EST3
Programming Manual

P/N 270381 • Rev 3.0 • 21OCT99

DEVELOPED BY	Edwards Systems Technology 6411 Parkland Drive Sarasota, FL 34243 (941) 739–4300
COPYRIGHT NOTICE	
TRADEMARKS	
CREDITS	This manual was designed and written by EST Technical Services-Documentation Department, Sarasota.

DOCUMENT HISTORY

Revision	Date	Reason For Change
1.0	18JUL96	Initial release.
1.1–1.4	N/A	Documentation development releases.
1.5	9MAY97	Update manual to coincide with release of software version 1.0.
1.6–1.9	N/A	Documentation development releases.
2.0	14DEC98	Update manual to coincide with release of software version 1.3. Added chapter 15. Major revisions made to chapters 3, 6, and 9. Updated various screen captures throughout.
3.0	21OCT99	Update manual to to coincide with release of software version 1.5. Added glossary and removed chapters 4-15. The 3-SDU help file, revision 1.5, includes the user information formerly contained in chapters 4-15.

Content

Important information

Limitation of liability

This product has been designed to meet the requirements of NFPA Standard 72, 1996 Edition; Underwriters Laboratories, Inc., Standard 864, 7th Edition; and Underwriters Laboratories of Canada, Inc., Standard ULC S527. Installation in accordance with this manual, applicable codes, and the instructions of the Authority Having Jurisdiction is mandatory. EST shall not under any circumstances be liable for any incidental or consequential damages arising from loss of property or other damages or losses owing to the failure of EST products beyond the cost of repair or replacement of any defective products. EST reserves the right to make product improvements and change product specifications at any time.

While every precaution has been taken during the preparation of this manual to ensure the accuracy of its contents, EST assumes no responsibility for errors or omissions.

FCC warning

This equipment can generate and radiate radio frequency energy. If this equipment is not installed in accordance with this manual, it may cause interference to radio communications. This equipment has been tested and found to comply within the limits for Class A computing devices pursuant to Subpart B of Part 15 of the FCC Rules. These rules are designed to provide reasonable protection against such interference when this equipment is operated in a commercial environment. Operation of this equipment is likely to cause interference, in which case the user at his own expense, will be required to take whatever measures may be required to correct the interference.

About this manual

This manual provides reference information to support the system programming function.

Intended audience

This manual and the information it contains is intended to be used by persons who have working knowledge of Windows and have successfully completed:

- The EST3 Self Study Course
- The EST3 Programming and Application Course.

Organization

This manual is organized as described below:

Chapter 1 provides a general overview of basic concepts that the system programmer should understand before attempting to program system functions.

Chapter 2 provides an alphabetical reference of the input event types used in the programming language.

Chapter 3 provides an alphabetical reference of the output commands used in the programming language.

Appendix A provides a quick reference to the information contained in chapters 2 and 3.

The EST3 library

A family of documents and multi-media presentations supports the EST3 network. A brief description of each document is provided below.

EST3 Installation Manual and Service Manual, *P/N 270380.* This manual provides complete information on how to install and service the EST3 hardware. This manual also includes installation information on selected Signature Series components.

EST3 Programming Manual, P/N 270381. This manual provides quick reference information for defining and labeling individual system components using the Systems Definition Utility (SDU), and for writing rules to govern system operation.

EST3 System Operations Manual, P/N 270382. This manual provides detailed information on how to operate the system and system components.

EST3 International Installation Supplement Manual, P/N 270925. This manual provides information specific to systems installed outside the United States and Canada.

EST3 Smoke Management Application Manual, P/N 270913. This manual provides information for designing, programming, and testing an EST3 smoke control system.

EST3 Users Self-Study Course, P/N 270684. This course contains a self-paced manual, and accompanying video. The course is designed for building personal, security guards, firefighters, and similar individuals that may be required to operate the system.

Signature Series Intelligent Smoke and Heat Detectors Applications Bulletin, *P/N 270145.* This manual provides additional applications information on the Signature series smoke and heat detector applications.

Signature Series Component Installation Manual, P/N 270497. This manual provides detailed mounting and wiring information for all Signature series devices.

Speaker Application Guide, P/N 85000-0033. This manual provides information on the placement and layout of speakers for fire alarm signaling and emergency voice communications.

Strobe Applications Guide*, P/N 85000-0049*. This manual provides information on the placement and layout of strobes for fire alarm signalings.

Related documentation

National Fire Protection Association (NFPA)
1 Batterymarch Park
P.O. Box 9101
Quincy, MA 02269-9101

NFPA 70	National Electric Code
NFPA 72	National Fire Alarm Code

Underwriters Laboratories Inc. (ULI)

333 Pfingsten Road
Northbrook, IL 60062-2096

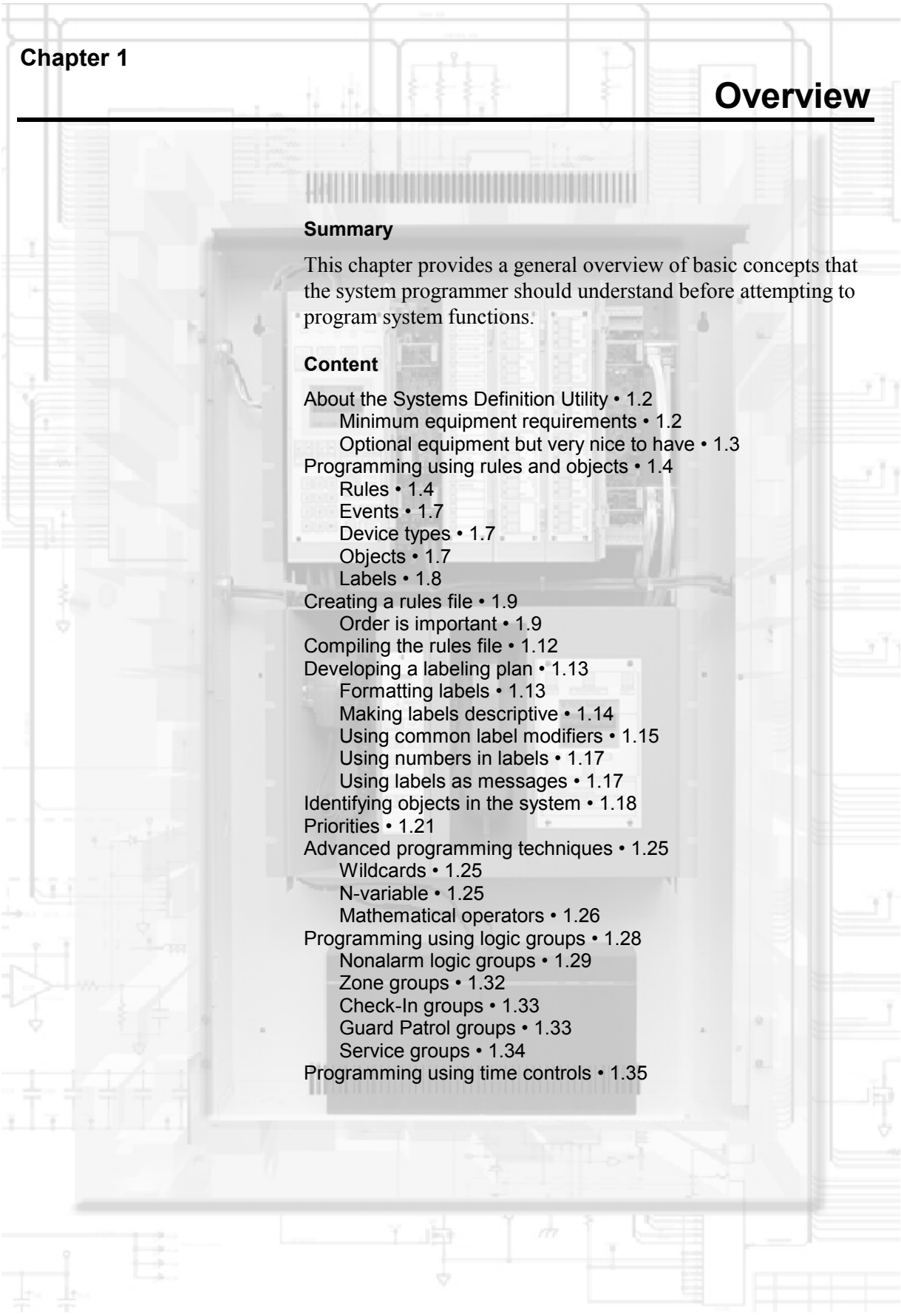
UL 38	Manually Actuated Signaling Boxes
UL217	Smoke Detectors, Single & Multiple Station
UL 228	Door Closers/Holders for Fire Protective Signaling Systems
UL 268	Smoke Detectors for Fire Protective Signaling Systems
UL 268A	Smoke Detectors for Duct Applications
UL 346	Waterflow Indicators for Fire Protective Signaling Systems
UL 464	Audible Signaling Appliances
UL 521	Heat Detectors for Fire Protective Signaling Systems
UL 864	Standard for Control Units for Fire Protective Signaling Systems
UL 1481	Power Supplies for Fire Protective Signaling Systems
UL 1638	Visual Signaling Appliances
UL 1971	Visual Signaling Appliances

Underwriters Laboratories of Canada (ULC)
7 Crouse Road
Scarborough, Ontario M1R 3A9

ULC S527	Standard for Control Units for Fire Alarm Systems
ULC S524	Standard for the Installation of Fire Alarm Systems
ULC S536	Standard for the Inspection and Testing of Fire Alarm Systems
ULC S537	Standard for the Verification of Fire Alarm Systems
PLUS	Requirements of state and local building codes.
	Requirements of the Authority Having Jurisdiction.

Chapter 1

Overview



Summary

This chapter provides a general overview of basic concepts that the system programmer should understand before attempting to program system functions.

Content

About the Systems Definition Utility

The Systems Definition Utility (SDU) is a database application used for setting up and programming an EST3 life safety system. Using the SDU you can:

- Build setup files using forms to specify system the hardware configuration and operating options for a given project
- Record audio messages to create an automated voice messaging system
- Create extensive system controls using advanced rules-based programming.

Minimum equipment requirements

Before installing the Systems Definition Utility, you should make sure your computer system meets the following minimum equipment requirements:

- IBM-compatible computer with Pentium-class micropocessor
- One or more serial communications (COM) ports for connecting a bar code reader or a download cable
- One parallel printer port (LPT)
- Hard disk drive with at least 40 megabytes or more free disk space
- 32 MB of Random Access Memory (RAM)
- One 3.5-inch floppy drive
- One 2X or faster CD-ROM drive
- SVGA color display, 800x600, 256-color resolution
- Microsoft Mouse or other compatible pointing device
- Microsoft Windows 95 or MS-DOS 5.0 and Windows 3.1x.

Note: The amount of free disk space required varies with the number of projects and the amount of audio messages you plan to save on the hard drive. A general rule of thumb is to have at least twice the amount of hard disk space required by your largest project.

The Systems Definition Utility executes highly disk-intensive functions. For best results, make sure your computer system is configured to achieve optimal performance. To optimize your system, refer to the documentation that came with your equipment.

Optional equipment but very nice to have

In addition to the basic system described above, you will also need the following optional equipment to make use of some of the advanced features available in the SDU:

- Sound card for configuring audio systems using the 3-ASU
- Bar code reader for configuring Signature data circuits
- 300-dpi laser printer or equivalent for printing reports

The following sound cards are recommended:

Manufacturer	Model
Creative Technology	CT2960 16-bit Audio Card CT03600 16-bit Advanced Wave Table Audio Card
Diamond Sound	Super Sound Origins Lite
Bravo Sound 16P	MT9ATC931
AT&T	Jazz 16 Business Audio
Ultra	Ultra 32 Audio Card

The following bar code readers are recommended:

Manufacturer	Model
Zebra Technologies VTI, Inc.	SCAN•ONE
Zebra Technologies VTI, Inc.	Barcode Anything SCAN 97

Install the bar code reader per the manufacturer's instructions and configure the reader to interpret Interleaved 2 of 5 bar codes.

Programming using rules and objects

System programming is accomplished using a set of rules that determine the output responses for given input events. The rules are written, compiled, and then downloaded into the panel. When an input event occurs (a device goes active), the panel connected to the device searches for the device/event type combination in its response tables and, if found, executes the appropriate output commands.

The most basic fire alarm systems can be programmed using one simple rule: when smoke detector “A” activates; sound horn “B”. As fire alarm systems become more extensive, they require a more sophisticated set of rules to program them properly. Before you begin writing a rule, you should have a thorough understanding of:

- Rules
- Events
- Device types
- Objects
- Labels

Rules

A rule is a programming statement that specifies which commands to execute when a certain event takes place. A rule consists of a label, an input statement, and an output statement or statements.

The basic syntax for a rule is:

```
[Rule_label]
Input_statement:
      Output_statement_1, {comments}
      Output_statement_2, {comments}
      Output_statement_3; {comments}
```

The rule label can be up to 40 characters in length and enclosed in brackets. The rule label can be any ASCII character except: braces “{ }”, the percent symbol “%”, the number symbol “#”, less than and greater than symbols “< >”, and asterisks “*”.

The input statement ends with a colon and the output statement ends with a semicolon. When more than one output statement is used in a rule, each output statement must end with a comma, except for the last output statement, which must end with a semicolon. A rule may contain up to 32 output statements.

When a rule has multiple output statements, each output command will be executed in the order it is listed in the rule (from first to last). When the event activating the rule restores,

the operations performed by the rule will automatically restore in the reverse order (from last to first).

Comments can be placed anywhere in the rule and must be enclosed in braces. Be sure to add comments for later reference and make sure they can be easily seen.

Input statements

An input statement is the part of a rule that determines what must happen before the corresponding output statements will be executed. There are two different syntaxes used in rule input statements depending on the input event type selected. The two input statement syntaxes are:

event_type :

event_type *device_type 'object_label'* :

where:

event_type	Specifies the type of input event required for the rule to execute. When a system input activates, the resulting change in state creates an event. Refer to *Chapter 2:Input event types* for detailed descriptions of event types and their usage.
device_type	Specifies the device type of the input device initiating the event. The device_type parameter is optional when using the 'object_label' parameter. When the device_type parameter is not included, all devices whose label matches 'object_label' will respond to the command.
'object_label'	Specifies the label of the input device or circuit that must go active for the rule to execute. The 'object_label' parameter is optional when using the device_type parameter. When the 'object_label' parameter is not included, all devices with the specified device type will trigger the rule.

Note: When the device_type and 'object_label' parameters are optional you must specify one or the other or both.

An input statement must be valid for the rule to be successfully compiled. This means that the device type of the object identified by 'object_label' must match that specified by device_type. Also, the input event specified by event_type must be applicable for the device_type.

Output statements

An output statement is the part of a rule that determines the commands that will be executed in response to a given input and

in what order. There are ten different syntaxes used in rule output statements depending on the output command selected. The ten output statement syntaxes are:

command ;

command delay_value ;

command 'cabinet_label' ;

command 'routing_label' ;

command *device_type 'object_label'* ;

command *priority 'object_label'* ;

command *priority device_type 'object_label'* ;

command *priority* 'amp_label' to 'channel_label' ;

command 'guard_label' Route route_id ;

command *priority* 'msg_label' *from 'asu_label'* to 'channel_label' ;

Note: When the device_type and 'object_label' parameters are optional you must specify one or the other or both.

where:

command	Specifies the required final state of the output device. Refer to *Chapter 3:Output commands* for a description of output commands and their usage.
priority	Specifies the relative importance this command has over other commands affecting the same output device. Refer to *Priorities* for a description on priority levels and their usage.
device_type	Specifies the device type of the output device responding to the command. The device_type parameter is optional when using the 'object_label' parameter. When the device_type parameter is not included, all devices whose label matches 'object_label' will respond to the command.
delay_value	Specifies the length of a delay in seconds.
'object_label'	Specifies the label of the output device or circuit responding to the command. The 'object_label' parameter is optional when using the device_type parameter. When the 'object_label' parameter is not included, all devices with the specified device type will respond to the command.

'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.
'amp_label'	Specifies the label of an amplifier.
'channel_label'	Specifies the label of an audio channel.
'msg_label'	Specifies the label of a voice message.
'asu_label'	Specifies the label of the audio source unit providing the voice message.
'guard_label'	Specifies the label of the Guard Patrol group.
route_id	Specifies the route number of the guard patrol tour.

Events

An event is the outcome produced by a panel's CPU module when an addressable point on the panel changes state. The information contained in an event includes the logical address of the point that changed state, the event type, and the event message.

Device types

A device type is the classification given to objects created in the database that defines the operating characteristics of the corresponding device. For example, the PULL device type is assigned to objects created for manual pull stations.

Refer to *Appendix A: Quick Reference* for a list of device types and their use.

Objects

An object is a database entity that represents an addressable point in the system, such as rail modules, smoke detectors, switches, and light emitting diodes (LEDs).

Objects can also be:

- Logical groups consisting of other objects
- Voice messages
- Pseudo points designed to monitor card-level and system-level functions

For example, the Signature controller module is an object as are any Signature devices connected to it. In contrast, the 3-IDC8/4 module is an object, each of its eight circuits are objects, but any two-wire smokes connected to the circuits are not.

Labels

A label is a descriptive word or words to identify a specific object in the project database in order to simplify programming. Labels are also used to identify a rule in the rules file. Typically, object labels describe the physical location of the device that the object represents.

Tip: Excessively long labels are generally harder to read. Sometimes less is more.

Labels have the following characteristics:

- Labels must be unique. Duplicate labels generate compiler errors and prevent the database from compiling.
- Labels are arbitrary except for labels that are automatically assigned by the system.
- Labels are not case sensitive and may contain up to 40 characters. The characters may be any ASCII character except: braces "{ }" , the percent symbol "%", the number symbol "#", less than and greater than symbols "< >", asterisks "*", and blank spaces.

The SDU automatically replaces invalid label characters as shown below to prevent programming errors.

Character substitution table

User types	User sees
space	underscore
*	@
%	@
#	@
<	(
>	)
{	(
}	)

Creating a rules file

The System Definition Utility provides an editing tool for creating and editing a rules file. The Rules Editor (default editor) is limited to editing files of 32 kilobytes or less. If it appears the 32K limit will be exceeded, a different text editor must be used to edit the rules file.

You can customize the SDU by changing the Rules Editor option from the default editor to another editing tool (Word, Wordpad, etc.). When you select Edit Rules from the Rules menu, the SDU will automatically run the external editor and load the project rules file.

Note: When using an external editor, always save your file before exiting the editor. It's also a good practice to save your file before compiling and before saving the project.

Tip: Always look for instances where advanced programming techniques may be used to reduce the rule file's size.

Order is important

When you have more than one rule that uses the same input requirements to trigger separate output responses, the order in which the rules appear in the rules file affects how the rules are executed. The compiler takes multiple rules with like input statements and executes them as though they were a single rule containing multiple output statements. The output statements are executed in the same order that they appear in the rules file.

Say you have a rules file containing the following rules:

```
[Rule 1]
ALARM SMOKE 'LVL5_SMK1' :
      DELAY 30,
      FANON 'STAIRWELL_PRESSURE_2';

[Rule 2]
ALARM SMOKE 'LVL5_SMK1' :
      FAST 'CAB1_PNL1_LED1';

[Rule 3]
ALARM SMOKE 'LVL5_SMK1' :
      ON AUDIBLE 'LVL5_HORN';
```

After compiling, the rules would be executed as though they were written as:

```
[Compiled Rule]
ALARM SMOKE 'LVL5_SMK1' :
      DELAY 30,
      FANON 'STAIRWELL_PRESSURE_2',
      FAST 'CAB1_PNL1_LED1',
      ON AUDIBLE 'LVL5_HORN';
```

Notice that placing Rule 1 before Rule 3 results in a delay before sounding the horn. By writing rules with multiple output statements to begin with, situations such as this are more easily recognizable and can be avoided.

