logFolder": "URL",
    "fileToUpdate": [
      {
        "fileUrl": "URL",
        "fileHashUrl": "URL",
        "fileType": "ACTION",
        "whenToDownload": "TIME",
        "whenToApplyUpdate": "TIME",
      }
    ]
  }
]
```

## Overview of Manifest Parameters

The control system uses the following top-level parameters to determine which associated actions apply to it in order to initiate the auto update mechanism. At least one of these parameters must be defined, and they may contain wildcards for partial matching. A control system matches all specific values before taking the associated action.

**NOTE:** All keywords are case insensitive.

- **ControllerHostName** - The hostname of the control system. Indicates that the following schema is meant for a control system. All non-control systems continue to parse the schema and act upon it if the details match.
- **deviceHostname** - The hostname or IP address of an Ethernet device. This field will have different meanings depending on the operation to be performed (refer to [Description of Manifest Parameters on page 54](#)). If this parameter is omitted, the value will default to any.

- `deviceModel` - The device type of the target client and the physical transport (refer to [Description of Manifest Parameters on page 54](#)). If this parameter is omitted, the value will default to `any`.
- `controlSystemHostname` - The hostname or IP address of a control system that is located in the client's IP table. If this parameter is omitted, the value will default to `any`.
- `deviceId` - The IP ID, Cresnet ID, or RF ID of the device (in hexadecimal format). This parameter can be displayed as a range (AA-ZZ) and/or as a list separated by commas (A,B,...Z).
- `Path` - Locates the path of a specific, generic device (without using a hostname) within a system. Each path node consists of the following:
  - A letter identifying the specific subnet where the device or parent device(s) is located
    - C - Cresnet
    - E - Ethernet
    - R - RF
    - S - Slot
  - The subnet ID of the parent device

**NOTE:** The ID of the target device will always be specified in the device ID field and not the path. Therefore, the last path node will only consist of a subnet letter.

- The path may consist of several parent devices. In this case, each element is delimited by a colon.
- **Examples:**
  - A control system with an Ethernet to Cresnet bridge (at IP ID 03) that contains an EX gateway on Cresnet leg 1 (at ID A0) with a dim switch (at RFID 04) would have a path of "E03:C1.A0:R" and a device ID property of "04."
  - An EX gateway located on Ethernet (at ID 05) has a path of "E" and a device ID of "05."

The following top-level parameter is optional and is not necessary to complete the auto-update process:

- `logFolder` - Each control system will create a result log file in this folder for each action. The URL must be an SFTP location. For more information, refer to [Results File on page 60](#).

The actions to be taken by the client are defined by the `filesToUpdate` array:

- `filesToUpdate` - The files associated with the action will be downloaded and installed in the order they are listed. The following parameters are used to describe each entry in the array and are required:
  - `fileUrl` - The URL of the file to be downloaded. The control system must support parsing a full URL and downloads over HTTPS and SFTP: `[http or sftp]://username:password@[hostname or ip]:port/path/file`.

- `fileHashUrl` - The URL of the hash file. This hash is downloaded first and then compared to the previous hash that is cached locally on the control system.
  - If the hash in this file has changed, then the file defined in `fileUrl` will be downloaded and installed.
  - If the has in this file has not changed, then the file defined in `fileUrl` will not be downloaded and installed.
  - If no local hash cache exists, then the file will be downloaded.
- The client must support parsing a full URL and downloads over HTTPS and SFTP: `[http or sftp]://username:password@[hostname or ip]:port/path/file`.
- `fileType` - Determines how the control system will handle the file once it is downloaded. The value of this parameter must be one of the following predefined types; undefined types are ignored by the client:
  - `"firmware"` - Firmware update file (.puf or .zip)
  - `"project"` - Standard VTZ project file
  - `"config"` - Text file containing a list of console commands to be executed
  - `"UserProgram"` - Indicates a user program

## Description of Manifest Parameters

### ControllerHostName

This parameter indicates the hostname of the control system. The controller must support wildcard characters "\*" and "?", where "\*" matches x number of characters and "?" matches exactly one character. A "\*" indicates that all controllers will act upon a manifest schema, while a partial match can be used to limit the schema to a subset (for example, PRO-??).