It does not make a difference however when you have more than one rule that uses the same device but different event types. To illustrate this suppose you have the following two rules:

```
[Rule 1]
TROUBLE SMOKE 'LVL5_*':
      STEADY 'CAB1_PNL1_LED1',
      DELAY 300,
      SLOW 'CAB1_PNL1_LED1';

[Rule 2]
ALARM SMOKE 'LVL5_*':
      FAST 'CAB1_PNL1_LED1',
      ON AUDIBLE 'LVL5_HORN';
```

If any smoke detector on level 5 goes into trouble, the panel will activate Rule 1 and turn the specified LED on, wait 5 minutes (300 seconds), and then flash the LED at a slow rate. If during the delay period another smoke detector on level 5 goes into alarm, the panel will activate Rule 2 and flash the LED at a fast rate, and turn on the specified horn.

The panel does not wait for the trouble response (Rule 1) to finish before running the alarm response (Rule 2). In this example however, once the delay period has ended and the trouble has not been restored, the LED will be set to flash at the slow rate.

Tip: When adding rules to a previously compiled rules file, place them at the beginning of the file so they will be checked first. If they need to be in a certain spot in the file, you can move them afterwards.

Avoid careless use of wildcards

A wildcard is a very powerful programming aid and can help reduce the size of your rules file. You should be aware that using wildcards carelessly could have possible negative side effects. Take for example the following three rules:

```
[Rule 1]
      ALARM '*':
      FAST 'LED_1';

[Rule 2]
ALARM 'SD_*':
      DELAY 10,
      FAST 'LED_2';

[Rule 3]
ALARM 'SD_L1_*':
      DELAY 10,
      FAST 'LED_3';
```

When a wildcard is placed in the input statement, the compiler creates an output response for each device that matches the input. Using the above rules and remembering that order is important, the output response for an alarm-initiating device labeled 'SD_L1_1' would be equivalent to:

```
[Compiled Rule]
ALARM 'SD_L1_1':
      FAST 'LED_1',
      DELAY 10,
      FAST 'LED_2',
      DELAY 10,
      FAST 'LED_3';
```

In this example, adding the smoke detector device type to the input statement would limit the output responses to only smoke detectors with matching labels.

Likewise, you must be careful when using the wildcard character in an output statement. For example, placing a wildcard immediately following the N-variable may return undesirable results. 'LVL<N:1>*' will select 'LVL1', 'LVL19', 'LVL199', and so on.

Compiling the rules file

After creating rules in the rules editor, they must be compiled. The compiler checks the rules file for any faults, such as improper syntax, before translating the database into a binary setup file. The compiler also checks the database for objects that are not labeled and objects with duplicate labels.

The speed at which the compiler can check the rules file depends on the size of the database and how the rules are constructed. Compile speed is relative; however the basic guidelines are:

- Rules that only specify device types compile fastest
- Rules that only specify the object label compile slowest
- Rules that specify both compile somewhere in the middle

The rules compiler ignores any characters between opening and closing braces. The rules compiler also ignores tabs, spaces, and line breaks. You should always run the rules compiler anytime changes are made to the project database.

Tip: When troubleshooting your rules file, use braces to temporarily comment out parts of the file that are correct.

Developing a labeling plan

The system definition process will require that you assign labels to cabinets, modules, and other objects in the database. Before you start defining your system, develop a labeling plan. A good labeling plan will ensure that your labels will be understandable and useful.

There are five things you should consider in your labeling plan:

- Label format
- Label content
- Using common label modifiers
- Label numbering
- Using labels as messages

Formatting labels

To make your labels more readable, and more understandable, your labeling plan should include how labels will be formatted. You should take into consideration that labels will be viewed online, on printed reports and on the system display panel. Formatting considerations may include:

- How to separate label modifiers
- Whether to use all uppercase or all lowercase characters or a combination of both
- How label modifiers may be abbreviated.

Note: Labels are not case sensitive. To the compiler, 'ABCDEF' and 'abcdef' are duplicate labels, but 'ABC_DEF' is not.

Suppose that you are creating labels for several cabinets in an industrial park of several buildings. The following shows three examples using different methods of label formatting for a cabinet in one of the buildings:

```
BLDG1CAB    BLDG1_CAB   Bldg1Cab
```

The first label may be hard to read because it uses all uppercase characters and there is no separation between the label modifiers BLDG1 and CAB. The second label is easier to read because of the space, represented by an underscore, between the two modifiers. The third label uses upper and lowercase characters to differentiate between label modifiers.

Always remember to be consistent and find a comfortable balance between readability and length when formatting labels. Consistency is the most important factor in making your labels easy to use and understand. Notice in the example above that each label abbreviates “Building” the same way. You want to avoid using B1, Bldg_1, and Bldg1 as label modifiers to reference the same building. Adding extra spaces to separate

label modifiers, as in Bldg_1_Level_7_Cab, may make the label more readable but unnecessarily adds to the length.

Making labels descriptive

To make your labels more useful, your plan should include how labels will be constructed. The content of the label should include some descriptive modifiers that reference the object's location, function or device type. Depending on the application, label modifiers may be combined to describe location and function.

Using labels to describe location

The most common use for labels is to describe the physical location of a device in the system. The panel can easily identify each device by their assigned logical address, but some times that address doesn't mean much to an operator. Typical modifiers for devices might include:

```
LVL1     ROOM_101     EAST_WING
```

Typical modifiers for cabinet labels in a campus application might include building names, such as:

```
B1_C1    SUTTON_HALL   ADMIN_BLDG
```

Using labels to describe function

Some devices in a system, either by design or configuration, provide specific functions. These devices may include operator panels and amplifiers, among others. Typical modifiers for devices that provide a specific function might include:

Label	Description
DMPR_CNTRL	Damper control panel.
AMP_LEVEL7	Amplifier for floor designated as the seventh level in a multi-floor building.
LEDPANEL	LED panel used as a point annunciator.

Using labels to describe a device type

A system may contain a large number of certain devices, such as detectors. For devices that exist in large quantities, you might want to consider using a label to describe the object's device type. Typical modifiers for these devices include:

Label	Description
SMK	Smoke detectors
PULL	Pull stations
LED	Control/display module LEDs
SW	Control/display module switches

Using common label modifiers

To make programming easier, consider using common label modifiers when developing your labeling plan. For example, using floor numbers as label modifiers can sometimes create extra work for the system designer when numberless areas also exist in the same building. Using common label modifiers will allow you to assign labels faster using the prefabricated labels editor and also permit using wildcards when writing rules.

Using common labels for building levels

Basements and mezzanines are good examples of areas that are not typically referred to by floor number. On projects where numberless floors exist, you might want to use a more generic label modifier, such as “levels”, which can apply to all areas, as shown in Figure 1-1.

Note: You may want event messages that appear on the 3-LCD module to still refer to floors.

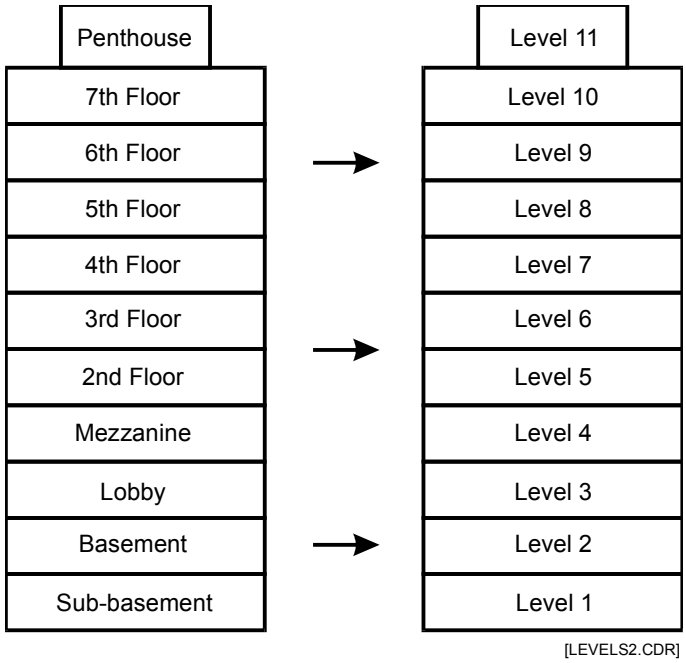


Figure 1-1: By converting “floors” to “levels”, all areas can be labeled with a common modifier that is also meaningful.

Using common labels for vertical applications

Vertical spaces, such as stairwells, elevator shafts, and pipe chases, may be identified by placing a special character in the label modifier, for instance a “V” for vertical. Vertical spaces typically have unique application requirements. The ability to uniquely identify objects associated with vertical spaces often proves useful when writing rules.

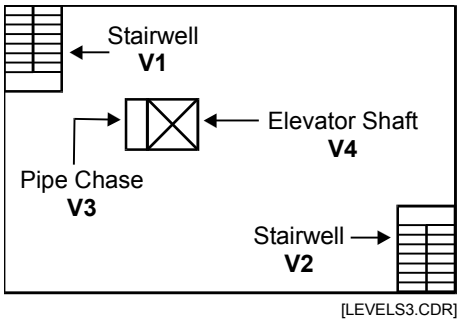


Figure 1-2: Labeling vertical areas

Using numbers in labels

Another way to make programming easier is to have your plan include numbers as part of a label. Using numbers in labels is particularly useful when you need to uniquely identify many objects having common label modifiers. You will also be able to apply some of the advanced programming techniques, such as numeric indexing for floor-above and floor-below applications.

If devices are to be activated on a by-floor basis, some reference to the floor's level number should be made in the device's label. Unique labels may identify special control functions such as elevator lobby, smoke detectors, stairwell, and duct applications.

To identify a device as a duct smoke detector, add the label modifier DUCT to the label. Additional duct detectors on the same floor can use a similar label but having a different number at the end of the label. For example, 'LEVEL2_DUCT1' is one possible label for a duct smoke detector located on the 2nd level. The number 1 is added at the end of the duct detector label to make the label unique, assuming there is more than one duct smoke detector on the second level.

From a programming perspective, having labels with one or more common modifiers is very advantageous. When it comes time to write rules for the system, 'LEVEL2' could easily be replaced with 'LEVEL<N>'.

Using labels as messages

Each addressable point can have a message that is displayed when the point changes state. If you design your labels with this function in mind, you may be able use the label as the message instead of creating a separate message.

Suppose that when a smoke detector goes into alarm, you want the 3-LCD module to show its location. If you label the smoke detector as ' SMOKE_ROOM_101', the display would read:

```
ALARM ACTIVE
SMOKE ROOM 101
```

Note: Only the first 42 characters of a message appear in the 3-LCD module; two rows of 21 characters each. Underscores appear as spaces.

Identifying objects in the system

It is important when reading a job specification to get a feeling for the functions required of the system, as well as the location and types of devices required to initiate these functions. The goal is to identify common functions, areas, devices, etc., to create labels that permit efficient system programming in user-friendly terms. The most effective method of recognizing common input and output functions is to create an input vs. output matrix chart.

In an input vs. output matrix chart, the input devices are listed on one axis of the matrix and the output devices are listed on the other. Starting with the first input, an “X” is placed at the intersection of each output activated by the input. Figure 1-3 is an abbreviated example of an input vs. output matrix chart.

When all inputs and outputs have been entered on the matrix, inspect the matrix for groups or patterns of “X”s. These may appear as lines of “X”s, or blocks of “X”s. In Figure 1-3, there are four groups of “X”s that can be grouped together. When the groups have been identified, look at the inputs that are part of the group to see what they have in common.

The largest group is made up of the General Alarm, Notify Central Station, and Stairwell Door Locks columns which are completely filled with “X”s. The function these objects have in common is that they all must be initiated whenever any input device is activated. The first rule to be written must activate these three functions whenever any device goes into alarm.

The next group is the Elevator Capture function. Inspection of the inputs that initiate the elevator capture function reveals that they are all elevator lobby smoke detectors. This should signal the designer that a label modifier such as “Elevator” or “Lobby” should be included as part of the device label.

Because there are other smoke detectors on the job, the label modifier “Smoke” is not the best choice as a label modifier to identify the elevator capture function. (“Smoke” may be included as a label modifier in the device label, but it will not be used for the elevator capture function.) A better label modifier choice would be “Elevator” or “Lobby”, or a combination of both.

Figure 1-3: Input Vs Output Function Matrix

Add a label modifier to the label of a 2nd floor smoke detector identify the detector as a elevator lobby smoke detector, for future use in an elevator capture rule.

```
'FLOOR2_SMOKE1_ELEVLOBBY'
```

is one possible label for the 2ND floor elevator lobby smoke detector.

The keyword “ELEVLOBBY” will be incorporated in the elevator capture rule. Any detector having “ELEVLOBBY” as part of its label will activate the elevator capture function.

The next object group is the Computer Shutdown function. Inspection of the inputs that initiate the computer shutdown

function reveals that they are all devices located on the 2ND floor. This should signal the designer that a label modifier such as “FLOOR2” or “LEVEL2” should be included as part of the device label for every device on the second floor. This also implies that the floor number should be included in the labels of all devices.

The keyword “FLOOR2” will be incorporated in the computer shutdown rule. Any device having “FLOOR2” as part of its label will activate the computer shutdown function. Note that the elevator lobby detector labeled in the previous example will also shutdown the second floor computer.

The final object group identified is the 1-3 Floor Exhaust function. Examination of the inputs that initiate the 1-3 exhaust function reveals that they are all duct smoke detectors. A label modifier such as “DUCT” should be included as part of the device label for all duct detectors. Notice that the input/output matrix requires that the basement duct detector does not activate the 1-3 Exhaust function. This should not affect the choice of “DUCT” as a label modifier in the device label, as it is useful to the designer in determining the device's function.

The keyword “DUCT” will be incorporated in the 1-3 Exhaust rule. Because not all detectors with the keyword “DUCT” in their labels are required to activate the 1-3 exhaust function, the additional keywords “FLOOR1”, “FLOOR2”, and “FLOOR3” will be added to the 1-3 exhaust function rule.

Notice that there are a few matrix entries that are not part of any group. These functions are activated by a specific device. The rules used to activate these outputs are written for the specific function, rather than use by multiple objects.

The matrix for a major project would be very large and detailed. Many inputs will have only one or two output correlations. The experienced system designer will quickly recognize the patterns that develop.

In Figure 1-3, notice that the first, second and third floors are basically copies of each other. This is typical of high rise buildings. This duplication of functions permits a designer to recognize that the floor number will be an important label modifier when creating labels for devices.

Our example also reveals the importance of well-planned label modifiers. The proper selection of device labels permits the designer to account for the common functions throughout the building by writing only a few rules.

Priorities

The output priority parameter specifies the relative importance of the output with respect to other rules activating the same output and is an optional parameter.

Rules and objects programming has the ability to prioritize output commands. There are two output priority levels, HIGH and LOW. A high priority command will override a low priority command. A low priority command can not override a high priority command. A special priority called LATCH is discussed later in this section.

Because an output device may be affected by any number of input devices, each output has two software counters associated with it to keep track of the output's status. The High priority counter reflects the number of high priority set and reset commands issued to the output, and the resulting high priority status. The low priority counter reflects the number of low priority set and reset commands issued to the output, and the resulting low priority status.

An output's status is the result of both the high and low priority counters as governed by the rules below.

A. Every set command increments the appropriate counter by one; restoring a set command decrements the appropriate counter by one.

B. Every reset command decrements the appropriate counter by one; restoring a reset command increments the appropriate counter by one.

C. When the high priority counter is equal to or less than negative one (-1), the output is off.

D. When the high priority counter is equal to or greater than one (+1), the output is on.

E. When the high priority counter is zero (0) and the low priority counter is equal to zero (0), the output is off.

F. When the high priority counter is zero (0) and the low priority counter is equal to or greater than one (+1), the output is on.

When the input (left) side of a rule references the alarm event type, the High priority level is the default value automatically assigned to all output commands on the right side of the rule. The output priority can be changed for the special cases where low output priority is required in command to an alarm input.

When the input (left) side of a rule references any event type except the alarm state, the Low priority level is the default value

automatically assigned to all output commands on the right side of the rule. The output priority can be changed when high output priority is required.

Latch priority

The latch priority function is a special output priority level used to force an output to a specific state and reset the high and low priority counters. The latch priority is not normally required for normal system programming.

Some functions such as a scheduled time control with no output duration specified can cause the priority counters to increment indefinitely. The latch priority can be used to force the time control to a known state and reset the counters.

Example: Priority settings

Eight inputs, IN1 through IN9, are issuing commands to an output. Table 1-1 lists the contents of each counter and the resultant state of the output.

0 All counters initially start out at zero. The output is off, per rule E.

1 Input IN1 sets the output with a low priority. The Low Priority Counter (LPC) increments to 1 per rule A. Output is on, per rule F.

2 Input IN2 sets the output with a low priority. The LPC increments to 2 per rule A. The output is still on, per rule F.

3 Input IN3 resets the output with a high priority. The High Priority Counter (HPC) decrements to -1 per rule B. The output is now off, per rule C.

4 Input IN4 resets the output with a low priority. LPC decrements to 1 per rule B. The output is still off, per rule C.

5 Input IN5 sets the output with a low priority. The LPC increments to 2 per rule A. The output is still off per rule C.

6 Input IN6 sets the output with a low priority. The LPC increments to 3 per rule A. The output is still off per rule C.

7 Input IN7 sets the output with a low priority. The LPC increments to 4 per rule A. The output is still off per rule C.

8 Input IN8 sets the output with a low priority. The LPC increments to 5 per rule A. The output is still off per rule C.

9 Input IN9 resets the output with a high priority. The HPC decrements to -2 per rule B. The output is still off per rule C.

10 IN3 restores. The HPC increments to -1 per rule B. The output is still off per rule C.

11 IN9 restores. The HPC increments to 0 per rule B. The output is now on per rule F.

12 IN1 restores. The LPC decrements to 4 per rule A. The output is still on per rule F.

13 IN2 restores. The LPC decrements to 3 per rule A. The output is still on per rule F.

14 IN4 restores. The LPC increments to 4 per rule B. The output is still on per rule F.

15 IN5 restores. The LPC decrements to 3 per rule A. The output is still on per rule F.

16 IN6 restores. The LPC decrements to 2 per rule A. The output is still on per rule F.

17 IN7 restores. The LPC decrements to 1 per rule A. The output is still on per rule F.

18 IN8 restores. The LPC decrements to 0 per rule A. The output is now off per rule E.

All inputs have restored, and the output is back at the quiescent state. Rewriting the rule used in the elevator capture example to include output priorities:

```
ALARM SMOKE 'BLD2_ELOBBY':
      ON -LOW 'ELEVRELAY';
```

Table 1-1: Output Priority Example (Reset = -1 Set = +1)

Step	Event INx = Input	High Priority Counter ≤ -1 = Output Off 0 = See Low Priority ⇒ ≥1 = Output On	Low Priority Counter ≤ 0 = Off >1 = On,	Output State
0	Quiescent	0	0	OFF
1	Activate IN1 : ON -LOW;	0	1	ON
2	Activate IN2 : ON -LOW;	0	2	ON
3	Activate IN3 : OFF -HIGH;	-1	2	OFF
4	Activate IN4 : OFF-LOW;	-1	1	OFF
5	Activate IN5 : ON -LOW;	-1	2	OFF
6	Activate IN6 : ON -LOW;	-1	3	OFF
7	Activate IN7 : ON -LOW;	-1	4	OFF
8	Activate IN8 : ON -LOW;	-1	5	OFF
9	Activate IN9 : OFF -HIGH;	-2	5	OFF
10	Restore IN3 : OFF -HIGH;	-1	5	OFF
11	Restore IN9 : OFF -HIGH;	0	5	ON
12	Restore IN1 : ON -LOW;	0	4	ON
13	Restore IN2 : ON -LOW;	0	3	ON
14	Restore IN4 : OFF -LOW;	0	4	ON
15	Restore IN5: ON -LOW;	0	3	ON
16	Restore IN6 : ON -LOW;	0	2	ON
17	Restore IN7 : ON -LOW;	0	1	ON
18	Restore IN8 : ON -LOW;	0	0	OFF

Advanced programming techniques

Advanced programming techniques allow the system programmer to create more robust rules files. You can use any of the following:

- Wildcards
- N-variables
- Mathematical operators

Wildcards

The asterisk (*) can be used in a rule as a wildcard character to conditionally select devices based on character patterns. The asterisk may be substituted for any single character or group of characters anywhere in the label.

Example	Selects
'LVL*'	Any devices whose labels begin with "LVL"
'*_SMK'	Any devices whose labels end with "_SMK"
'LVL*_SMK'	Any devices whose labels begin with "LVL" and end with "_SMK"
'*_SMK_*'	Any devices that have "_SMK_" somewhere in the label
'*'	All devices whose device types are valid for the specified input event type or output command

N-variable

The N-variable can be used in a rule to conditionally select devices based on the numerical indexing used in their object labels. When using the N-variable, the numbers required to make the conditional true are specified in the input statement. When the input statement becomes true, the number is substituted for the variable N in the output statement.

Example	Selects
<N : #>	A single number entry
<N : # - #>	A range of numbers
<N : #, #, # - #, # - #>	A combination of single numbers and number ranges

Notes:

- The N-variable may be any number between 0 and 32767, inclusive.
- The N-variable may only be used in a label once
- Wildcards may not be used in place of a number.
- Any combination of single number and ranges may be used "< n: #, #-#, #, #-# >" with the restriction that there must be less than 255 characters between the single quotes in the object label.

In the following rule, pressing the switch labeled 'LVL1_PHONE_CNTRL' turns on a module labeled 'LVL1_PHONE_JACK'. The same rule turns on a module labeled 'LVL5_PHONE_JACK' when the switch labeled 'LVL5_PHONE_CNTRL' is pressed.

```
[PHONE_SWITCH]
SWITCH 'LVL<N:1,3-5>_PHONE_CNTRL':
      ON FIREPHONE 'LVL<N>_PHONE_JACK';
```

Note: The above rule does not produce an output response when a switch labeled 'LVL2_PHONE_CNTRL' is pressed.

Mathematical operators

Mathematical operators can be used in conjunction with the N-variable in the rule output statement to conditionally select devices based on the numerical indexing used in their object labels.

When using mathematical operators, the numbers required to make the conditional true are specified by the N-variable in the input statement. When the input statement becomes true, the number is substituted for the variable N in the output statement and then increased or decreased by the number specified by the # parameter.

Example	Selects
<N + #>	A number greater than the number determined by N in the input statement
<N - #>	A number less than the number determined by N in the input statement

Notes:

- The # parameter may be any number between 1 and 32767, inclusive.

- <N + #> may not equal greater than 32767.
- <N - #> may not equal less than zero.
- Wildcards may not be used in place of the #-parameter.

An application calls for all devices on a floor of a 10-story building to activate the horns and strobes on the fire floor, the floor above the fire floor, and the floor below the fire floor. Using mathematical operators, the rule required might appear as shown below.

Note: You can not combine multiple N-variables in the same pair of brackets, i.e. <n, n+1, n-1> is not valid.

```
[FireFlrHorn]
ALARM 'LEVEL<n:1-10>*':
      ON 'LEVEL<n>_HORN',
      ON 'LEVEL<n>_STROBE',
      ON 'LEVEL<n+1>_HORN',
      ON 'LEVEL<n+1>_STROBE',
      ON 'LEVEL<n-1>_HORN',
      ON 'LEVEL<n-1>_STROBE';
```

Programming using logic groups

System programming permits devices to be combined together in a logic group to provide a separate group response from individual device responses. The following types of logic groups may be defined for system programming.

- Nonalarm logic groups (And, Matrix, Instruction Text)
- Alarm logic groups (Zone)
- Trouble logic groups (Zone)
- Check-In groups
- Guard Patrol groups
- Service groups

In general:

- Any 1 device can be a member of up to1 Zone group, 10 And groups, 10 Matrix groups, 1 Service group, and 4 Instruction Text groups
- Any device that is a member of a Check-In group can only be a member of 1 Check-In group and no other groups.
- Any device that is a member of a Guard Patrol group can only be a member of 1 Guard Patrol group and no other groups.

Logic groups have the following parameters:

And	499 per project, max 64,000 devices per group, max 1 to 255 devices required to activate
Matrix	255 per project, max 64,000 devices per group, max 1 to 10 devices required to activate 10-device search radius, max
Instruction Text	999 per project, max 64,000 devices per group, max
Zones	999 per project, max 64,000 devices per group, max
Check-In	255 per project, max 255 devices per group, max
Guard Patrol	255 per project, max 10 tours per group, max 64 devices per tour, max
Service	255 per project, max 64,000 devices per group, max

Nonalarm logic groups

And groups

And groups are a collection of devices that are grouped together to provide a unique response based on:

- a specific number of device activations (changes in state)
- the activation type

For example, you can configure an And group of eight smoke detectors (SMK_1–SMK_8) to activate when any three go into alarm with the exception of SMK_5 which can go into alarm or trouble.

Notes

- Each And group member's change in state produces a separate event that may be viewed on the 3-LCD in addition to the And group's activation event.
- If a member can have more than one concurrent change in state, an example being a smoke detector that is in trouble and then goes into alarm, each change in state counts as 1 activation.

And group activations can be configured for any event type (alarm, supervisory, trouble, monitor). For example, you can have an And group that produces a trouble event when it activates and another that produces an alarm event.

And groups can consist of any of the following device types:

Acknowledge, AlarmSilence, AllCall, AlternateLanguage, AlternateMsg, AlternateSensitivity, And, Audible, CardDBIncompatibility, CfgMismatch, CommFailure, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Drill, Evacuation, ExtDBIncompatibility, FailSafe, FanControl, FanFeedback, Firephone, FirstAlarm, FirstDisable, FirstMonitor, FirstSupervisory, FirstTrouble, GAInhibit, Gatevalve, GenAlarm, GroundFault, Heat, Lamptest, LocalAlarm, LocalMonitor, LocalRelay, LocalTrouble, Matrix, Monitor, NonsupervisedOutput, Power, Pull, R1, R2, R3, RebootFault, Reset, Security, ServiceDeviceSupervision, ServiceGroupActive, Smoke, Smoke Vfy, StageOne, StageTwo, Startup, SupervisedOutput, Supervisory, Switch, Tamper, TaskFailure, Temperature, Text, TroubleSilence, TwoStageTimerActive, TwoStageTimerExpiration, UserTrouble, Visible, Waterflow, Zone

Matrix groups

Matrix groups are a collection of device types that are grouped together to provide a unique response similar to the way an And group operates. In addition to an activation number, a Matrix group searches a range of devices around the first device in the group that goes active for a second device to go active in order to activate the group. Each device within a Matrix group is assigned a set of x,y coordinates to create a matrix grid according to their mounting location.

The radius of a Matrix specifies the size of the search radius. The search radius is the number of devices away from the first active device that the system will search for a second device. The activation number of a Matrix group specifies the number of device activations that must occur to turn on the output of the Matrix group. Either condition will activate the output of the Matrix group.

A Matrix group can be configured to generate any activation type (alarm, supervisory, trouble, monitor) and can consist of any of the device types found in an And group.

Figure 1-4 illustrates the activation process for a Matrix group with only alarm activations.

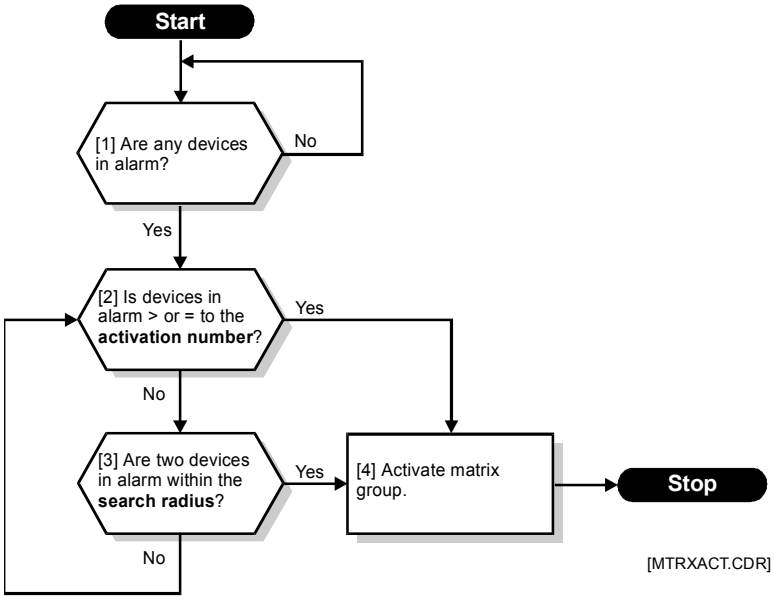
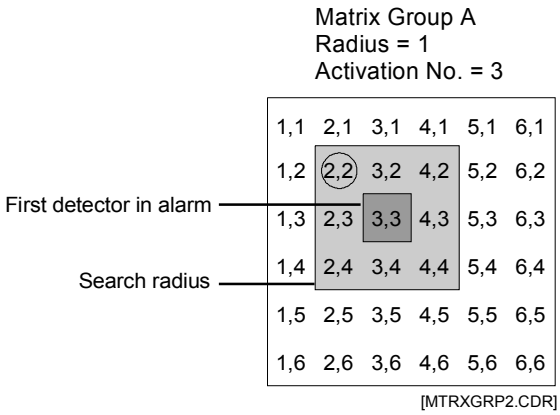


Figure 1-4: Matrix group activation process flow chart

Matrix group activated by the radius setting

A computer room has 36 detectors installed in a six-by-six pattern. A matrix group is created, locating all 36 detectors in

their respective positions in the matrix. The Matrix group is configured having a radius of 1 and an activation number of 3.



In the illustration above, the detector at location 3,3 was the first detector to go into alarm, as indicated by the dark gray box. The Matrix group creates a search radius around the first active device, as indicated by the light gray box. Should any additional detector in the light gray box go into alarm, the Matrix group will activate its output response.

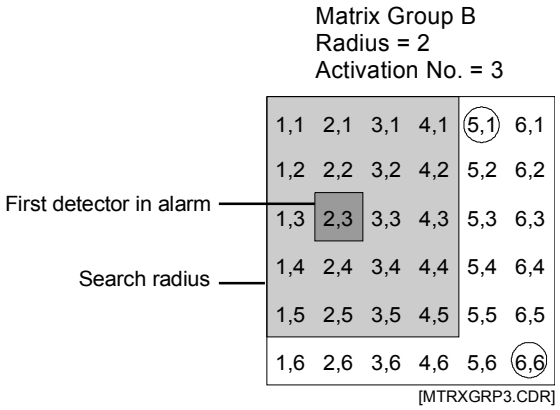
The following rule activates a device labeled 'HALON' when the matrix group labeled 'MATRIX_GROUP_A' changes to the active state and produces an alarm event.

```
[SampleRule]
Alarm Matrix 'MATRIX_GROUP_A':
      ON 'HALON';
```

Note: The input event type used in rules written for And and Matrix groups depend on the group's Activation Event setting. For example, you must use the Trouble event type when the activation event is set for trouble.

Matrix group activated by activation number setting

Matrix Group B consists of 36 detectors and is configured with a radius setting of 2 and an activation number of 3.



The detector at matrix location 2,3 was the first detector to go into alarm, as indicated by the dark gray box. The Matrix group creates a search radius around the first active device, as indicated by the light gray box. Detectors 5,1 and 6,6 go into alarm. The Matrix group will activate its output even though a second detector within the search radius did not go into alarm, but because the total number of detectors in alarm was equal to the activation number setting.

Instruction Text groups

Instruction Text groups are used to provide additional detailed instructions or warnings for when any device type in a particular group goes active. You can further qualify an active device by its active state. For example, you can have the Instruction Text group activate when a device goes into alarm or into trouble.

When any device in the group goes active, the *Instruction Text Active* event is sent to the Monitor queue on the 3–LCD. The message text is routed through the printer port. The message text may also be reviewed by selecting the *Instruction Text Active* event in the Monitor queue and pressing the Details switch.

An Instruction Text group can consist of any of the device types found in an And group.

Zone groups

Zone groups are a collection of input devices that are grouped together to provide a unique response separate from their individual device responses. For all the devices in the group that go into alarm, only a single *Zone Active* event is sent to the Alarm queue on the 3-LCD. To determine which devices in the zone are active, select the *Zone Active* event in the Alarm queue and press the Details switch.

Zone groups can consist of any of the following device types:

Evacuation, Failsafe, GenAlarm, Heat, LocalAlarm, Pull, Smoke, SmokeVfy, StageOne, StageTwo, Waterflow

Zone groups can also be configured to activate when any device in the group goes into trouble.

With firmware version 1.4, it is no longer required to create a Zone group or be programmed for Proprietary operation in order to utilize a CDR-3 Zone Coder. You can place the zone code at the beginning of the message of each individual point used to activate the CDR-3. There are two advantages to not activating a CDR-3 from a Zone group:

- The 3-LCD displays each device activation as a separate alarm event.
- Subsequent device activations will initialize succeeding alarms.

Setting the Zone Group Inhibit option prevents the zone group's response from re-activating when subsequent members of the group go active after the outputs have been silenced.

Check-In groups

Check-in groups are a collection of Emergency device types that are grouped together to provide a unique response when members of the group fail to check in during a prescribed time period. If one or more devices do not check in during the allotted time, a *Check-In Active* response is generated, indicating those devices which have failed to check in. Check-in groups are typically used in senior citizen housing to monitor the wellness of occupants.

Note: Devices that do not check in are removed from the check-in list and must be restored in order to be available for the next check-in period.

If a device connected to a Check-in group is activated outside of the check-in time window, an *Emergency* response is activated. If a device that checked in during the required time period is activated a second time within the check-in window, an *Emergency* activation will also be generated.

Guard Patrol groups

Guard Patrol groups are groups of input devices that must be activated in a sequential order and within specified time constraints. Each defined patrol lists the number of stations, and the minimum and maximum times to reach each patrol station. A rule must be written that defines the actions to be taken in the event of an out of sequence or delinquent patrol.

Service groups

Service groups are a collection of devices that are grouped together for testing purposes. When enabled, the Service group automatically disables the member device's normal alarm response, and provides a common alternate test response.

Service groups can consist of any of the following device types:

Audible, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, CommonTroubleOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Emergency, FanControl, FanFeedback, Firephone, Gatevalve, GenAlarm, Guard, Heat, Monitor, Power, Pull, Security, Smoke, SmokeVfy, StageOne, StageTwo, SupervisedOutput, Supervisory, Tamper, Temperature, Visible, Waterflow

Programming using time controls

Time controls are used as an argument in the input statement of rule to initiate system events at a specific date and time. Time-controlled events could include such things as: locking/unlocking doors, turning lights on/off, and switching device sensitivities.

A time control becomes active (true) when all its settings match the day/date/time value generated by the system calendar/clock. After becoming active, a time control stays active for the length of time specified by its duration setting.

You can also configure a time control to activate only when it is a scheduled holiday. Holidays are scheduled from the 3-LCD module. Holiday time controls activate in addition to non-holiday time controls.

To make configuring regularly occurring time-controlled events easier, follow these general rules when making month, day, and date settings:

If you want a time control to activate…	then select…
Once a year	The target month, every day of the week, and the date
Once a month	Every month, every day of the week, and the target date
Once a week	Every month, the target day of the week, and every date
Once a day	Every month, every day of the week, and every day

Time control for a specific date each month

Suppose you have an application requiring the system to execute a specific command on the 15th day of every month. Instead of configuring twelve separate time controls, this could be accomplished using one time control labeled 'Day_15_only' as follows:

1. Select every month.
2. Select every day of the week.
3. Select 15 in the date field.

Time control for a specific day each month

Suppose you have an application requiring that system to execute a specific command on the first Monday of every month. This could be accomplished using one time control labeled 'First_Monday', as follows:

1. Select every month.

2. Select Monday.

3. Select numbers 1-7 in the date field.

In this example, dates 1-7 are selected because the first Monday of a month could fall on any of the first seven days.

Time control for a range of days

Suppose you have an application requiring that system to execute a specific command on a Monday–Friday schedule. You could create a time control labeled 'Mon_thru_Fri' as follows:

1. Select every month.

2. Select only Monday, Tuesday, Wednesday, Thursday, and Friday.

3. Select every date because Monday - Friday can land on any date depending on the month.

Chapter 2

Input event types

Summary

This chapter provides an alphabetical reference of the input event types used to program the system. Optional parameters are shown in *italics*.

Content

Acknowledge (ACK)

Use the Acknowledge event to activate a rule when an operator acknowledges an event displayed on the 3-LCD module.

The Acknowledge event requires that you specify a device type or an object label, or both.

Acknowledge *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: And, Audible, CommFailure, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Emergency, ExtDBIncompatibility, Failsafe, FanControl, FanFeedback, Firephone, Gatevalve, GenAlarm, GenSmoke, GroundFault, Guard, GuardPatrol, Heat, LocalAlarm, LocalMonitor, LocalRelay, LocalTrouble, LoopControllerResetExt, Matrix, Monitor, NSCommonAlarmOutput, NSCommonMonitorOutput, NSCommonSupervisoryOutput, NSCommonTroubleOutput, PanelCommFault, Power, Pull, Security, ServiceDeviceSupervision, ServiceGroup, ServiceGroupActive, Smoke, SmokeVfy, SprinklerSupervisory, StageOne, StageTwo, SupervisedOutput, Supervisory, Switch, Tamper, TaskFailure, Temperature, TwoStageTimerActive, TwoStageTimerExpiration, UserTrouble, Visible, Waterflow, Zone
'object_label'	Specifies the label of the device initiating the event.

Note: The Acknowledge input event can only be used when the system is configured as a proprietary fire alarm system, in accordance with NFPA 72.

Example

```
{Light LED on control/display module to indicate
message for SMK_1 has been acknowledged}

[ACKMSGON]

ACK SMK 'SMK_1':

   STEADY -LOW 'DISPLAY_1_LED1';
```

Alarm

Use the Alarm event to activate a rule when any point on a panel or any panel in the same network routing group changes to the alarm state.

The Alarm event requires that you specify a device type or an object label, or both.

Alarm *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: And, Failsafe, GenAlarm, GenSmoke, Heat, Matrix, Pull, Smoke, SmokeVfy, StageOne, StageTwo, Waterflow, Zone
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Turn on fan relay when any lobby smoke detector
goes into alarm}

[FAN_ON]

ALARM SMK 'LVL*_LOBBY*':

   FANON -HIGH 'FAN_RELAY';
```

Tip: Use the GenSmoke device type when you want to write a single rule that applies to devices that can have either Smoke or SmokeVfy device types.

AlarmSilence (AS)

Use the AlarmSilence event to activate a rule when an operator presses a switch that starts a panel's Alarm Silence function. The switch can be the Alarm Silence switch on the 3-LCD module or a control/display module switch programmed to execute the AlarmSilence command. Typically, you use the AlarmSilence event to silence device types in addition to those automatically silenced by the Alarm Silence function.

The AlarmSilence event does not require a that you specify a device type or an object label.

AlarmSilence :

Note: Project configuration settings determine if the Alarm Silence function automatically silences only audible or audible and visible device types.

Example

```
{Flash LED on control/display module when
notification devices have been silenced}

[ALARM_SILENCE_ON]

AS:

   FAST -LOW 'DISPLAY_1_LED_1';
```

AlarmVerify (AVER)

Use the AlarmVerify event to activate a rule when a smoke detector starts its smoke verification cycle. Typically, the AlarmVerify event is used to provide an indication of potential alarm or pre-alarm conditions.

The AlarmVerify event requires that you specify a device type or an object label, or both.

AlarmVerify *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: GenSmoke, SmokeVfy
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Flash LED on control/display module when any
smoke detector is in its smoke verification
process}
[VERIFY_ON]
AVER GENSMOKE 'SMK_*':
   SLOW -LOW 'DISPLAY_1_LED_1';
```

AllCall

Use the AllCall event to activate a rule when an operator presses the All Call or All Call Minus switch on the 3-ASU.