Using these wildcard characters, the same manifest can be used across multiple control systems, which means that all control systems will try to update the devices specified based on the details in the manifest. While this method is recommended for Cresnet devices and devices on an internal gateway (since it is controller-specific), it could cause issues with Ethernet devices, as they can be accessed from multiple control systems.

Refer to the following best practices to manage the system properly:

- Cresnet and EX devices on the internal gateway can be safely updated when using "wildcards" for the controller hostname.
- If updating Ethernet devices, it is recommended to specify device IDs so that the controller can be selective when updating.
  - The statement above assumes that the system is configured and that the IP table is set up correctly on the Ethernet devices. It also assumes that the devices follow the auto-discovery specification.
  - The above statement might not work for a first-time setup.
- If updating peripherals connected to an Ethernet-connected gateway or bridge, the user is responsible for managing the manifest file.

## Control System Update Parameters

On a control system, a combination of "deviceId" and "deviceModel" indicates what device needs to get updated. The "deviceModel" always needs to be specified, since this parameter indicates the class of device and the corresponding plugin that handles the update. If "deviceId" is set to "any", all devices of the type specified by "deviceModel" will be updated.

### DeviceId

Multiple devices that share the same type are often connected to the same control system. In this case, all of these devices can be updated using the "any" keyword. A single device can be updated using its specific device ID. If multiple device IDs need to be updated, multiple sections must be created in the manifest. To support this schema, the control system accepts a range and/or a list of the devices, which can be specified as follows:

- **List:** A comma separated list (3,4,78,95)
- **Range:** A continuous range of device IDs (5-15)
- **List and Range:** A combination of the above two methods (3,73,25, A-1E)

**NOTE:** The device IDs must be in hexadecimal format with no preceding "0x".

## DeviceHostname

This parameter is only valid for an Ethernet device and can have multiple meanings:

- If the "deviceModel" specifies an Ethernet device, then the "deviceHostname" identifies the device to be updated. If the "deviceHostname" is the wildcard character "\*", then it identifies all the Ethernet devices of the type specified by "deviceModel".
- For any combination of "deviceModel" and "deviceId", it needs to be established whether the device to be updated is on the controller or on a peripheral connected to the controller:
  - If the "deviceHostname" is defined, then the device to be updated is connected to an Ethernet-connected parent device.
  - If the "deviceHostname" is not defined, then the device to be updated is connected to the controller (Cresnet device/internal gateway).
  - If the "deviceHostname" is "\*", then all devices will be updated as defined by the "deviceModel" parameter in one of the following three ways:
    - All gateways will be updated
    - All specific devices connected to internal/external gateways will be updated
    - All Cresnet devices of a specified type connected to a CEN-CN or all EX devices connected to an external Ethernet gateway will be updated.

## IndirectCresnetId

This parameter is only valid when the "deviceModel" specifies an EX device:

- If the "deviceModel" specifies an EX device, then it indicates the Cresnet ID of the gateway.
- If "indirectCresnetId" is "\*", then it indicates the specific device connected to any Cresnet gateway.
- If the value of this parameter is not defined, then the device to be updated is connected to the controller (Ethernet device/internal gateway).

## UserPrograms

This parameter indicates that a user program should be updated. The following keywords are defined for this parameter:

- "programNumber" - Valid ranges are 1–10 or 1 only, and the range depends on the controller to be updated. The Program ID tag is intended for interactive systems only. The default value is "1".
- "programLocation" - Indicates whether the program needs to be loaded onto internal or external memory. The default value is "internal". For external memory, the controller loops through RM and RM2.

## Sample Manifest File

The following shows a complete example of a JSON-encoded manifest file for a control system with multiple components. Note that actions are performed in the order they appear on the file.