The AllCall event does not require that you specify a device type or an object label.

AllCall :

Example

```
{Flash LED on control/display module when All
Call in process}

[ALL_CALL_ON]

ALLCALL:

   FAST -LOW 'DISPLAY_1_LED_1';
```

CallIn (CI)

Use the CallIn event to activate a rule when a fire safety professional plugs a handset into a firefighter's telephone jack.

The CallIn event requires that you specify a device type or an object label, or both.

CallIn *device_type 'object_label'*:

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Firephone
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Flash LED on control/display module when
connection made to firefighter's telephone
circuit}

[CALLIN_ON]

CI FP 'FP_LVL_<N:1-10>':
   FAST -LOW 'DISPLAY_1_LED_<N>';
```

Drill

Use the Drill event to activate a rule when an operator presses a switch that starts a panel's Drill function. The switch can be the Drill switch on the 3-LCD module or a control/display module switch programmed to execute the Drill command. Typically, you use the Drill event to activate devices in addition to those automatically activated by the Drill function.

The Drill event does not require that you specify a device type or an object label.

Drill :

Note: Project configuration settings determine if the Drill function automatically activates only audible or audible and visible device types.

Example

```
[DRILL_RESPONSE]
DRILL:
   AMPON 'LEVEL3_AMP' TO 'CH_PAGE_01_08',
   MSGON 'DRILL_MESSAGE' TO 'CH_PAGE_01_08';
```

Emergency (EMER)

Use the Emergency event to activate a rule when a member of a Check-in group activates their check-in device anytime outside of their check-in period or a second time during their check-in period.

The Emergency event requires that you specify a device type or an object label, or both.

Emergency *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Emergency
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Turn on the common distress relay}
[DISTRESS]
EMER '*':
   ON SUP 'DISTRESS_RELAY';
```

Evacuation (EVAC)

Use the Evacuation event to activate a rule when an operator presses a control/display module switch programmed to execute a panel's Evacuation function. Typically, you use the Evacuation event to activate devices in addition to those automatically activated by the Evacuation function.

The Evacuation event does not require that you specify a device type or an object label.

Evacuation :

Example

```
{Flash LED on control/display module when an
Evacuation has been initiated}

[EVACUATION_RESPONSE]

EVAC:

   FAST -LOW 'DISPLAY_1_LED_1';
```

FirstAlarm (FA)

Use the FirstAlarm event to activate a rule the first time that any point on a panel or any panel in the same network routing group changes to the alarm state.

The FirstAlarm event does not require that you specify a device type or an object label.

FirstAlarm :

Example

```
{Light LED on control/display module when the
first alarm event occurs}

[FIRST_ALARM_ON]

FA :

   STEADY -LOW 'FIRST_ALARM_LED';
```

FirstDisable (FD)

Use the FirstDisable event to activate a rule the first time that any point on a panel or any panel in the same network routing group changes to the disabled state.

The FirstDisable event does not require that you specify a device type or an object label.

FirstDisable :

Example

```
{Flash LED on control/display module when the
first disable event occurs}

[FIRST_DISABLE_ON]

FD:

   SLOW -LOW 'FIRST_DISABLE_LED';
```

FirstMonitor (FM)

Use the FirstMonitor event to activate a rule the first time that any point on a panel or any panel in the same network routing group changes to the monitor state.

The FirstMonitor event does not require that you specify a device type or an object label.

FirstMonitor :

Example

```
{Flash LED on control/display module when the
first monitor event occurs}

[FIRST_MONITOR_ON]

FM:

   SLOW -LOW 'FIRST_MONITOR_LED';
```

FirstSupervisory (FS)

Use the FirstSupervisory event to activate a rule the first time that any point on a panel or any panel in the same network routing group changes to the supervisory state.

The FirstSupervisory event does not require that you specify a device type or an object label.

FirstSupervisory :

Example

```
{Flash LED on control/display module when the
first supervisory event occurs}

[FIRST_SUPERVISORY_ON]

FS:

   FAST -LOW 'FIRST_SUPR_LED';
```

FirstTrouble (FT)

Use the FirstTrouble event to activate a rule the first time that any point on a panel or any panel in the same network routing group changes to the trouble state.

The FirstTrouble event does not require that you specify a device type or an object label.

FirstTrouble :

Example

```
{Flash LED on control/display module when the
first trouble event occurs}

[FIRST_TROUBLE_ON]

FT:

   FAST -LOW 'FIRST_TROUBLE_LED';
```

GroundFault (GNDF)

Use the GroundFault event to activate a rule when a rail module detects a ground fault on its field wiring.

The GroundFault event requires that you specify a device type or an object label, or both.

GroundFault *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: GroundFault
'object_label'	Specifies the label of the pseudo point initiating the event.

Example

```
{Flash LED on control/display module to indicate
a ground fault}

[GROUNDFAULT_RESPONSE]

GNDF 'Grnd_Fault_Data_Card_1_01_05':

   FAST -LOW 'GROUND_FAULT_LED';
```

GuardPatrol (GPG)

Use the GuardPatrol event to activate a rule when a patrol guard fails to activate a patrol tour station at the proper time. Typically, you use the GuardPatrol event to create an output response for the entire Guard Patrol group.

The GuardPatrol event requires that you specify a device type or an object label, or both.

GuardPatrol *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: GuardPatrol
'object_label'	Specifies the label of the Guard Patrol group initiating the event.

Example

```
{Turn on the guard patrol late/out of sequence
horn}
[GUARDPATROL_RESPONSE]
GPG 'GUARD_PATROL_Group1':
   ON -HIGH AUD 'GUARD_HORN';
```

LocalAlarm (LALM)

Use the LocalAlarm event to activate a rule when a rail module's LocalAlarm pseudo point goes active.

The LocalAlarm event requires that you specify a device type or an object label, or both.

LocalAlarm *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: LocalAlarm
'object_label'	Specifies the label of the pseudo point initiating the event.

Example

```
{Light LED on control/display module when
unprogrammed device detected on Signature
controller module}

[LOCAL_ALARM_ACTIVE]

LALM 'Unprogrammed_Device_Data_Card_1_01_05':
   STEADY -LOW 'DISPLAY_1_LED_1';
```

LocalMonitor (LMON)

Use the LocalMonitor event to activate a rule when a rail module's LocalMonitor pseudo point goes active.

The LocalMonitor event requires that you specify a device type or an object label, or both.

LocalMonitor *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: LocalMonitor
'object_label'	Specifies the label of the pseudo point initiating the event.

Example

```
{Flash LED on control/display module when the
Signature controller is identifying the devices
on the loop}

[LOCAL_MONITOR_ACTIVE]

LMON 'Reconstct_Line_Data_Card_1_01_05':
   SLOW -LOW 'DISPLAY_1_LED_1';
```

LocalTrouble (LTRB)

Use the LocalTrouble event to activate a rule when a rail module's LocalTrouble pseudo point goes active.

The LocalTrouble event requires that you specify a device type or an object label, or both.

LocalTrouble *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: LocalTrouble
'object_label'	Specifies the label of the pseudo point initiating the event.

Example

```
{Flash LED on control/display module when a map
fault occurs on the Signature controller module}
[LOCAL_TROUBLE_ACTIVE]
LTRB 'Map_Fault_Data_Card_1_01_05':
   FAST -LOW 'DISPLAY_1_LED_1';
```

Monitor (MON)

Use the Monitor event to activate a rule when any point on a panel or any panel in the same network routing group changes to the monitor state.

The Monitor event requires that you specify a device type or an object label, or both.

Monitor *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: And, DamperFeedback, DoorFeedback, FanFeedback, Matrix, Monitor
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Turn on fan status LEDs 1-10}
[FAN_LED]
MON FANFB 'FAN<N:1-10>_STATUS':
   STEADY -LOW 'FAN<N>_ON';
```

R1

Use the R1 event to activate a rule when the first phase of the 3-phase reset cycle starts after an operator presses the Reset switch on the 3-LCD.

The R1 event does not require that you specify a device type or an object label.

R1 :

Note: The R1 event is self-restoring. Any outputs activated by this event automatically restore at the end of the reset cycle.

Example

```
{Flash LED on control/display module during
first phase of the reset cycle}

[FIRST_PHASE_RESET]

R1:

   SLOW -LOW 'R1_LED';
```

R2

Use the R2 event to activate a rule when the second phase of the 3-phase reset cycle starts. The second phase starts after the first phase finishes.

The R2 event does not require that you specify a device type or an object label.

R2 :

Note: The R2 event is self-restoring. Any outputs activated by this event automatically restore at the end of the reset cycle.

Example

```
{Flash LED on control/display during second
phase of the reset cycle}

[SECOND_PHASE_RESET]

R2:

   FAST -LOW 'R2_LED';
```

R3

Use the R3 event to activate a rule when the third phase of the 3-phase reset cycle starts. The third phase starts after the second phase finishes.

The R3 event does not require that you specify a device type or an object label.

R3 :

Note: The R3 event is self-restoring. Any outputs activated by this event automatically restore at the end of the reset cycle.

Example

```
{Light LED on control/display module during
third phase of the reset cycle}

[THIRD_PHASE_RESET]

R3:

   STEADY -LOW 'R3_LED';
```

RelayConfirmation (RLYCFG)

Use the RelayConfirmation event to activate a rule when a control relay indicates that its electrical contacts have switched positions. Typically, you use the RelayConfirmation event in applications that require verifying the proper operation of remote controlled functions.

The RelayConfirmation event requires that you specify a device type or an object label, or both.

RelayConfirmation *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Audible, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DoorControl, FanControl, Firephone, LocalRelay, NonsupervisedOutput, SupervisedOutput, Visible
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Light LED on control/display module when relay
confirmation occurs}

[RELAY_OK]

RLYCFG DOOR 'DOOR_CONTROL_RELAY':

   STEADY -LOW 'DOOR_RELAY_LED';
```

Reset

Use the Reset event to activate a rule when an operator presses a switch that starts a panel's Reset function. The switch can be the Reset switch on the 3-LCD module or a control/display module switch programmed to execute the Reset command. Typically, you use the Drill event to activate output responses in addition to those automatically activated by the Reset function.

The Reset event does not require that you specify a device type or an object label.

Reset :

Example

```
{Flash LED on control/display module during
reset cycle}

[SYSTEM_RESET_ON]

RESET:

   FAST -LOW 'RESET_LED';
```

Note: See also R1, R2, and R3 input event types.

Security (SEC)

Use the Security event to activate a rule when the open input to a device or circuit that monitors a supervisory or tamper switch closes.

The Security event requires that you specify a device type or an object label, or both.

Security *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Security
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Light LED on control/display module when relay
confirmation occurs}
[RELAY_OK]
SEC SEC 'CAB1_TAMPER_SW':
   STEADY -LOW 'TAMPER_SW_LED';
```

ServiceDevice (SERV)

Use the ServiceDevice event to activate a rule when an authorized service technician activates a device in a Service group under test. The ServiceDevice event is used to program individual responses for each device in the Service group.

The ServiceDevice event requires that you specify a device type or an object label, or both.

ServiceDevice *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Audible, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, CommonTroubleOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Emergency, FanControl, FanFeedback, Firephone, Gatevalve, GenAlarm, GenSmoke, Guard, Heat, Monitor, Power, Pull, Security, Smoke, SmokeVfy, SprinklerSupervisory, StageOne, StageTwo, SupervisedOutput, Supervisory, Tamper, Temperature, Visible, Waterflow
'object_label'	Specifies the label of the device initiating the event.

Note: Outputs activated by a rule using the ServiceDevice event stay latched until the service test is canceled.

Example

```
{Flash LED on control/display module when device
activates}
[DEVICE_TEST]
SERV PULL 'IDC_RM_CKT<N:1-8>':
   FAST 'CDM_1_LED<N>';
```

ServiceGroup (SG)

Use the ServiceGroup event to activate a rule when an authorized service technician activates any device in a Service group under test. The ServiceGroup event is used to program a single response for the entire Service group in order to perform a one-man test.

The ServiceDevice event requires that you specify a device type or an object label, or both.

ServiceGroup *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: ServiceGroup
'object_label'	Specifies the label of the service group initiating the event.

Notes

Outputs activated by a rule using the ServiceGroup event automatically restore after a programmed delay period.

You must include a delay command at the end of a rule activated by the ServiceGroup event with a delay value sufficient to perform the one-man test

Example

```
{Flash LED on control/display module when
ServiceGroup event occurs}
[SERVICE_GROUP_RESPONSE]
SG 'SERVICE_GROUP_<N:1-8>':
   ON VIS 'LVL<N>_STROBES',
   DLYR 300;
```

ServiceGroupActive (SGA)

Use the ServiceGroupActive event to activate a rule when an authorized service technician starts a test on a service group from the 3-LCD module.

The ServiceGroupActive event does not require that you specifya device type or an object label.

ServiceGroupActive :

Example

```
{Light LED on control/display module when any
service group is in test mode}

[SERVICE_GROUP_ACTIVE_RESPONSE]

SGA:

   STEADY 'CDM_1_LED1';
```

SprinklerSupervisory (SPSUP)

Use the SprinklerSupervisory event to activate a rule when the open input to a device or circuit that supervises a component of the sprinkler system closes. Typically, you use the SprinklerSupervisory event to provide local and off-premises notification of sprinkler system status.

The SprinklerSupervisory event requires that you specify a device type or an object label, or both.

SprinklerSupervisory *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Gatevalve, Power, SprinklerSupervisory, Tamper, Temperature
'object_label'	Specifies the label of the device initiating the event.

Note: The Supervisory event can be used with a greater number of device types and should be used in place of the SprinklerSupervisory event. Any existing rules using the SprinklerSupervisory event will still function properly and should not be rewritten.

Example

```
{Light LED on control/display module when any
tamper switch is activated}

[TAMPER_LED]

SPSUP TAMP 'FLOOR*':

   STEADY -LOW 'TAMPER_LED';
```

Startup (STUP)

Use the Startup event to activate a rule when the panel is initially powered up or when an operator initiates the Restart command from the 3-LCD module. Typically, you use the Startup event to program the initial operating state of system components or functions.

The Startup event does not require that you specify a device type or an object label.

Startup :

Example

```
{Disable HVAC switches on startup}
[STARTUP]
STUP:
   DISABLE -LOW SW '*HVAC*';
```

StationActivation (STACT)

Use the StationActivation event to activate a rule when a patrol guard activates a patrol tour station. Typically, you use the StationActivation event to monitor a patrol guard's progress through the tour.

The StationActivation event requires that you specify a device type or an object label, or both.

StationActivation *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Guard
'object_label'	Specifies the label of the object triggering the event.

Example

```
{Light LED on control/display module when any
tamper guard station is activated}

[GUARD_STATION_OK]

STACT 'STATION<N:1-10>':

   STEADY -LOW 'STATION<N>_OK';
```

Supervisory (SUP)

Use the Supervisory event to activate a rule when any point on a panel or any panel in the same network routing group changes to the supervisory state.

The Supervisory event requires that you specify a device type or an object label, or both.

Supervisory *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: And, Gatevalve, Matrix, Power, SprinklerSupervisory, Supervisory, Tamper, Temperature
'object_label'	Specifies the label of the object triggering the event.

Note: The Supervisory event can be used with a greater number of device types and should be used in place of the SprinklerSupervisory event. Any existing rules using the SprinklerSupervisory event will still function properly and should not be rewritten.

Example

```
[AND_1_IN_SUP]
SUP AND 'AND_GROUP1':
   STEADY -LOW 'DISPLAY_1_LED1';
```

Switch (SW)

Use the Switch event to activate a rule when an operator presses a control/display module switch.

The Switch event requires that you specify a device type or an object label, or both.

Switch *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: Switch
'object_label'	Specifies the label of the switch initiating the event.

Note: Any commands listed in an output statement activated by the Switch event are automatically assigned a high priority.

Example

```
{Light all LEDs on control/display module while
lamptest switch is pressed}
[REMOTE_LAMPTEST]
SW 'LAMPTEST_SWITCH':
   LAMP;
```

TimeControl (TIME)

Use the TimeControl event to activate a rule when a specific combination of days, dates, and/or time of day occurs. Typically, you use the TimeControl event to program output responses that hold and release doors or change smoke detector sensitivity.

The TimeControl event requires that you specify a device type or an object label, or both.

TimeControl *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: TimeControl
'object_label'	Specifies the label of the time control initiating the event.

Example

```
{Hold all doors at 6:30 PM}
[DOOR_LOCK]
TIME 'LOCKDOORS':
   HOLD -LOW DOOR '*DOORLOCKS*';
```

Trouble (TRB)

Use the Trouble event to activate a rule when any point on a panel or any panel in the same network routing group changes to the trouble state.

The Trouble event requires that you specify a device type or an object label, or both.

Trouble *device_type 'object_label'* :

Parameter	Description
device_type	Specifies the device type of the device initiating the event. Valid device types are: And, Audible, CommFailure, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Emergency, ExtDBIncompatibility, Failsafe, FanControl, FanFeedback, Firephone, Gatevalve, GenAlarm, Gensmoke, Guard, Heat, LoopControllerResetExt, Matrix, Monitor, NSCommonAlarmOutput, NSCommonMonitorOutput, NSCommonSupervisoryOutput, NSCommonTroubleOutput, PanelCommFault, Power, Pull, RebootFault, Security, ServiceDeviceSupervision, Smoke, SmokeVfy, SprinklerSupervisory, StageOne, StageTwo, SupervisedOutput, Supervisory, Tamper, TaskFailure, Temperature, UserTrouble, Visible, Waterflow, Zone
'object_label'	Specifies the label of the device initiating the event.

Example

```
{Energize the remote common trouble relay}
[TROUBLE_LED]
TRB '*':
ON SUP 'TROUBLE_RELAY';
```

TwoStageTimerActivation (2STAGEA)

Use the TwoStageTimerActivation event to activate a rule when a panel's two-stage alarm timer starts counting down. Typically, you use the TwoStageTimerActivation event in jurisdictions that require some time delay in order to verify the source of the alarm before sounding a general alarm.

The TwoStageTimerActivation event does not require that you specify a device type or an object label.

TwoStageTimerActivation :

Note: The project parameters' setting determines the two-stage timer's count value.

Example

```
{Flash LED on control/display module when the
two-stage timer starts}

[2STAGE_TIMER_ACTIVE]

2STAGEA:

   FAST -LOW '2STAGE_ACTIVE_LED';
```

TwoStageTimerExpiration (2STAGETO)

Use the TwoStageTimerExpiration event to activate a rule when a panel's two-stage alarm timer expires. Typically, you use the TwoStageTimerExpiration event to sound a general alarm in jurisdictions that require some time delay to verify of the source of the alarm.

The TwoStageTimerExpiration event does not require that you specify a device type or an object label.

TwoStageTimerExpiration :

Note: The project parameters setting determines the two-stage timer's count value.

Example

```
{Flash LED on control/display module when the
two-stage timer ends}

[2STAGE_TIMER_EXPIRED]

2STAGETO:

   FAST -LOW '2STAGE_EXPIRE_LED';
```

Chapter 3

Output commands

Summary

This chapter provides an alphabetical reference of the commands used to program the system. Optional parameters are shown in *italics*.

Content

AlarmSilence (AS)

Use the AlarmSilence command to activate a panel's Alarm Silence function from a control/display module switch. By programming a control/display module switch to execute the AlarmSilence command, an operator can silence alarm notification signals from panels that do not contain a 3-LCD module.

The AlarmSilence command only requires that you specify a cabinet label or a routing label.

AlarmSilence 'cabinet_label' ;
– or –
AlarmSilence 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Notes

You must configure the switch used to execute the AlarmSilence command as a momentary switch.

Project parameter settings determine whether the Alarm Silence function silences only audible, or audible and visible notification appliance circuits.

The system Alarm Silence function treats devices assigned the CommonAlarmOutput device type as audibles.

Wildcards may be used in the cabinet label but not in the routing label.

Example