```
[
  {
    "deviceHostname": "Room101-panel",
    "deviceModel": "TSW-770",
    "deviceId": "03",
    "controlSystemHostname": "192.186.1.1",
    "logFolder": "sftp://xxabet-xx2/html/Office-TouchPanel/results",
    "fileToUpdate": [
      {
        "fileUrl": "sftp://xxabet-xx2/html/Office-TouchPanel/xx.puf",
        "fileHashUrl": "sftp://xxabet-xx2/html/Office-TouchPanel/firmwareHash.txt",
        "fileType": "firmware",
        "whenToDownload": "Friday 23:00",
        "whenToApplyUpdate": "now",
      }
    ]
  }
]
{
  "deviceHostname": "Room101-panel",
  "deviceModel": "TSW-770",
  "logFolder": "sftp://xxabet-xx2/html/Office-TouchPanel/results",
  "fileToUpdate": [
    {
      "fileUrl": "sftp://xxabet-xx2/html/Office-TouchPanel/xx.ini",
      "fileHashUrl": "sftp://xxabet-xx2/html/Office-TouchPanel/configHash.txt",
      "fileType": "config",
      "whenToDownload": "Sunday 2:00",
    }
    {
      "fileUrl": "sftp://xxabet-xx2/html/Office-TouchPanel/xx.vtz",
      "fileHashUrl": "sftp://xxabet-xx2/html/Office-TouchPanel/projectHash.txt",
      "fileType": "project",
      "whenToDownload": "Sunday 2:00",
    }
  ]
}
{
  "ControllerHostName": "Crestron-PR04",
```

```

"deviceModel": "PRO4",
"logFolder": "sftp://xxabet-xx2/html/Office-PRO4/results",
"fileToUpdate": [
  {
    "fileUrl": "sftp://xxabet-xx2/html/Office-PRO4/program1.lpz",
    "fileHashUrl": "sftp://xxabet-xx2/html/Office-PRO4/UpdaterHash.txt",
    "fileType": "userProgram",
    "ProgramNumber": "5",
    "ProgramLocation": "Internal",
    "whenToDownload": "Sunday 2:00",
  }
]
}
{
"ControllerHostName": "Crestron-PRO4",
"deviceHostname": "Crestron-EXGateway",
"deviceModel": "CEN-GWEXER",
"logFolder": "sftp://xxabet-xx2/html/Office-PRO4/results",
"fileToUpdate": [
  {
    "fileUrl": "sftp://xxabet-xx2/html/Office-PRO4/GatewayFirmware.zip",
    "fileHashUrl": "sftp://xxabet-xx2/html/Office-PRO4/GatewayHash.txt",
    "fileType": "firmware",
    "whenToDownload": "Sunday 2:00",
  }
]
}
{
"ControllerHostName": "Crestron-PRO4",
"deviceHostname": "Crestron-EXGateway",
"deviceId": "10",
"deviceModel": "CLW-LDIMEX",
"logFolder": "sftp://xxabet-xx2/html/Office-PRO4/results",
"fileToUpdate": [
  {
    "fileUrl": "sftp://xxabet-xx2/html/Office-PRO4/CLWFirmware.zip",
    "fileHashUrl": "sftp://xxabet-xx2/html/Office-PRO4/CLWHash.txt",
    "fileType": "firmware",
    "whenToDownload": "Sunday 2:00",
  }
]
}
{
"ControllerHostName": "Crestron-PRO4",
"deviceId": "11",
"deviceModel": "CSM-QMT50-DCCN",
"logFolder": "sftp://xxabet-xx2/html/Office-PRO4/results",
"fileToUpdate": [
  {
    "fileUrl": "sftp://xxabet-xx2/html/Office-PRO4/ShadeFirmware.zip",
    "fileHashUrl": "sftp://xxabet-xx2/html/Office-PRO4/ShadeFirmwareHash.txt",
    "fileType": "firmware",
    "whenToDownload": "Sunday 2:00",
  }
]
}
]

```

## Manifest File Code Flow

The code flow of the auto update mechanism using the manifest file, as performed by the control system, is described below:

1. The control system downloads the manifest file at the predetermined day and time.
2. The control system parses the manifest file for applicable actions.
3. The control system verifies that the action is valid for itself and its current state. If not, the control system skips to step 8.