```
{Initiate the Alarm Silence function from a
remote panel}
[REMOTE_ALARM_SILENCE]
SW 'B1_C1_REM_ALRM_SIL':
   AS 'ALL_CABINETS';
```

AlternateLanguage (ALTL)

Use the AlternateLanguage command to change the language that the 3-LCD module uses to display text from a control/display module switch. By programming a control/display module switch to execute the AlternateLanguage command, an operator can control bilingual operation of the panel. You can also use to a time control to automatically switch between the primary and secondary language.

The AlternateLanguage command does not require that you specify a device type or an object label.

AlternateLanguage ;

Notes

You must configure the switch used to execute the AlternateLanguage command as a momentary switch.

The AlternateLanguage command only switches languages on the panel that contains the switch programmed to execute the command.

Device event messages are not affected by the Language settings and are displayed as they are entered in the SDU. You can customize the SDU to change the target language used for entering device event messages.

Example

```
{Change character set on display from
control/display module}
[ACTIVATE_ALTERNATE_LANGUAGE]
SW 'B1_C1_ALT_LANG':
   ALTL;
```

AlternateMsgOff (ALTMOFF)

Use the AlternateMsgOff command to activate the primary routing settings that a panel uses for routing device event messages.

The AlternateMsgOff command only requires that you specify a cabinet label or a routing label.

AlternateMsgOff 'cabinet_label' ;
– or –
AlternateMsgOff 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Note: Wildcards may be used in the cabinet label but not in the routing label.

Example

```
{Change message routing from control/display
module to primary}

[PRIMARY_MSG_ON]

SW 'B1_C1_PRI_MSGON':

   ALTMOFF 'All_Cabinets';
```

AlternateMsgOn (ALTMON)

Use the AlternateMsgOn command to activate the alternate routing settings that a panel uses for routing device event messages.

The AlternateMsgOn command only requires that you specify a cabinet label or a routing label.

AlternateMsgOn 'cabinet_label' ;
– or –
AlternateMsgOn 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Note: Wildcards may be used in the cabinet label but not in the routing label.

Example

```
{Switch to alternate message routing
configuration during non-regular work hours}

[ALTERNATE_MSG_ON]

TIME 'MON_FRI_NONREG_HOURS':

   ALTMON 'All_Cabinets';
```

AlternateSensitivityOff (ALTSOFF)

Use the AlternateSensitivityOff command to load the primary sensitivity and alarm verification settings into every smoke detector in the system. Typically, the primary settings are used during normal business hours when the protected premises is occupied.

The AlternateSensitivityOff command does not require that you specify a device type or an object label.

AlternateSensitivityOff ;

Example

```
{Switch smoke detectors to primary sensitivity
level during regular work hours}

[ALT_SENS_OFF]

TIME 'MON_FRI_REGULAR_HOURS':

   ALTSOFF;
```

Tip: Use the RemoteAltSensitivityOff command for greater control over smoke detector selection. The RemoteAltSensitivityOff command allows you to load primary sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels.

AlternateSensitivityOn (ALTSON)

Use the AlternateSensitivityOn command to load the alternate sensitivity and alarm verification settings into every smoke detector in the system. Typically, the alternate settings are used during off-normal business hours when the protected premises is unoccupied.

The AlternateSensitivityOn command does not require that you specify a device type or an object label.

AlternateSensitivityOn ;

Example

```
{Switch smoke detectors to alternate sensitivity
level setting during non-regular work hours}

[ALT_SENS_ON]

TIME 'MON_FRI_AFTER_HOURS':

   ALTSON;
```

Tip: Use the RemoteAltSensitivityOn command for greater control over smoke detector selection. The RemoteAltSensitivityOn command allows you to load alternate sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels.

AmpOff

Use the AmpOff command to turn off an amplifier's audio output and remove the connected audio channel from its input. The AmpOff command contains two parts. The first part of the command (AmpOff) identifies the amplifier and the second part (to) identifies the audio channel.

The AmpOff command only works on devices with the Amp device type. You are required to specify the amplifier label and the channel label. Specifying a priority value is optional.

AmpOff *priority* 'amp_label' to 'channel_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'amp_label'	Specifies the label of the amplifier module to turn off.
'channel_label'	Specifies the label of the audio channel to remove from the amplifier input.

Example

```
{Turn amplifiers off and deselect the page
channel}

[PAGE_OFF_SWITCHES]

SW 'LVL<N:1-10>_PAGE_OFF':

   AMPOFF -LOW 'LEVEL<N>_AMP' TO
'CH_PAGE_01_08';
```

AmpOn

Use the AmpOn command to turn on an amplifier's audio output and connect an audio channel to its input. The AmpOn command contains two parts. The first part of the command (AmpOn) identifies the amplifier and the second part (to) identifies the audio channel.

The AmpOn command only works on devices with the Amp device type. You are required to specify the amplifier label and the channel label. Specifying a priority value is optional.

AmpOn *priority* 'amp_label' to 'channel_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'amp_label'	Specifies the label of the amplifier module to turn on.
'channel_label'	Specifies the label of the audio channel to connect to the amplifier input.

Example

```
{Turn amplifiers on and select the page channel}
[PAGE_ON_SWITCHES]
SW 'LVL<N:1-10>_PAGE_ON':
   AMPON -LOW 'LEVEL<N>_AMP' TO 'CH_PAGE_01_08';
```

Close

Use the Close command to turn off an output circuit or relay that connects to the control mechanism used to close a damper.

The Close command only works on devices with the DamperControl device type. You are required to specify the object label. Specifying a priority value is optional.

Close *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit off (nonactive) the control mechanism closes the damper.

Example

```
{Close dampers from control/display module
switch}
[DAMPER_CLOSE_SWITCHES]
SW 'LVL<N:1-10>_DMPR_CLOSE':
   CLOSE -LOW DAMP 'DMPR_RELAY_LVL<N>';
```

CommonAlarmOff (CAOFF)

Use the CommonAlarmOff command to turn off a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs.

The CommonAlarmOff command only works on devices with the CommonAlarmOutput device type. You are required to specify the object label. Specifying a priority value is optional.

CommonAlarmOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: Active common alarm outputs can not be turned off using the CommonAlarmOff command.

Example

```
[COMMON_ALARM_OFF_SWITCHES]
SW 'COM_ALARM_OFF_LVL<N:1-10>':
   CAOFF -LOW 'LVL<N>_COM_ALARM_CKT*';
```

CommonAlarmOn (CAON)

Use the CommonAlarmOn command to turn on a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs.

The CommonAlarmOn command only works on devices with the CommonAlarmOutput device type. You are required to specify the object label. Specifying a priority value is optional.

CommonAlarmOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[COMMON_ALARM_ON_SWITCHES]
SW 'COM_ALARM_ON_LVL<N:1-10>':
   CAON -LOW 'LVL<N>_COM_ALARM_CKT*';
```

CommonMonitorOff (CMOFF)

Use the CommonMonitorOff command to turn off a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs.

The CommonMonitorOff command only works on devices with the CommonMonitorOutput device type. You are required to specify the object label. Specifying a priority value is optional.

CommonMonitorOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Notes

Active common monitor outputs can not be turned off using the CommonMonitorOff command.

You can not use any device to turn off a common monitor output that upon activation produces a monitor event. For example, you can not use a control/display module switch.

Example

```
[COMMON_MON_OFF_RESPONSE]
ALARM GENALARM:
   CMOFF -LOW 'EVAC_DOOR';
```

CommonMonitorOn (CMON)

Use the CommonMonitorOn command to turn on a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs.

The CommonMonitorOn command only works on devices with the CommonMonitorOutput device type. You are required to specify the object label. Specifying a priority value is optional.

CommonMonitorOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[COMMON_MON_ON_RESPONSE]
ALARM GENALARM:
   CMON -LOW 'COM_MON_CKT_LVL*';
```

CommonSupervisoryOff (CSOFF)

Use the CommonSupervisoryOff command to turn off a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs.

The CommonSupervisoryOff command works on devices with the CommonSupervisoryOutput device type. You are required to specify the object label. Specifying a priority value is optional.

CommonSupervisoryOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: Active common supervisory outputs can not be turned off using the CommonSupervisoryOff command.

Example

```
[COMMON_SUP_OFF_SWITCHES]
SW 'COM_SUP_OFF_LVL<N:1-10>':
   CSOFF -LOW 'LVL<N>_COM_SUP_CKT*';
```

CommonSupervisoryOn (CSON)

Use the CommonSupervisoryOn command to turn on a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs.

The CommonSupervisoryOn command works on devices with the CommonSupervisoryOutput device type. You are required to specify the object label. Specifying a priority value is optional.

CommonSupervisoryOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[COMMON_SUP_ON_SWITCHES]
SW 'COM_SUP_ON_LVL<N:1-10>':
   CSON -LOW 'LVL<N>_COM_SUP_CKT*';
```

Delay (DLY)

Use the Delay command to insert a time delay between rule output command statements when a rule activates and restores.

The Delay command only requires that you specify the delay value.

Delay delay_value ;

Parameter	Description
delay_value	Specifies the length of the delay from 1 to 32767 seconds, inclusive.

Example

```
{Turn amps on at startup to accept the General
channel}
[STARTUP_AMPS]
STUP:
   AMPON 'LVL_5_AMP' TO 'CH_GEN_01_08',
   DLY 10,
   AMPON 'LVL_6_AMP' TO 'CH_GEN_01_08';
```

DelayActivate (DLYA)

Use the DelayActivate command to insert a time delay between rule output command statements only when the rule activates. The delay is not inserted when the rule restores.

The DelayActivate command only requires that you specify the delay value.

DelayActivate delay_value ;

Parameter	Description
delay_value	Specifies the length of the delay from 1 to 32767 seconds, inclusive.

Example

```
{Supply and pressure fan alarm response}
[ALARM_FAN]
ALARM GENSMOKE:
   FANOFF 'SUPPLY_FAN',
   DLYA 15,
   FANON 'PRESSURE_FAN1';
```

DelayRestore (DLYR)

Use the DelayRestore command to insert a time delay between rule output command statements only when the rule restores. The delay is not inserted when the rule activates.

The DelayRestore command only requires that you specify the delay value.

DelayRestore delay_value ;

Parameter	Description
delay_value	Specifies the length of the delay from 1 to 32767 seconds, inclusive.

Example

```
{Supply and pressure fan alarm response}
[ALARMFAN]
ALARM GENSMOKE:
   FANON 'PRESSURE_FAN1',
   DLYR 15,
   FANON 'PRESSURE_FAN2';
```

Disable

Use the Disable command to inhibit the automatic or manual control of a system hardware component, circuit, or logic group.

The Disable command requires that you specify a device type or an object label, or both.

Disable *device_type 'object_label'* ;

Parameter	Description
device_type	Specifies the device type of the device responding to the command. Valid device types are: 12SW/12LED, 12SW/24LED, 24LED, 3-AADC, 3-ASU, 3-DSDC, 3-FTCU, 3-IDC8/4, 3-OPS, 3SW/3LEDX6, Amp, And, Audible, CheckIn, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Emergency, FanControl, FanFeedback, Firephone, Gatevalve, GenAlarm, GenSmoke, Guard, GuardPatrol, Heat, LED, LocalAlarm, LocalMonitor, LocalRelay, LocalTrouble, Matrix, Monitor, NonsupervisedOutput, NSCommonAlarmOutput, NSCommonMonitorOutput, NSCommonSupervisoryOutput, NSCommonTroubleOutput, Power, Pull, Security, Smoke, SmokeVfy, SprinklerSupervisory, StageOne, StageTwo, SupervisedOutput, Supervisory, Switch, Tamper, Temperature, Text, TimeControl, Visible, Waterflow, Zone
'object_label'	Specifies the label of the output device responding to the command.

Note: Disabling a Zone group disables the group and all the devices in the group.

Example

```
[CIRCUIT_DISABLE]
SW 'PANEL_1_SW1':
   DISABLE DSDC 'SIGA_CIRCUIT_1';
```

Drill

Use the Drill command to activate a panel's Drill function from a control/display module switch. By programming a control/display module switch to execute the Drill command, an operator can turn on alarm notification signals from panels that do not contain a 3-LCD module.

The Drill command only requires that you specify a cabinet label or routing label.

Drill 'cabinet_label' ;
– or –
Drill 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Notes

You must configure the switch used to execute the Drill command as a momentary switch.

Project parameter settings determine whether the Drill function activates only audible, or audible and visible notification appliance circuits. The Drill function activates notification appliance circuits but does not put the panel into alarm.

Wildcards may be used in the cabinet label but not in the routing label.

Example

```
[REM_DRILL_CNTRL]
SW 'B1_CAB<N:1-5>_DRILL':
   DRILL 'CAB<N>';
```

Enable

Use the Enable command to allow automatic or manual control of a system hardware component, circuit, or logic group.

The Enable command requires that you specify a device type or an object label, or both.

Enable *device_type 'object_label'* ;

Parameter	Description
device_type	Specifies the device type of the device responding to the command. Valid device types are: 12SW/12LED, 12SW/24LED, 24LED, 3-AADC, 3-ASU, 3-DSDC, 3-FTCU, 3-IDC8/4, 3-OPS, 3SW/3LEDX6, Amp, And, Audible, CheckIn, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DamperFeedback, DoorControl, DoorFeedback, Emergency, FanControl, FanFeedback, Firephone, Gatevalve, GenAlarm, GenSmoke, Guard, GuardPatrol, Heat, LED, LocalAlarm, LocalMonitor, LocalRelay, LocalTrouble, Matrix, Monitor, NonsupervisedOutput, NSCommonAlarmOutput, NSCommonMonitorOutput, NSCommonSupervisoryOutput, NSCommonTroubleOutput, Power, Pull, Security, Smoke, SmokeVfy, SprinklerSupervisory, StageOne, StageTwo, SupervisedOutput, Supervisory, Switch, Tamper, Temperature, Text, TimeControl, Visible, Waterflow, Zone
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[CIRCUIT_ENABLE]
SW 'PANEL_1_SW1':
   ENABLE 3-DSDC 'SIGA_CIRCUIT_1';
```

Evacuation (EVAC)

Use the Evacuation command to activate a panel's Evacuation function from a control/display module switch. By programming a control/display module switch to execute the Evacuation command, an operator can put a panel into alarm and force the panel to activate its programmed alarm responses from panels that do not contain a 3-LCD module..

The Evacuation command only requires that you specify a cabinet label or routing label.

Evacuation 'cabinet_label' ;
– or –
Evacuation 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Note: Wildcards may be used in the cabinet label but not in the routing label.

Example

```
[REM_EVAC_CNTRL]
SW 'DISPLAY_1_SW1':
   EVAC 'All_Cabinets';
```

FanOff

Use the FanOff command to turn off an output circuit or relay that connects to the control mechanism used to turn off a fan.

The FanOff command only works on devices with the FanControl device type. You are required to specify the object label. Specifying a priority value is optional.

FanOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit off (nonactive) the control mechanism turns off the fan.

Example

```
[SUPPLY_FAN_CNTRL]
SW 'SUPPLY_FAN<N:1,2>_OFF':
   FANOFF 'SUPPLY_FAN<N>';
```

FanOn

Use the FanOn command to turn on an output circuit or relay that connects to the control mechanism used to turn on a fan.

The FanOn command only works on devices with the FanControl device type. You are required to specify the object label. Specifying a priority value is optional.

FanOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit on (active) the control mechanism turns on the fan.

Example

```
[SUPPLY_FAN_CNTRL]
SW 'SUPPLY_FAN<N:1,2>_ON':
   FANON -LOW 'SUPPLY_FAN<N>';
```

FastBlink (FAST)

Use the FastBlink command to turn a light-emitting diode on a control/display module on and off at a fast interval.

The FastBlink command only works on devices with the LED device type. You are required to specify the object label. Specifying a priority value is optional.

FastBlink *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
{Flash LED on control/display module when second
phase of the system reset process occurs}
[ANN_RESET_2ND_PHASE]
R2:
   FAST -LOW 'R2_LED';
```

GAInhibit (GAIN)

Use the GAInhibit command to stop a panel's two-stage timer from a control/display module switch. By programming a control/display module switch to execute the GAInhibit command, an operator can stop a panel's two-stage alarm timer before it expires and prevent the panel from sounding a general alarm.

The GAInhibit command only requires that you specify the cabinet label or the routing label.

GainInhibit 'cabinet_label' ;
– or –
GainInhibit 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Notes

You must configure the switch used to execute the GAInhibit command as a momentary switch.

Wildcards may be used in the cabinet label but not in the routing label.

Example

```
{Cancel the two-stage alarm timer}
[GAINHIBIT_CNTRL]
SW 'GAINHIBIT_SW':
   GAIN 'ALL_CABINETS';
```

HoldDoor (HOLD)

Use the HoldDoor command to turn on an output circuit or relay that connects to the control mechanism used to hold a door.

The HoldDoor command only works with devices assigned the DoorControl device type. You are required to specify the object label. Specifying a priority value is optional.

HoldDoor *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit on (active) the control mechanism holds the door.

Example

```
{Energize magnetic door holders from panel
control switch}

[MANUAL_DOOR_CNTRL]

SW 'PANEL_1_SW1':

   HOLD -HIGH 'DOORCONTROL_*';
```

LampTest (LAMP)

Use the LampTest command to activate the lamp test function from a control/display module switch. By programming a control/display module switch to execute the LampTest command, an operator can perform a lamp test on panels that do not contain a 3-LCD module.

The LampTest command does not require that you specify a device type or an object label.

LampTest ;

Notes

You must configure the switch used to execute the LampTest command as a momentary switch.

The LampTest command operates only on the panel containing the switch programmed to execute the command.

Example

```
[LAMPTEST]
SW 'LAMPTEST_ON':
   LAMP;
```

LEDOff

Use the LEDOff command to turn a light-emitting diode on a control/display module off.

The LEDOff command only works on devices with the LED device type. You are required to specify the object label. Specifying a priority value is optional.

LEDOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
{Turn Damper Open LEDs Off when closed limit
switch is activated}
[DAMPER_LIMIT_LED_OFF]
MON DAMPERFB 'FAN<N:1-5>_DAMP_CLIMIT':
   LEDOFF -LOW 'DMPR<N>_OPEN_LED';
```

MsgOff

Use the MsgOff command to stop broadcasting a voice message over the selected audio channel. The MsgOff command contains two parts. The first part of the command (MsgOff) identifies the message and the second part (to) identifies the audio channel.

The MsgOff command only works on devices with the Msg device type. You are required to specify the message label and the channel label. Specifying an ASU label or priority value is optional.

MsgOff *priority* 'msg_label' *from 'asu_label'* to 'channel_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'msg_label'	Specifies the label of the voice message to stop broadcasting.
'asu_label'	Specifies the label of the audio source unit providing the voice message.
'channel_label'	Specifies the label of the audio channel broadcasting the voice message.

Note: Only use the *from 'asu_label'* parameter with systems containing multiple audio source units.

Example

```
[EVAC_MSG_OFF_TEST]
SW 'FLOOR1_TEST':
   MSGOFF 'TEST_MESSAGE' FROM '3-ASU1' TO
'CH_EVAC_01_08';
```

MsgOn

Use the MsgOn command to start broadcasting a voice message over a selected audio channel. The MsgOn command contains two parts. The first part of the command (MsgOn) identifies the message and the second part (to) identifies the audio channel.

The MsgOn command only works on devices with the Msg device type. You are required to specify the message label and the channel label. Specifying an ASU label or priority value is optional.

MsgOn *priority* 'msg_label' *from 'asu_label'* to 'channel_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'msg_label'	Specifies the label of the voice message to start broadcasting.
'asu_label'	Specifies the label of the audio source unit providing the voice message.
'channel_label'	Specifies the label of the audio channel broadcasting the voice message.

Notes

The MsgOn command must be placed after the AmpOn command in a rule and selected to the same audio channel as the amplifier.

Only use the *from 'asu_label'* parameter with systems containing multiple audio source units.

Example

```
{Send test message to Floor 1 on EVAC channel
from panel control switch}

[EVAC_MSG_ON_TEST]

SW 'FLOOR1_TEST':

   AMPON 'LEVEL3_AMP' TO 'CH_EVAC_01_08',

   MSGON 'TEST_MESSAGE' FROM '3-ASU1' TO
'CH_EVAC_01_08';
```

NCClose

Use the NCClose command to turn on an output circuit or relay that connects to the control mechanism used to close a damper.

The NCClose command only works on devices with the DamperControl device type. You are required to specify the object label. Specifying a priority value is optional.

NCClose *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit on (active) the control mechanism closes the damper.

Example

```
{Close dampers from control/display module
switch}
[DAMPER_CLOSE_SWITCHES]
SW 'LVL<N:1-10>_DMPR_CLOSE':
   NCCLOSE -LOW 'DMPR_RELAY_LVL<N>';
```

NCFanOff

Use the NCFanOff command to turn on an output circuit or relay that connects to the control mechanism used to turn off a fan.

The NCFanOff command only works on devices with the FanControl device type. You are required to specify the object label. Specifying a priority value is optional.

NCFanOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit on (active) the control mechanism turns off the fan.

Example

```
[SUPPLY_FAN_CNTRL]
SW 'SUPPLY_FAN<N:1,2>_OFF':
   NCFANOFF -LOW 'SUPPLY_FAN<N>';
```

NCFanOn

Use the NCFanOn command to turn off an output circuit or relay that connects to the control mechanism used to turn on a fan.

The NCFanOn command only works on devices with the FanControl device type. You are required to specify the object label. Specifying a priority value is optional.

NCFanOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit off (nonactive) the control mechanism turns on the fan.

Example

```
[SUPPLY_FAN_CNTRL]
SW 'SUPPLY_FAN<N:1,2>_ON':
   NCFANON -LOW 'SUPPLY_FAN<N>';
```

NCHoldDoor (NCHOLD)

Use the NCHoldDoor command to turn off an output circuit or relay that connects to the control mechanism used to hold a door.

The NCHoldDoor command only works with devices with the DoorControl device type. You are required to specify the object label. Specifying a priority value is optional.

NCHoldDoor *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit off (nonactive) the control mechanism holds the door.

Example

```
{Energize magnetic door holders from panel
control switch}
[MANUAL_DOOR_CNTRL]
SW 'PANEL_1_SW1':
   NCHOLD -HIGH 'DOORCONTROL_*';
```

NCOpen

Use the NCOpen command to turn off an output circuit or relay that connects to the control mechanism used to open a damper.

The NCOpen command only works on devices with the DamperControl device type. You are required to specify the object label. Specifying a priority value is optional.

NCOpen *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit off (nonactive) the control mechanism opens the damper.

Example

```
{Open dampers from control/display module
switch}
[MANUAL_DAMPER_CNTRL]
SW 'PANEL_1_SW1':
   NCOPEN -HIGH 'DAMPERCONTROL_*';
```

NCReleaseDoor (NCRELEASE)

Use the NCReleaseDoor command to turn on an output circuit or relay that connects to the control mechanism used to release a door.

The NCReleaseDoor command only works on devices with the DoorControl device type. You are required to specify the object label. Specifying a priority value is optional.

NCReleaseDoor *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit on (active) the control mechanism releases the door.

Example

```
{Release door locks from panel control switch}
[DOOR_RELEASE_CNTRL]
SW 'PANEL_1_SW1':
   NCRELEASE -HIGH 'DOORCONTROL_*';
```

NSCommonAlarmOff (NSCAOFF)

Use the NSCommonAlarmOff command to turn off a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs.

The NSCommonAlarmOff command only works on devices with the NSCommonAlarmOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonAlarmOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Notes

Active nonsupervised common alarm outputs can not be turned off using the NSCommonAlarmOff command.

The NSCommonAlarmOff command does not affect the operation of the Form C common alarm relay on the 3-CPU1 module.

Example

```
[NSCOMMON_ALARM_OFF_SWITCHES]
SW 'NSCOM_ALARM_OFF_LVL<N:1-10>':
   NSCAOFF -LOW 'LVL<N>_NSCOM_ALARM_CKT*';
```

NSCommonAlarmOn (NSCAON)

Use the NSCommonAlarmOn command to turn on a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs.

The NSCommonAlarmOn command only works on devices with the NSCommonAlarmOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonAlarmOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[NSCOMMON_ALARM_ON_SWITCHES]
SW 'NSCOM_ALARM_ON_LVL<N:1-10>':
   NSCAON -LOW 'LVL<N>_NSCOM_ALARM_CKT*';
```

NSCommonMonitorOff (NSCMOFF)

Use the NSCommonMonitorOff command to turn off a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs.

The NSCommonMonitorOff command only works on devices with the NSCommonMonitorOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonMonitorOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Notes

Active nonsupervised common monitor outputs can not be turned off using the NSCommonMonitorOff command.

You can not use any device to turn off a nonsupervised common monitor output that upon activation produces a monitor event. For example, you can not use a control/display module switch.

Example

```
[NSCOMMON_MON_OFF_RESPONSE]
ALARM GENALARM:
   NSCMOFF -LOW 'NSCOM_MON_CKT_LVL*';
```

NSCommonMonitorOn (NSCMON)

Use the NSCommonMonitorOn command to turn on a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs.

The NSCommonMonitorOn command only works on devices with the NSCommonMonitorOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonMonitorOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[COMMON_MON_ON_RESPONSE]
ALARM GENALARM:
   CMON -LOW 'COM_MON_CKT_LVL*';
```

NSCommonSupervisoryOff (NSCSOFF)

Use the NSCommonSupervisoryOff command to turn off a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs.

The NSCommonSupervisoryOff command only works on devices with the NSCommonSupervisoryOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonSupervisoryOff *priority* 'object_label';

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Notes

Active nonsupervised common supervisory outputs can not be turned off using the NSCommonSupervisoryOff command.

The NSCommonSupervisoryOff command does not affect the operation of the Form C common supervisory relay on the 3-CPU1 module.

Example

```
[NSCOMMON_SUP_OFF_SWITCHES]
SW 'NSCOM_SUP_OFF_LVL<N:1-10>':
   NSCSOFF -LOW 'LVL<N>_NSCOM_SUP_CKT*';
```

NSCommonSupervisoryOn (NSCSON)

Use the NSCommonSupervisoryOn command to turn on a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs.

The NSCommonSupervisoryOn command only works on devices with the NSCommonSupervisoryOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonSupervisoryOn *priority* 'object_label';

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[NSCOMMON_SUP_ON_SWITCHES]
SW 'NSCOM_SUP_ON_LVL<N:1-10>':
   NSCSON -LOW 'LVL<N>_NSCOM_SUP_CKT*';
```

NSCommonTroubleOff (NSCTOFF)

Use the NSCommonTroubleOff command to turn off a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs.

The NSCommonTroubleOff command only works on devices with the NSCommonTroubleOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonTroubleOff *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Notes

Active nonsupervised common trouble outputs can not be turned off using the NSCommonTroubleOff command.

The NSCommonTroubleOff command does not affect the operation of the Form C common trouble relay on the 3-CPU1 module.

Example

```
[NSCOMMON_TRB_OFF_SWITCHES]
SW 'NSCOM_TRB_OFF_LVL<N:1-10>':
   NSCTOFF -LOW 'LVL<N>_NSCOM_TRB_CKT*';
```

NSCommonTroubleOn (NSCTON)

Use the NSCommonTroubleOn command to turn on a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs.

The NSCommonTroubleOn command only works on devices with the NSCommonTroubleOutput device type. You are required to specify the object label. Specifying a priority value is optional.

NSCommonTroubleOn *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
[NSCOMMON_TRB_ON_SWITCHES]
SW 'NSCOM_TRB_ON_LVL<N:1-10>':
   NSCTON -LOW 'LVL<N>_NSCOM_TRB_CKT*';
```

Off

Use the Off command to turn a system hardware component, circuit, or logic group off.

The Off command requires that you specify a device type or an object label, or both. Specifying a priority value is optional.

Off *priority device_type 'object_label'* ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
device_type	Specifies the device type of the device responding to the command. Valid device types are: Audible, CheckIn, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DoorControl, FanControl, Firephone, LED, NonsupervisedOutput, NSCommonAlarmOutput, NSCommonMonitorOutput, NSCommonSupervisoryOutput, NSCommonTroubleOutput, ServiceGroup, SupervisedOutput, Visible
'object_label'	Specifies the label of the output device responding to the command.

Example

```
{Turn off all horns in the building from
control/display module}

[REMOTE_HORN_OFF]

SW 'PANEL_1_SW1':

   OFF -LOW AUD 'HORN*';
```

OffGuard

Use the OffGuard command to cancel a specific guard patrol tour. The OffGuard command contains two parts. The first part of the command (OffGuard) identifies the GuardPatrol group and the second part (Route) identifies the route number.

The OffGuard command only works on devices with the GuardPatrol device type. You are required to specify the guard label and the route identifier number.

OffGuard 'guard_label' Route route_id

Parameter	Description
'guard_label'	Specifies the label of the Guard Patrol group responding to the command.
route_id	Specifies the route number of the guard patrol tour.

Example

```
{Deactivate individual guard patrol route from
remote panel toggle switch}

[GUARDPATROL_ROUTE_OFF]

SW 'PANEL_2_SW1':

   OFFGUARD 'GUARD_PATROL_GROUP1' ROUTE 1;
```

On

Use the On command to turn a system hardware component, circuit, or logic group on.

The On command requires that you specify a device type or an object label, or both. Specifying a priority value is optional.

On *priority device_type 'object_label'* ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
device_type	Specifies the device type of the device responding to the command. Valid device types are: Audible, CheckIn, CommonAlarmOutput, CommonMonitorOutput, CommonSupervisoryOutput, DamperControl, DoorControl, FanControl, Firephone, LED, NonsupervisedOutput, NSCommonAlarmOutput, NSCommonMonitorOutput, NSCommonSupervisoryOutput, NSCommonTroubleOutput, ServiceGroup, SupervisedOutput, Visible
'object_label'	Specifies the label of the output device responding to the command.

Example

```
{Turn on all horns in the building from
control/display module}

[REMOTE_HORN_ON]

SW 'PANEL_1_SW2':
   ON -HIGH AUD 'HORN*';
```

OnGuard

Use the OnGuard command to start a specific guard patrol tour. The OnGuard command contains two parts. The first part of the command (OnGuard) identifies the GuardPatrol group and the second part (Route) identifies the route number.

The OnGuard command only works on devices with the GuardPatrol device type. You are required to specify the guard label and the route identifier number.

OnGuard 'guard_label' Route route_id

Parameter	Description
'guard_label'	Specifies the label of the Guard Patrol group responding to the command.
route_id	Specifies the route number of the guard patrol tour.

Example

```
{Activate guard patrol route from
control/display module switch }

[Group1GuardPatrolRoutesOn]

SW 'Panel_1_SW1':

   OnGuard 'GUARD_PATROL_Group1' Route 1;
```

Open

Use the Open command to turn on an output circuit or relay that connects to the control mechanism used to open a damper.

The Open command only works on devices with the DamperControl device type. You are required to specify the object label. Specifying a priority value is optional.

Open *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit on (active) the control mechanism opens the damper.

Example

```
{Open dampers from control/display module
switch}
[MANUAL_DAMPER_CNTRL]
SW 'PANEL_1_SW1':
   OPEN -HIGH 'DAMPERCONTROL_*';
```

ReleaseDoor (RELEASE)

Use the ReleaseDoor command to turn off an output circuit or relay that connects to the control mechanism used to release a door.

The ReleaseDoor command only works on devices with the DoorControl device type. You are required to specify the object label. Specifying a priority value is optional.

ReleaseDoor *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Note: The output circuit must be wired to the control mechanism so that with the output circuit off (nonactive) the control mechanism releases the door.

Example

```
{Release door locks from panel control switch}
[DOOR_RELEASE_CNTRL]
SW 'PANEL_1_SW1':
   RELEASE -HIGH 'DOORCONTROL_*';
```

RemoteAltSensitivityOff (RASOFF)

Use the RemoteAltSensitivityOff command to load the primary sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels. Typically, the primary settings are used during normal business hours when the protected premises is occupied.

The RemoteAltSensitivityOff command only requires that you specify a cabinet label or a routing label.

RemoteAltSensitivityOff 'cabinet_label' ;
– or –
RemoteAltSensitivityOff 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Note: Wildcards may be used in the cabinet label but not in the routing label.

Example

```
{Automatically switch smoke detectors only on
Cab2 to primary sensitivity level during regular
work hours}

[PRIMARY_SENSITIVITY_CAB2]

TIME 'MON-FRI_REGULAR_HOURS':

   RASOFF 'CAB_2';
```

RemoteAltSensitivityOn (RASON)

Use the RemoteAltSensitivityOn command to load the alternate sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels. Typically, the alternate settings are used during normal business hours when the protected premises is unoccupied.

The RemoteAltSensitivityOn command only requires that you specify a cabinet label or a routing label.

RemoteAltSensitivityOn 'cabinet_label' ;
– or –
RemoteAltSensitivityOn 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Note: Wildcards may be used in the cabinet label but not in the routing label.

Example

```
{Automatically switch smoke detectors to
alternate sensitivity level only on Cab2 during
non-regular work hours}

[ALTERNATE_SENSITIVITY_CAB2]

TIME 'MON-FRI_AFTER_HOURS':
   RASON 'CAB_2';
```

Reset

Use the Reset command to activate a panel's Reset function from a control/display module switch. By programming a control/display module switch to execute the Reset command, an operator can reset the system from panels that do not contain a 3-LCD module.

The Reset command only requires that you specify a cabinet label or a routing label.

Reset 'cabinet_label' ;
– or –
Reset 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

Notes

You must configure the switch used to execute the Reset command as a momentary switch.

Wildcards may be used in the cabinet label but not in the routing label.

Example

```
[REMOTE_RESET]
SW 'REM_RESET_SWITCH':
   RESET 'All_Cabinets';
```

SlowBlink (SLOW)

Use the SlowBlink command to turn a light-emitting diode on a control/display module on and off at a slow interval.

The SlowBlink command only works on devices with the LED device type. You are required to specify the object label. Specifying a priority value is optional.

SlowBlink *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
{Flash LED on control/display module to indicate
FirePhone device type is activated}

[ANN_RESET_2ND_PHASE]

R2:

   SLOW -LOW 'R2_LED';
```

Steady

Use the Steady command to turn a light-emitting diode on a control/display module LED on and have it remain on.

The Steady command only works on devices with the LED device type. You are required to specify the object label. Specifying a priority value is optional.

Steady *priority* 'object_label' ;

Parameter	Description
priority	Specifies the order of importance this command has over other commands affecting the same output device. Valid priority levels are: -low, -high, -latch, -set
'object_label'	Specifies the label of the output device responding to the command.

Example

```
{Light LED on control/display module to indicate
Firephone device type is activated}

[PHONE_CONNECT_REQ]

CALLIN:

   STEADY -LOW 'DISPLAY_1_LED_1';
```

TroubleSilence (TS)

Use the TroubleSilence command to activate a panel's Trouble Silence function from a control/display module switch. By programming a control/display module switch to execute the TroubleSilence command, an operator can turn off a panel's trouble buzzer from panels that do not contain a 3-LCD module.

The TroubleSilence command only requires that you specify a cabinet label or a routing label.

TroubleSilence 'cabinet_label' ;
– or –
TroubleSilence 'routing_label' ;

Parameter	Description
'cabinet_label'	Specifies the label of the panel responding to the command.
'routing_label'	Specifies the label of the network routing group responding to the command.

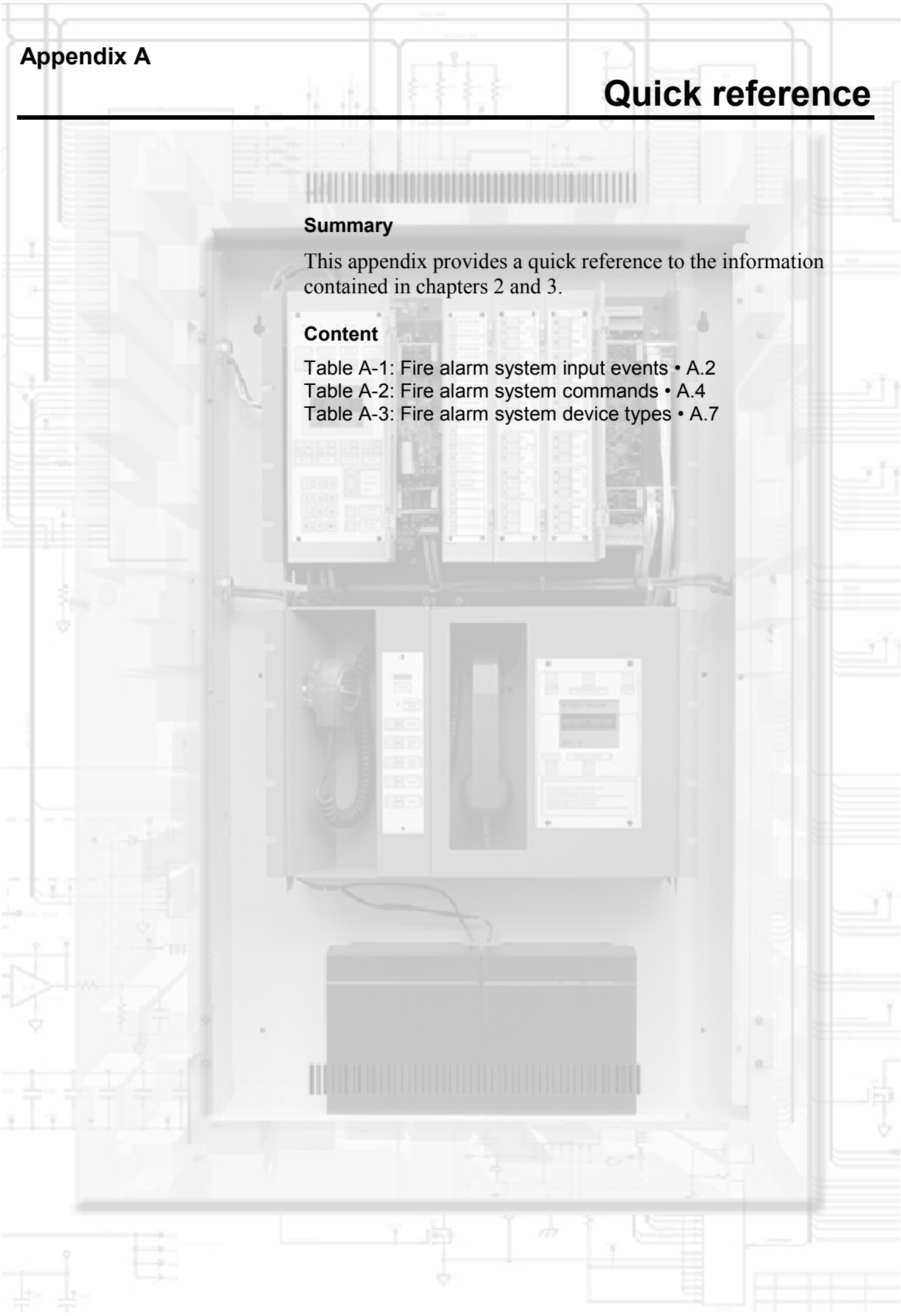
Note: You must configure the switch used to execute the TroubleSilence command as a momentary switch.

Example

```
[REMOTE_TROUBLE_SILENCE]
SW 'B1_C1_REM_TRBL_SIL':
   TS 'ALL_CABINETS';
```

Appendix A

Quick reference



Summary

This appendix provides a quick reference to the information contained in chapters 2 and 3.