4. The control system downloads the hash file associated with the given action. If the hash file matches the hash cached on the control system, the control system skips to step 8.
5. The control system downloads the update file.
6. The control system applies the update file as directed. Retries are defined in the [Error Handling on page 62](#).
7. The control system updates the local copy of the hash if the update is successful.
8. If more actions are defined in the manifest, the control system returns to step 3.
9. The mechanism ceases once all actions defined in the manifest are completed.

# Results File

The result of each action taken by the control system is uploaded to the location specified by the "logFolder" parameter that is associated with the action. Results filenames have the following syntax:

- [MAC\_address\_of\_CS].[timestamp].[index].log
- **Example:** 00107f44901e.20190321\_115636.1.log

When an action is performed, a result file is uploaded for each part of the action: When the action is downloaded, when the action starts, and when the action finishes. The timestamp is fixed at the initial step of the action and is expressed in "yyyyMMdd\_HHmms" format.

It is the responsibility of the administrator to manage purging and storing these files as necessary. Using the 3-Series control system, this task can be accomplished via an automatic job that rotates the results files.

If any locations are inaccessible to the control system (for example, a downed server), then failure results are recorded in the client's error log. If the results location is accessible, a failure result is indicated in the results file and is uploaded to the results location.

## Description of Results File Parameters

Use the following parameters inside the result files to identify the control system, the action taken, and the results of the action. The following parameters are always included in the result file:

- `clientModel` - The device type of the control system
- `updateAction` - The type of action taken by the control system
- `updateLog` - Indicates the section below that contains the actual log information for the preceding action
  - `fileName` - The name of the file that was downloaded
  - `fileHash` - The hash value associated with the file that was downloaded
  - `whenWasDownloaded` - The time the action started and the initial file was downloaded (in "HH:mm:ss" format)
  - `whenWasApplied` - The time when the update was applied (in "yyyyMMdd\_HHmms" format)
  - `whenWasCompleted` - The time when the update was completed (in "yyyyMMdd\_HHmms" format)
  - `result` - The result of the action, which is one of the following:
    - `success` - The action succeeded.
    - `error` - The action failed. See [Error Handling on page 62](#).
  - `estActionDuration` - The estimated time to complete the action (in seconds) where applicable

- `errorCode` - The error code number (if applicable)
- `errorText` - Any text that is associated with the error condition (if applicable)

The following parameters are dependent on the type of action taken:

- `numberOfSubActions` - The number of sub actions that are associated with the main action taken by the client.
- `subActions` - A sequence of sub action results that are associated with the main action taken by the client.
  - `command` - The console command executed by the client
  - `commandResult` - The result text returned to the console when executing a console command

The following is an example of a results file for a control system automatic update:

```
{
  "ControllerHostName": "Crestron-PR04",
  "deviceModel": "PR04",
  "updateAction": "firmware",
  "updateLog": [
    {
      "fileName": "xx.txt",
      "fileHash": "HASHCODE",
      "whenWasDownloaded": "2020-08-29T23:05:10Z",
      "whenWasApplied": "2020-08-29T23:15:10Z",
      "whenWasCompleted": "2020-08-29T23:25:10Z",
      "numberOfSubActions": "2",
      "subActions": [
        {
          "component": "eboot",
          "result": "success",
          "PreviousVersion": "1.001.0033",
          "CurrentVersion": "1.001.0034",
        }
        {
          "component": "updater",
          "result": "success",
          "PreviousVersion": "1.02.35",
          "CurrentVersion": "2.001.0034",
        }
      ]
    }
  ]
}
```

# Error Handling

The following errors may be encountered during the auto-update process. If the below solutions do not resolve the error, contact Crestron True Blue support via phone, email, or chat as described at [www.crestron.com/Support](http://www.crestron.com/Support).

- **File download fails:** the client is unable to download the hash file or the actual component
  - Retry the download on the next polling interval.