Content

Table A-1: Fire alarm system input events

Choose this event	To activate a rule when
Acknowledge	An operator acknowledges an event displayed on the 3-LCD module
Alarm	Any point on a panel or any panel in the same network routing group changes to the alarm state
AlarmSilence	An operator presses a switch that starts a panel's Alarm Silence function
AlarmVerify	A smoke detector starts its smoke verification cycle
AllCall	An operator presses the All Call or All Call Minus switch on the 3-ASU
CallIn	A fire safety professional plugs a handset into a firefighter's telephone jack
Drill	An operator presses a switch that starts a panel's Drill function
Emergency	A member of a Check-in group activates their check-in device anytime outside of their check-in period or a second time during their check-in period
Evacuation	An operator presses a control/display module switch programmed to execute a panel's Evacuation function
FirstAlarm	The first time that any point on a panel or any panel in the same network routing group changes to the alarm state
FirstDisable	The first time that any point on a panel or any panel in the same network routing group changes to the disabled state
FirstMonitor	The first time that any point on a panel or any panel in the same network routing group changes to the monitor state
FirstSupervisory	The first time that any point on a panel or any panel in the same network routing group changes to the supervisory state
FirstTrouble	The first time that any point on a panel or any panel in the same network routing group changes to the trouble state
GroundFault	A rail module detects a ground fault on its field wiring
GuardPatrol	A patrol guard fails to activate a patrol tour station at the proper time
LocalAlarm	A rail module's LocalAlarm pseudo point goes active
LocalMonitor	A rail module's LocalMonitor pseudo point goes active
LocalTrouble	A rail module's LocalTrouble pseudo point goes active
Monitor	Any point on a panel or any panel in the same network routing group changes to the monitor state
R1	The first phase of the 3-phase reset cycle starts after an operator presses the Reset switch on the 3-LCD module

Table A-1: Fire alarm system input events

Choose this event	To activate a rule when
R2	The second phase of the 3-phase reset cycle starts
R3	The third phase of the 3-phase reset cycle starts
RelayConfirmation	A control relay indicates that its electrical contacts have switched positions
Reset	An operator presses a switch that starts a panel's Reset function
Security	The open input to a device or circuit that monitors a supervisory or tamper switch closes
Service	An authorized service technician activates a device in a Service group under test
ServiceGroup	An authorized service technician activates any device in a Service group under test
ServiceGroupActive	An authorized service technician starts a test on a service group from the 3-LCD module
SprinklerSupervisory	The open input to a device or circuit that supervises a component of the sprinkler system closes
Startup	The panel is initially powered up or when an operator initiates the Restart command from the 3-LCD module
StationActivation	A patrol guard activates a patrol tour station
Supervisory	Any point on a panel or any panel in the same network routing group changes to the supervisory state
Switch	An operator presses a control/display module switch
TimeControl	A specific combination of days, dates, and/or time of day occurs
Trouble	Any point on a panel or any panel in the same network routing group changes to the trouble state
TwoStageTimerActivation	A panel's two-stage alarm timer starts counting down
TwoStageTimerExpiration	A panel's two-stage alarm timer expires

Table A-2: Fire alarm system commands

Choose this command	To do this
AlarmSilence	Activate a panel's Alarm Silence function from a control/display module momentary switch in order to silence alarm notification signals from panels that do not contain a 3-LCD
AlternateLanguage	Change the language that the 3-LCD uses to display text from a control/display module momentary switch
AlternateMsgOff	Activate the primary routing settings that a panel uses for routing device event messages
AlternateMsgOn	Activate the alternate routing settings that a panel uses for routing device event messages
AlternateSensitivityOff	Load the primary sensitivity and alarm verification settings into every smoke detector in the system
AlternateSensitivityOn	Load the alternate sensitivity and alarm verification settings into every smoke detector in the system
AmpOff	Turn off an amplifier's audio output and remove the connected audio channel from its input
AmpOn	Turn on an amplifier's audio output and connect an audio channel to its input
Close	Turn off an output circuit or relay that connects to the control mechanism used to close a damper
CommonAlarmOff	Turn off a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs
CommonAlarmOn	Turn on a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs
CommonMonitorOff	Turn off a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs
CommonMonitorOn	Turn on a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs
CommonSupervisoryOff	Turn off a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs
CommonSupervisoryOn	Turn on a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs
Delay	Insert a time delay between rule output command statements when a rule activates and restores
DelayActivate	Insert a time delay between rule output command statements only when the rule activates

Table A-2: Fire alarm system commands

Choose this command	To do this
DelayRestore	Insert a time delay between rule output command statements only when the rule restores
Disable	Inhibit the automatic or manual control of a system hardware component, circuit, or logic group
Drill	Activate a panel's Drill function from a control/display module momentary switch in order to turn on alarm notification signals from panels that do not contain a 3-LCD
Enable	Allow automatic or manual control of a system hardware component, circuit, or logic group
Evacuation	Activate a panel's Evacuation function from a control/display module momentary switch in order to activate the panel's programmed alarm responses
FanOff	Turn off an output circuit or relay that connects to the control mechanism used to turn off a fan
FanOn	Turn on an output circuit or relay that connects to the control mechanism used to turn off a fan
FastBlink	Turn a control/display module LED on and off at a fast interval
GAInhibit	Stop a panel's two-stage timer from a control/display module momentary switch in order to prevent the panel from sounding a general alarm
HoldDoor	Turn on an output circuit or relay that connects to the control mechanism used to hold a door
LampTest	Activate a panel's lamp test function from a control/display module momentary switch in order to perform a lamp test on panels that do not contain a 3-LCD
LEDOff	Turn a control/display module LED off
MsgOff	Stop broadcasting an voice message over the selected audio channel
MsgOn	Start broadcasting a voice message over a selected audio channel
NCClose	Turn on an output circuit or relay that connects to the control mechanism used to close a damper
NCFanOff	Turn on an output circuit or relay that connects to the control mechanism used to turn off a fan
NCFanOn	Turnoff an output circuit or relay that connects to the control mechanism used to turn on a fan
NCHoldDoor	Turn off an output circuit or relay that connects to the control mechanism used to hold a door

Table A-2: Fire alarm system commands

Choose this command	To do this
NCOpen	Turn off an output circuit or relay that connects to the control mechanism used to open a damper
NCReleaseDoor	Turn on an output circuit or relay that connects to the control mechanism used to release a door
NSCommonAlarmOff	Turn off a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs
NSCommonAlarmOn	Turn on a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs
NSCommonMonitorOff	Turn off a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs
NSCommonMonitorOn	Turn on a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs
NSCommonSupervisoryOff	Turn off a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs
NSCommonSupervisoryOn	Turn on a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs
NSCommonTroubleOff	Turn off a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs
NSCommonTroubleOn	Turn on a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs
Off	Turn off a system hardware component, circuit, or logic group
OffGuard	Cancel a specific guard patrol tour
On	Turn on a system hardware component, circuit, or logic group
OnGuard	Start a specific guard patrol tour
Open	Turn on an output circuit or relay that connects to the control mechanism used to open a damper
ReleaseDoor	Turn off an output circuit or relay that connects to the control mechanism used to release a door
RemoteAltSensitivityOff	Load the primary sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels
RemoteAltSensitivityOn	Load the alternate sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels
Reset	Activate a panel's Reset function from a control/display module momentary switch in order to reset the system from panels that do not contain a 3-LCD
SlowBlink	Turn a control/display module LED on and off at a slow interval

Table A-2: Fire alarm system commands

Choose this command	To do this
Steady	Turn a control/display module LED on and have it remain on
TroubleSilence	Activate a panel's Trouble Silence function from a control/display module momentary switch in order to turn off a panel's trouble buzzer from panels that do not contain a 3-LCD

Table A-3: Fire alarm system device types

Use this device type	with these events	and these commands
12SW/12LED (12S12L)	none	Disable, Enable
12SW/24LED (12S24L)	none	Disable, Enable
24LED (24L)	none	Disable, Enable
3-AADC (AADC)	none	Disable, Enable
3-ASU (ASU)	none	Disable, Enable
3-DSDC (DSDC)	none	Disable, Enable
3-FTCU (FTCU)	none	Disable, Enable
3-IDC8/4 (IDC)	none	Disable, Enable
3-OPS (OPS)	none	Disable, Enable
3SW/3LEDX6 (3S3L6)	none	Disable, Enable
AlarmSilence (AS)	none	none
AlternateLanguage (ALTL)	none	none
AlternateMsg (ALTM)	none	none
AlternateSensitivity (ALTS)	none	none
Amp	none	AmpOff, AmpOn, Disable, Enable

Table A-3: Fire alarm system device types

Use this device type	with these events	and these commands
And	Acknowledge, Alarm, Monitor, Supervisory, Trouble	Disable, Enable
Audible (AUD)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On
CheckIn (CIG)	none	Disable, Enable, Off, On
CommFailure (CFAIL)	Acknowledge, Trouble	none
CommonAlarmOutput (CAO)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On; but not with CommonAlarmOff, CommonAlarmOn
CommonMonitorOutput (CMO)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On; but not with CommonMonitorOff, CommonMonitorOn
CommonSupervisoryOutput (CSO)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On; but not with CommonSupervisoryOff, CommonSupervisoryOn
DamperControl (DAMP)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On; but not with Close, NCClose, NCOpen, Open
DamperFeedback (DAMPFB)	Acknowledge, Monitor, ServiceDevice, Trouble	Disable, Enable
DoorControl (DOOR)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On; but not with HoldDoor, NCHoldDoor, NCReleaseDoor, ReleaseDoor
DoorFeedback (DOORFB)	Acknowledge, Monitor, ServiceDevice, Trouble	Disable, Enable
Drill	none	none
Emergency (EMER)	Acknowledge, Emergency, ServiceDevice, Trouble	Disable, Enable
Evacuation (EVAC)	none	none
ExtDBIncompatibility (EXTDBIN)	Acknowledge, Trouble	none
Failsafe (FSAFE)	Acknowledge, Alarm, Trouble	none

Table A-3: Fire alarm system device types

Use this device type	with these events	and these commands
FanControl (FAN)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On; but not with FanOff, FanOn, NCFanOff, NCFanOn
FanFeedback (FANFB)	Acknowledge, Monitor, ServiceDevice, Trouble	Disable, Enable
Firephone (FP)	Acknowledge, CallIn, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On
GAInhibit (GAIN)	none	none
Gatevalve (GATE)	Acknowledge, ServiceDevice, SprinklerSupervisory, Supervisory, Trouble	Disable, Enable
GenAlarm (GENA)	Acknowledge, Alarm, MaintenanceAlert, ServiceDevice, Trouble	Disable, Enable
GenSmoke (GENS)	Acknowledge, Alarm, AlarmVerify, MaintenanceAlert, ServiceDevice, Trouble	Disable, Enable
GroundFault (GNDF)	Acknowledge, GroundFault	none
Guard	Acknowledge, ServiceDevice, StationActivation, Trouble	Disable, Enable
GuardPatrol (GPG)	Acknowledge, GuardPatrol	Disable, Enable, OffGuard, OnGuard
Heat	Acknowledge, Alarm, ServiceDevice, Trouble	Disable, Enable
LampTest (LAMP)	none	none
LED	none	Disable, Enable, Off, On; but not with FastBlink, LEDOff, SlowBlink, Steady
LocalAlarm (LALM)	Acknowledge, LocalAlarm	Disable, Enable
LocalMonitor (LMON)	Acknowledge, LocalMonitor	Disable, Enable
LocalRelay (LRLY)	Acknowledge, RelayConfirmation	Disable, Enable

Table A-3: Fire alarm system device types

Use this device type	with these events	and these commands
LocalTrouble (LTRB)	Acknowledge, LocalTrouble	Disable, Enable
LoopControllerResetExt (LCREXT)	Acknowledge, Trouble	none
Matrix	Acknowledge, Alarm, Monitor, Supervisory, Trouble	Disable, Enable
Monitor (MON)	Acknowledge, Monitor, ServiceDevice, Trouble	Disable, Enable
MSG	none	none
NonsupervisedOutput (NSO)	Acknowledge, RelayConfirmation	Disable, Enable, Off, On
NSCommonAlarmOutput (NSCAO)	Acknowledge, Trouble	Disable, Enable, Off, On; but not with NSCommonAlarmOff, NSCommonAlarmOn
NSCommonMonitorOutput (NSCMO)	Acknowledge, Trouble	Disable, Enable, Off, On; but not with NSCommonMonitorOff, NSCommonMonitorOn
NSCommonSupervisoryOutput (NSCSO)	Acknowledge, Trouble	Disable, Enable, Off, On; but not with NSCommonSupervisoryOff, NSCommonSupervisoryOn
NSCommonTroubleOutput (NSCTO)	Acknowledge, Trouble	Disable, Enable, Off, On; but not with NSCommonTroubleOff, NSCommonTroubleOn
PanelCommFault (PCF)	Acknowledge, Trouble	none
Power	Acknowledge, ServiceDevice, SprinklerSupervisory, Supervisory, Trouble	none
POWER (POFF)	none	Disable, Enable
Pull	Acknowledge, Alarm, ServiceDevice, Trouble	Disable, Enable
RebootFault (RBF)	Acknowledge, Trouble	none
Security (SEC)	Acknowledge, Security, ServiceDevice, Trouble	Disable, Enable

Table A-3: Fire alarm system device types

Use this device type	with these events	and these commands
ServiceDeviceSupervision (SERVSUP)	Acknowledge, Trouble	none
ServiceGroup (SG)	Acknowledge, ServiceGroup	Off, On
ServiceGroupActive (SGA)	Acknowledge, ServiceGroupActive	none
Smoke (SMK)	Acknowledge, Alarm, MaintenanceAlert, ServiceDevice, Trouble	Disable, Enable
SmokeVfy (VFY)	Acknowledge, Alarm, AlarmVerify, MaintenanceAlert, ServiceDevice, Trouble	Disable, Enable
SprinklerSupervisory (SPSUP)	Acknowledge, ServiceDevice, SprinklerSupervisory, Supervisory, Trouble	Disable, Enable
StageOne (STAGE1)	Acknowledge, Alarm, ServiceDevice, Trouble	Disable, Enable
StageTwo (STAGE2)	Acknowledge, Alarm, ServiceDevice, Trouble	Disable, Enable
SupervisedOutput (SUP)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On
Supervisory (SUP)	Acknowledge, ServiceDevice, Supervisory, Trouble	Disable, Enable
Switch (SW)	Acknowledge, Switch	Disable, Enable
Tamper (TAMP)	Acknowledge, ServiceDevice, SprinklerSupervisory, Supervisory, Trouble	Disable, Enable
TaskFailure (TFAIL)	Acknowledge, Trouble	none
Temperature (TEMP)	Acknowledge, ServiceDevice, SprinklerSupervisory, Supervisory, Trouble	Disable, Enable
Text	none	Disable, Enable
TimeControl (TIME)	TimeControl	Disable, Enable
TroubleSilence (TS)	none	none

Table A-3: Fire alarm system device types

Use this device type	with these events	and these commands
TwoStageTimerActive (2STAGEA)	Acknowledge, TwoStageTimerActive	none
TwoStageTimerExpiration (2STAGETO)	Acknowledge, TwoStageTimerExpiration	none
UserTrouble (USRTRB)	Acknowledge, Trouble	none
Visible (VIS)	Acknowledge, RelayConfirmation, ServiceDevice, Trouble	Disable, Enable, Off, On
Waterflow (FLOW)	Acknowledge, Alarm, ServiceDevice, Trouble	Disable, Enable
Zone	Acknowledge, Alarm, Trouble	Disable, Enable

Glossary

12SW/12LED device type Classification used for a control/display module with 12 switches and 12 light-emitting diodes. Abbreviation: 12S12L.

12SW/24LED device type Classification used for a control/display module with 12 switches and 24 light-emitting diodes. Abbreviation: 12S24L.

24LED device type Classification used for a control/display module with 24 light-emitting diodes. Abbreviation: 24L.

3-AADC device type Classification used for a 3-AADC addressable analog controller module. Abbreviation: AADC.

3-ASU device type Classification used for a 3-ASU audio source unit. Abbreviation: ASU.

3-DSDC device type Classification used for a Signature controller module. Abbreviation: DSDC.

3-FTCU device type Classification used for a 3-FTCU firefighter's telephone control unit. Abbreviation: FTCU.

3-IDC8/4 device type Classification used for a 3-IDC8/4 initiating device circuit module. Abbreviation: IDC.

3-OPS device type Classification used for a 3-OPS off-premises signaling module Abbreviation: OPS.

3SW/3LEDx6 device type Classification used for a control/display module with 6 groups of 3 switches and 3 light-emitting diodes. Abbreviation: 3S3L6.

Acknowledge event Event produced when an operator acknowledges an event displayed on the 3-LCD module. Abbreviation: ACK.

activation number Total number of state changes required to activate an And or Matrix group. For example, the system counts a device that goes into trouble and then into alarm as two activations.

active state Condition of a circuit when the circuit is turned on.

active project Project currently opened in the SDU program.

Alarm event Event produced when any point on a panel or any panel in the same network routing group changes to the alarm state.

AlarmSilence command Command used to activate the panel's programmed Alarm silence response. Abbreviation: AS.

AlarmSilence device type Classification used for the system pseudo point that changes to the active state when an operator presses a switch that executes the AlarmSilence command. Abbreviation: AS.

AlarmSilence event Event produced when an operator presses a switch that executes the AlarmSilence command. Abbreviation: AS.

AlarmVerify event Event produced when a smoke detector starts its smoke verification cycle. Abbreviation: AVER.

AllCall device type	Classification used for the system pseudo point that changes to the active state when an operator presses the All Call or All Call Minus switch on the 3-ASU.
AllCall event	Event produced when an operator presses the All Call or All Call Minus switch on the 3-ASU.
AlternateLanguage command	Command used to change the language setting on a panel. Abbreviation: ALTL.
AlternateMsgOff command	Command used to toggle the event message routing from the alternate setting to the primary setting. Abbreviation: ALTMOFF.
AlternateMsgOn command	Command used to toggle the event message routing from the primary setting to the alternate setting. Abbreviation: ALTMON.
AlternateSensitivityOff command	Command used to load the primary alarm sensitivity and alarm verification settings into every smoke detector in the system. Abbreviation: ALTSOFF.
AlternateSensitivityOn command	Command used to load the alternate alarm sensitivity and alarm verification settings into every smoke detector in the system. Abbreviation: ALTSON.
Amp device type	Classification used for a zoned amplifier module.
AmpOff command	Command used to turn off an amplifier's audio output and removes the connected audio channel from its input.
AmpOn command	Command used to turn on an amplifier's audio output and connect an audio channel to its input.
And device type	Classification used for an And logic group.
And group	A group of devices in the database combined to provide a unique response based on a number of device activations.
application code	Program file that controls the way a rail module performs fire-related functions.
Audible device type	Classification used for a notification appliance circuit that produces a signal able to be perceived by the sense of hearing. Abbreviation: AUD.
bootloader code	Program file that controls the way a rail module performs system-level functions like startup tests, downloading, and rail bus communications.
bus	A set of conductive paths used for transferring data between electronic components.
cabinet number	1. Index used to classify a cabinet when it is added to the database. 2. Panel number.
CallIn event	Event produced when someone plugs a handset into a firefighter's telephone jack. Abbreviation: CI.
CardDBIncompat device type	Classification used for a pseudo point that the system changes to the active state when the CPU module's database contains different information than a rail module's database. Abbreviation: DBIN.