  - If download still fails, delete the hash file (The hash file was downloaded but not the actual component to the update).
- **Update operations:** The unzipping operation fails or the firmware component did not get updated.
  - Retry the operation.
- **Cannot resolve hostname:** The client cannot resolve the hostname.
  - Check that the target hostname is valid, and then retry resolving the hostname on the next polling interval.
- **Cannot connect to the server/device:** The client cannot connect to the server or the device.
  - Check that the server or the device is currently accessible, and then retry connecting to the server or the device on the next polling interval.

A control system can handle errors in different ways depending on the component being updated and the dependencies between the components. The control system will do one of the following:

- **Abort the current operation.**

For example, if the control system is updating a .puf component, which has dependencies defined, and the first part of the update fails, the control system will abort this particular component and will continue to the next item in the manifest file.
- **Log the error and continue on to the next step.**

For example, if the control system is updating EX devices with the type "CLW-LDIMEX" and one device update fails, the controller will continue with the next set of devices.

# Connect to XiO Cloud Service

The [XiO Cloud® service](#) allows supported devices across an enterprise to be managed and configured from one central and secure location in the cloud. Supported Crestron® devices are configured to connect to the service out of the box.

Use of the service requires a registered XiO Cloud account. To register for an XiO Cloud account, refer to <https://www.crestron.com/Support/Tools/Licensing-Registration/XiO-Account-Registration>.

**NOTE:** The device may be disconnected from the XiO Cloud service by navigating to the **Cloud Services** tab in Crestron Toolbox™ software (**Functions > Device Info > Cloud Services**). For details, refer to the Crestron Toolbox help file.

To connect the device to the XiO Cloud service:

1. Record the MAC address and serial number that are labeled on the shipping box or the device. The MAC address and serial number are required to add the device to the XiO Cloud service.

**NOTE:** If the device has multiple MAC addresses, use the MAC address that is providing the primary connection back to the network. For most devices, the Ethernet MAC address should be used. However, if your device is connecting to the network over a different protocol (such as Wi-Fi® communications), use the MAC address for that protocol instead.

2. Log in to your XiO Cloud account at [portal.crestron.io](https://portal.crestron.io).
3. Claim the device to the XiO Cloud service as described in the [XiO Cloud User Guide](#).

Select the device from the cloud interface to view its status and settings. The device may now also be managed and assigned to a group or room. For more information, refer to the [XiO Cloud User Guide](#).

**NOTE:** For XiO Cloud accounts with room-based licenses, the device must be added to a licensed room before its status and settings can be viewed.

# Appendix A: Restore to Factory Defaults

If the control system is no longer communicating via USB or Ethernet, use the following procedure to restore the device to its factory default settings.

**NOTE:** This procedure erases the control system's firmware and reinstalls it. If problems persist before a SIMPL program is loaded, contact Crestron True Blue support via phone, email, or chat as described at [www.crestron.com/Support](http://www.crestron.com/Support). If the system locks up after a SIMPL or SIMPL# Pro program is loaded, there is likely an issue with the program.

1. Use a small, pointed object (such as the tip of a pen) to press and release the **HW-R** button on the front of the control system.
2. Use a small, pointed object (such as the tip of a pen) to quickly press the **SW-R** button on the front of the control system five times, with under a 1-second gap between each press.
3. Wait up to 5–10 minutes for the self-recovery process to complete.
4. Attempt to make a connection to the control system over USB or Ethernet.
5. Once the device has been discovered, use the Text Console tool in Crestron Toolbox to check for a prompt. The standard device prompt should display.

**NOTE:** Repeat steps 1–5 if the first attempt does not correct the issue. If the control system is still unresponsive, contact [Crestron True Blue support](#) for assistance.

6. The restore process may enable SSL (Secure Sockets Layer) on the control system. After communication returns following the restore, issue the `ssl off` command using the Text Console tool to disable SSL.

**NOTE:** If a connection cannot be established using the Text Console tool, change the connection type from **Auto Detect** to **SSL** in the **Edit Connections** dialogue.