CheckIn device type	Classification used for a Check-In logic group. Abbreviation: CIG.
Close command	Command used to turn off an output circuit or relay that connects to the control mechanism used to close a damper.
COM port	I/O port on desktop and laptop computers used for connecting RS-232 serial devices. In most cases, COM1 (I/O range 03F8H-03FFH, IRQ4) is used to connect a mouse while COM2 (I/O range 02F8H-02FFH, IRQ3) is used to connect a modem or other serial device.
CommFailure device type	Classification used for a pseudo point that the system changes to the active state when a CPU module can not communicate with other CPU modules on the network. Abbreviation: CFAIL.
CommonAlarmOff command	Command used to turn off a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs. Abbreviation: CAOFF.
CommonAlarmOn command	Command used to turn on a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs. Abbreviation: CAON.
CommonAlarmOutput device type	Classification used for a supervised notification signal output circuit that a panel automatically activates when the FirstAlarm event occurs. Abbreviation: CAO.
CommonMonitorOff command	Command used to turn off a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs. Abbreviation: CMOFF.
CommonMonitorOn command	Command used to turn on a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs. Abbreviation: CMON.
CommonMonitorOutput device type	Classification used for a supervised notification signal output circuit that a panel automatically activates when the FirstMonitor event occurs. Abbreviation: CMO.
CommonSupervisoryOff command	Command used to turn off a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs. Abbreviation: CSOFF.
CommonSupervisoryOn command	Command used to turn on a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs. Abbreviation: CSON.
CommonSupervisoryOutput device type	Classification used for a supervised notification signal output circuit that a panel automatically activates when the FirstSupervisory event occurs. Abbreviation: CSO.
DamperControl device type	Classification used for an output circuit or relay that connects to a control mechanism used for opening or closing a damper. The output circuit may be supervised or nonsupervised depending on the type and placement of the module used for the circuit. Abbreviation: DAMP.
DamperFeedback device type	Classification used for a non-latching input circuit that monitors the operation of a damper control circuit. Abbreviation: DAMPFB.

deenergize	1. To remove an electric current from a device in order for the device to cease operating. 2. To cause the normally-open contacts of a relay to open and the normally-closed contacts to close by removing an electric current from its coil.
Delay command	Command used insert a time delay between rule output command statements when a rule activates and restores. Abbreviation: DLY.
DelayActivate command	Command used to insert a time delay between rule output command statements only when the rule activates. Abbreviation: DLYA.
DelayRestore command	Command used to insert a time delay between rule output command statements only when the rule restores. Abbreviation: DLYR.
device type	A device type is the classification given to an object in the project database that defines the operating characteristics of the device the object represents.
Disable command	Command used to inhibit the automatic or manual control of a system hardware component, circuit, or logic group.
DoorControl device type	Classification used for an output circuit or relay that connects to a control mechanism used for holding or releasing a door. The output circuit may be supervised or nonsupervised depending on the type and placement of the module used for the circuit. Abbreviation: DOOR.
DoorFeedback device type	Classification used for a non-latching input circuit that monitors the operation of a door control circuit. Abbreviation: DOORFB.
download	1. The process of transferring a copy of a file from a local computer to a remote computer over a network or modem. 2. To transfer a copy of a file from a local computer to a remote computer over a network or modem.
Drill command	Command used to activate a panel's programmed Drill response.
Drill device type	Classification used for the system pseudo point that changes to the active state when an operator presses a switch that executes the Drill command.
Drill event	Event produced when an operator presses a switch that executes the Drill command.
Emergency device type	Classification used for check-in devices used in distress notification applications. Abbreviation: EMER.
Emergency event	Event produced when a member of a Check-in group activates their check-in device once anytime outside of their check-in period or twice during their check-in period. Abbreviation: EMER.
Enable command	Command used to allow automatic or manual control of a system hardware component, circuit, or logic group.

energize	1. To apply an electric current to a device in order for the device to operate. 2. To cause the normally-open contacts of a relay to close and the normally-closed contacts to open by applying an electric current to its coil.
Evacuation command	Command used to activate a panel's programmed Evacuation response. Abbreviation: EVAC.
Evacuation device type	Classification used for the system pseudo point that changes to the active state when an operator presses a switch that executes the Evacuation command. Abbreviation: EVAC.
Evacuation event	Event produced when an operator presses a switch that executes the Evacuation command.
event	Outcome produced when an addressable point on the panel changes state.
ExtDBIncompatibility	Classification used for the pseudo point that the system changes to the active state when a CPU module's database is not at the same revision level as other CPU modules on the network.
FailSafe device type	Classification used for the system pseudo point that changes to the active state when a device asserts the rail alarm-not line and the CPU module has not registered an alarm event. Abbreviation: FSAFE.
FanControl device type	Classification used for an output circuit or relay that connects to a control mechanism used for turning a fan on or off. The output circuit may be supervised or nonsupervised depending on the type and placement of the module used for the circuit. Abbreviation: FAN.
FanFeedback device type	Classification used for a non-latching input circuit that monitors the operation of a fan control circuit. Abbreviation: FANFB.
FanOff command	Command used to turn off an output circuit or relay that connects to the control mechanism used to turn off a fan.
FanOn command	Command used to turn on an output circuit or relay that connects to the control mechanism used to turn on a fan.
FastBlink command	Command used to turn a control/display module LED on and off at a fast interval. Abbreviation: FAST.
Firephone device type	Classification used for a telephone riser selector circuit. Abbreviation: FP.
FirstAlarm device type	Classification used for the system pseudo point that changes to the active state when the first point on a panel or any panel in the same network routing group changes to the alarm state.
FirstAlarm event	Event produced when the first point on a panel or any panel in the same network routing group changes to the alarm state. Abbreviation: FA.
FirstDisable device type	Classification used for the system pseudo point that changes to the active state when the first point on a panel or any panel in the same network routing group changes to the disable state.

FirstDisable event	Event produced when the first point on a panel or any panel in the same network routing group changes to the disable state. Abbreviation: FD.
FirstMonitor device type	Classification used for the system pseudo point that changes to the active state when the first point on a panel or any panel in the same network routing group changes to the monitor state.
FirstMonitor event	Event produced when the first point on a panel or any panel in the same network routing group changes to the monitor state. Abbreviation: FM.
FirstSupervisory device type	Classification used for the system pseudo point that changes to the active state when the first point on a panel or any panel in the same network routing group changes to the supervisory state.
FirstSupervisory event	Event produced when the first point on a panel or any panel in the same network routing group changes to the supervisory state. Abbreviation: FS.
FirstTrouble device type	Classification used for the system pseudo point that changes to the active state when the first point on a panel or any panel in the same network routing group changes to the trouble state.
FirstTrouble event	Event produced when the first point on a panel or any panel in the same network routing group changes to the trouble state. Abbreviation: FT.
GAInhibit command	Command used to stop a panel's two-stage timer and prevent the panel from sounding a general alarm. Abbreviation: GAIN.
Gatevalve device type	Classification used for an active-latching input circuit that supervises a gate valve to determine when the valve is not fully open. Abbreviation: GATE.
GenAlarm device type	Classification used for an alarm input circuit that connects to normally-open dry contact initiating devices, non-retarded waterflow alarm switches, or to devices used in applications that must differentiate between shorted and alarm conditions. Abbreviation: GENA.
GenSmoke	Artificial device type that replaces using the Smoke and SmokeVfy device types in a rule. Abbreviation: GENS.
GroundFault device type	Classification used for the pseudo point that a rail module changes to the active state when the rail module detects a ground fault. Abbreviation: GNDF.
GroundFault event	Event produced when a rail module detects a ground fault on its field wiring. Abbreviation: GNDF.
Guard device type	Classification used for an input device that connects to a guard patrol tour station.
GuardPatrol device type	Classification used for a Guard Patrol logic group. Abbreviation: GPG.
GuardPatrol event	Event produced when patrol guard fails to activate a patrol tour station at the proper time. Abbreviation: GPG.

Heat device type	Classification used for input circuit that changes to the alarm state when it detects the heat generated by a fire.
HoldDoor command	Command used to turn on an output circuit or relay that connects to the control mechanism used to hold a door. Abbreviation: HOLD.
label	Descriptive word or phrase used to identify a specific system component in the database.
LampTest command	Command used to activate the lamp test function from a control/display module switch. Abbreviation: LAMP.
LED	1. Semiconductor device that converts electrical energy into light. 2. Acronym for a light-emitting diode.
LED device type	Classification used for a light-emitting diode on a control/display module.
LEDOff command	Command used to turn a control/display module LED off.
local	Close at hand or restricted to a particular area.
local panel	A panel that can be operated directly rather than over the network.
LocalAlarm device type	Classification used for a pseudo point that a loop controller changes to the active state when an unprogrammed device goes into alarm. Abbreviation: LALM.
LocalAlarm event	Event produced when a rail module's LocalAlarm pseudo point goes active. Abbreviation: LALM.
LocalMonitor device type	Classification used for a pseudo point that monitors the activity of rail module functions. Abbreviation: LMON.
LocalMonitor event	Event produced when a rail module's LocalMonitor pseudo point goes active. Abbreviation: LMON.
LocalRelay device type	Classification used for a pseudo point that monitors a channel selection relay on a zoned amplifier module. Abbreviation: LRLAY.
LocalTrouble device type	Classification used for a pseudo point that monitors fault conditions on a rail module that could compromise the operation of the rail module. Abbreviation: LTRB.
LocalTrouble event	Event produced when a rail module's LocalTrouble pseudo point goes active. Abbreviation: LTRB.
logic group	A database object used to provide a single response for a set of devices
LoopControllerResetExt device type	Classification used for the system pseudo point changes to the active state when a loop controller stays in the reset mode longer than normally expected. Abbreviation: LCREXT.
Matrix device type	Classification used for a Matrix logic group.
mechanism	1. A machine or mechanical appliance. 2. The arrangement of connected parts in a machine.
Monitor device type	Classification used for a non-latching input circuit that monitors switch closures. Abbreviation: MON.

Monitor event	Event produced when any point on a panel or any panel in the same network routing group changes to the monitor state. Abbreviation: MON.
MSG device type	Classification used for the recorded audio messages stored in the 3-ASU.
MsgOff command	Command used to stop broadcasting a voice message over the selected audio channel.
MsgOn command	Command used to start broadcasting a voice message over a selected audio channel.
NCClose command	Command used to turn on an output circuit or relay that connects to the control mechanism used to close a damper.
NCFanOff command	Command used to turn on an output circuit or relay that connects to the control mechanism used to turn off a fan.
NCFanOn command	Command used to turn off an output circuit or relay that connects to the control mechanism used to turn on a fan.
NCHoldDoor command	Command used to turn off an output circuit or relay that connects to the control mechanism used to hold a door. Abbreviation: NCHOLD.
NCOpen command	Command used to turn off an output circuit or relay that connects to the control mechanism used to open a damper.
NCReleaseDoor command	Command used to turn on an output circuit or relay that connects to the control mechanism used to release a door. Abbreviation: NCRELEASE.
NonsupervisedOutput device type	Classification used for a notification appliance circuit that does not monitor its output for open or shorted wiring. Abbreviation: NSO.
nonactive	Condition of a circuit when the circuit is turned off.
normal state	Condition of a circuit when there are no alarms or troubles on the system. May be active or nonactive.
notification appliance	Fire alarm system component such as a bell, horn, speaker, or light that provides audible or visible outputs. Audible notification appliances produce a signal that can be heard while visibles produce a signal that can be seen.
NSCommonAlarmOff command	Command used to turn off a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs.
NSCommonAlarmOn command	Command used to turn on a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs.
NSCommonAlarmOutput device type	Classification used for a nonsupervised output relay that a panel automatically activates when the FirstAlarm event occurs. Abbreviation: NSCAO.
NSCommonMonitorOff command	Command used to turn off a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs.

NSCommonMonitorOn command	Command used to turn on a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs.
NSCommonMonitorOutput device type	Classification used for a nonsupervised output relay that a panel automatically activates when the FirstMonitor event occurs. Abbreviation: NSCMO.
NSCommonSupervisoryOff command	Command used to turn off a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs.
NSCommonSupervisoryOn	Command used to turn on a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs.
NSCommonSupervisoryOutput device type	Classification used for a nonsupervised output relay that a panel automatically activates when the FirstSupervisory event occurs. Abbreviation: NSCSO.
NSCommonTroubleOff command	Command used to turn off a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs.
NSCommonTroubleOn command	Command used to turn on a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs.
NSCommonTroubleOutput device type	Classification used for a nonsupervised output relay that a panel automatically activates when the FirstTrouble event occurs. Abbreviation: NSCTO.
n-variable	Programming variable used in a rule to replace the numbers contained in object labels. In order to use the n-variable, the object labels for the devices referenced by the rule must contain a number as part of their label modifier.
object	An object is a database entity that represents actual addressable devices or circuits in the system.
Off command	Command used to turn off a system hardware component, circuit, or logic group.
OffGuard command	Command used to cancel a specific guard patrol tour.
On command	Command used to turn on a system hardware component, circuit, or logic group.
OnGuard command	Command used to start a specific guard patrol tour.
Open command	Command used to turn on an output circuit or relay that connects to the control mechanism used to open a damper.
Power device type	Classification used for an active-latching input circuit that supervises the electrical power supplied to fire pumps or other sprinkler system equipment to determine when power is not present.
pseudo point	An artificial point that reports Alarm, Monitor, or Trouble conditions on a rail module's logic circuits.
Pull device type	Classification used for a pull station.

R1 device type	Classification used for the system pseudo point that changes to the active state when the first phase of the 3-phase reset cycle starts.
R1 event	Event produced when the first phase of the 3-phase reset cycle starts.
R2 device type	Classification used for the system pseudo point that changes to the active state when the second phase of the 3-phase reset cycle starts.
R2 event	Event produced when the second phase of the 3-phase reset cycle starts.
R3 device type	Classification used for the system pseudo point that changes to the active state when the third phase of the 3-phase reset cycle starts.
R3 event	Event produced when the third phase of the 3-phase reset cycle starts.
rail location	Area in a cabinet enclosure used for mounting chassis rail assemblies. Rail 1 refers to the location closest to the top of the enclosure, Rail 2 the next lower, and Rail 3 the next lower after that.
rail-slot position	Physical connection where the module is installed in the cabinet enclosure.
RebootFault device type	Classification used for the pseudo point that the system changes to the active state when a panel restarts unexpectedly. Abbreviation: RBF.
RelayConfirmation event	Event produced when a control relay indicates that its electrical contacts have switched positions. Abbreviation: RLYCFG.
ReleaseDoor command	Command used to turn off an output circuit or relay that connects to the control mechanism used to release a door.
remote panel	A panel that can be operated over the network rather than directly. In a network, all panels that are not the local panel.
RemoteAltSensitivityOff command	Command used to load the primary sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels.
RemoteAltSensitivityOn command	Command used to load the alternate sensitivity and alarm verification settings into every smoke detector on selected panels or on a group of panels.
Reset command	Command used to clear a panel.
Reset device type	Classification used for the system pseudo point that changes to the active state when an operator presses a switch that executes the Reset command.
Reset event	Event produced when an operator presses a switch that executes the Reset command.

rule	Programming statement that designates which commands to execute when a specific event takes place. When the condition exists that makes the input statement true (active), the system will respond as directed by the output statement.
SDU database	Collection of all of the individual projects created by the SDU program
search radius	Specific area within a matrix grid that the system looks for a second detector to go into alarm before generating a group response. The search radius is centered around the first detector to go into alarm and its size is determined by a radius number setting.
Security device type	Classification used for an active-latching input circuit that monitors supervisory or tamper switches. Abbreviation: SEC.
Security event	Event produced when the open input to a device or circuit that monitors a supervisory or tamper switch closes. Abbreviation: SEC.
ServiceDevice event	Event produced when an authorized service technician activates a device in a Service group under test. Abbreviation: SERV.
ServiceDeviceSupervision device type	Classification used for the system pseudo point that changes to the active state when an operator cancels a Service Group test while a circuit under test remained active. Abbreviation: SERVSUP.
ServiceGroup device type	Classification used for a Service logic group. Abbreviation: SG.
ServiceGroup event	Event produced when an authorized service technician activates any device in a Service group under test. Abbreviation: SG.
ServiceGroupActive device type	Classification used for the system pseudo point that changes to the active state when an operator enables a Service Group from the 3-LCD module. Abbreviation: SGA.
ServiceGroupActive event	Event produced when an operator enables a Service Group from the 3-LCD module. Abbreviation: SGA.
SlowBlink command	Command used to turn a control/display module LED on and off at a slow interval.
Smoke device type	Classification used for input circuit that changes to the alarm state when it detects the smoke generated by a fire. Abbreviation: SMK.
SmokeVfy device type	Classification used for an input circuit that changes to the alarm state after it verifies that it detects the smoke generated by a fire. Abbreviation: VFY.
SprinklerSupervisory device type	Classification used for a circuit that supervises a component of the sprinkler system. Abbreviation: SPSUP.
SprinklerSupervisory event	Event produced when the open input to a device or circuit that supervises a component of the sprinkler system closes. Abbreviation: SPSUP.

StageOne device type	Classification used for the first address (pre-alarm stage) of a circuit that connects to a 2-stage pull station. Abbreviation: STAGE1.
StageTwo device type	Classification used for the second address (alarm stage) of a 2-stage pull station. Abbreviation: STAGE2.
Startup device type	Classification used for the system pseudo point that changes to the active state when the panel is energized or an operator initiates a Restart from the 3-LCD module.
Startup event	Event produced when the panel is energized or an operator initiates a Restart from the 3-LCD module. Abbreviation: STUP.
StationActivation event	Event produced when a patrol guard activates a patrol tour station. Abbreviation: STACT.
Steady command	Command used to turn a control/display module LED on and have it remain on.
SupervisedOutput device type	Classification used for a notification appliance signaling circuit that monitors its output for open or shorted wiring. Abbreviation: SUP.
Supervisory device type	Classification used for an input circuit used for supervising switch closures. Abbreviation: SUP.
Supervisory event	Event produced when any point on a panel or any panel in the same network routing group changes to the supervisory state. Abbreviation: SUP.
Switch device type	Classification used for a control/display module switch. Abbreviation: SW.
Switch event	Event produced when an operator presses a control/display module switch.
Tamper device type	Classification used for an input circuit that supervises a secured component of the sprinkler system to determine when someone tries to gain access to it. Abbreviation: TAMP
TaskFailure device type	Classification used for the pseudo point that the system changes to the active state when a CPU module takes too long to complete a task. Abbreviation: TFAIL.
Temperature device type	Classification used for an input circuit that supervises the temperature surrounding a component of the sprinkler system to determine when freezing temperatures exist. Abbreviation: TEMP.
Text device type	Classification used for an Instruction Text logic group.
TimeControl device type	Classification used for a time control. Abbreviation: TIME.
TimeControl event	Event produced when a specific combination of days, dates, and/or time of day occurs. Abbreviation: TIME.
Trouble event	Event produced when any point on a panel or any panel in the same network routing group changes to the trouble state.
TroubleSilence command	Command used to activate the system Trouble Silence response from a control/display module switch.

TroubleSilence device type	Classification used for the pseudo point that the system changes to the active state in order to produce the TroubleSilence event. Abbreviation: TS.
TwoStageTimerActivation event	Event produced when a panel's two-stage alarm timer starts. Abbreviation: 2STAGEA.
TwoStageTimerActive device type	Classification used for the system pseudo point that changes to the active state when a panel's two-stage alarm timer starts. Abbreviation: 2STAGEA.
TwoStageTimerExpiration device type	Classification used for the system pseudo point that changes to the active state when a panel's two-stage alarm timer expires. Abbreviation: 2STAGETO.
TwoStageTimerExpiration event	Event produced when a panel's two-stage alarm timer expires. Abbreviation: 2STAGETO.
upload	1. The process of transferring a copy of a file from a remote computer to a local computer over a network or modem. 2. To transfer a copy of a file from a remote computer to a local computer over a network or modem.
UserTrouble device type	Classification used for the system pseudo point that changes to the active state
Visible device type	Classification used for a notification appliance circuit that produces a signal able to be perceived by the sense of sight. Abbreviation: VIS.
Waterflow device type	Classification used for an input circuit that changes to the alarm state when it detects water flowing through the fire protection sprinkler system. Abbreviation: FLOW.
Zone device type	Classification used for a Zone logic group.

Index

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

Z