7. Reload the control system firmware using the **Package Update Tool** in Crestron Toolbox.

If the control system is still communicating with Crestron Toolbox via USB or Ethernet, or if the `initialize` command was issued to the CP3-R as part of a troubleshooting procedure, issue the `restore` command using the Text Console tool, and then follow the post-restore process (steps 6–7 in the above procedure).

# Appendix B: Port Forwarding

Port forwarding can be used to provide connections from outside the local network for mobile and browser applications.

Observe the following points about port forwarding:

- Remap the external ports from the initial defaults. Remapping the external ports minimizes the number attempts that are allowed to access the system. A hacker will be unable to scan well-known ports for entry, and must instead scan all ports and then determine what protocols are supported before attempting to log in to the system.
- Most home routers allow different external and internal ports to be set. An example of a home router setup page is provided below.

## Home Router Setup Page Example

Single Port Forwarding		External Port	Internal Port	Protocol	To IP Address	Enabled
Application Name	None	---	---	---	192 . 168 . 194 . 0	<input type="checkbox"/>
	None	---	---	---	192 . 168 . 194 . 0	<input type="checkbox"/>
	None	---	---	---	192 . 168 . 194 . 0	<input type="checkbox"/>
	None	---	---	---	192 . 168 . 194 . 0	<input type="checkbox"/>
	None	---	---	---	192 . 168 . 194 . 0	<input type="checkbox"/>
CIP		9699	41796	Both	192 . 168 . 194 . 99	<input checked="" type="checkbox"/>
SSH		2299	22	Both	192 . 168 . 194 . 99	<input checked="" type="checkbox"/>
eControl		4499	443	Both	192 . 168 . 194 . 99	<input checked="" type="checkbox"/>
Policy File		843	843	Both	192 . 168 . 194 . 99	<input checked="" type="checkbox"/>

- Use external port numbers that are not commonly used. The actual number is not important; it simply must match the entry in the mobile app configuration.
- Note the exception on the policy file support. If the XPanel web browser is used, open port 843 under **External Port**.
- Open ports that are required only. For example, if mobile applications or XPanel applications are used, open only the secure CIP port (default is 41796) and HTTPS port (default is 443). Ensure that SSL settings are used in the mobile application.
- If XPanel browser support is required, the unsecured CIP port (default is 41794) must be used. The system is still secured because the user is prompted to enter his or her credentials prior to running the project. The XPanel browser required port 843 to be routed to the system.
- If ports 41795 or 41797 were opened for external use, reroute the external ports to port 22 and use the SSH console.

# Appendix C: CP3-GV Feature Set

The CP3-GV is a version of the CP3 that is targeted towards high security network environments. It was developed with consideration for IA Requirements and received a DIACAP scorecard.

To assure a high level of security, the following changes were made compared to the CP3:

- To ensure that communication with the control system is not vulnerable to network traffic sniffing, SSL has been enabled by default, and cannot be turned off. This also means that only the secure console is available.
- The built-in FTP server is disabled because it sends authentication over clear text.
- Authentication is enabled and cannot be turned off.
- Password rules are set to require secure passwords; these rules cannot be changed.
- To ensure all traffic in and out of the control system is secure:
  - TCP Clients/Servers are disabled
  - UDP is disabled
  - EISC is disabled
  - Direct sockets and SIMPL# Sockets are disabled
- The CP3-GV will only allow connections from devices that support the secure CIP Protocol.
- To protect the device from being discovered on the network, Autodiscovery is disabled.
- To protect against web-based attacks, Webserver is disabled.

**NOTE:** Firmware on the CP3-GV cannot be updated. As a result, the following software versions are required in order to program:

- **Crestron Database:** 35.06.004.00
- **Device Database:** 46.05.007.00
- **SIMPL:** 3.11.15

These software versions are from November 2012. They will not include newer products (such as the DM NVX® A/V Encoders and Decoders). Install these versions and review the **Configure View** in SIMPL for a list of compatible products.

