



CK722

Network Controller

Commissioning Guide

CK722

Network Controller

Commissioning Guide

December, 2008

24-10239-14 Revision B



Copyright 2008
Johnson Controls, Inc.
All Rights Reserved

No part of this document may be reproduced without the prior permission of Johnson Controls, Inc.

Acknowledgment

Cardkey P2000, BadgeMaster, and Metasys are trademarks of Johnson Controls, Inc.

All other company and product names are trademarks or registered trademarks of their respective owners.

If this document is translated from the original English version by Johnson Controls, Inc., all reasonable endeavors will be used to ensure the accuracy of translation. Johnson Controls, Inc. shall not be liable for any translation errors contained herein or for incidental or consequential damages in connection with the furnishing or use of this translated material.

Due to continuous development of our products, the information in this document is subject to change without notice. Johnson Controls, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with furnishing or use of this material. Contents of this publication may be preliminary and/or may be changed at any time without any obligation to notify anyone of such revision or change, and shall not be regarded as a warranty.

Federal Communications Commissions Notice

This equipment, CK722 has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

The user is cautioned that changes and modifications made to the equipment without approval of the manufacturer could void the user's authority to operate this equipment.

Canadian Notice

This Class B digital apparatus, CK722, complies with Canadian ICES-003.

Cet appareil numerique de la classe B, CK722, est conforme à la norme NMB-003 du Canada.



Declaration of Conformity

This product complies with the requirements of the European Council Electromagnetic Compatibility Directive 2004/108/EEC and the Low Voltage Directive 2006/95/EEC.

This equipment must not be modified for any reason and it must be installed as stated in the Manufacturer's instruction.

If this shipment (or any part thereof) is supplied as second-hand equipment, equipment for sale outside the European Economic Area or as spare parts for either a single unit or system, it is not covered by the Directives.

TABLE OF CONTENTS

1: Introduction

Chapter Summaries	1-2
Manual Conventions	1-2
Key Terms.....	1-3

2: System Components

Introduction	2-1
P2000 Server and Workstations	2-3
System Configuration Tool (SCT)	2-5
CK722	2-6
System Configuration with CK722 Controller	2-8
Network Communication	2-9
Communication with Host	2-9
Communication with Field Devices	2-9
10/100Base-T Networking Guidelines (specific to the CK722)	2-9
S300 I/O Modules	2-11
S300-DIN-RDR2S	2-12
S300-DIN-RDR2SA	2-13
S300-KDM	2-14
Keypad	2-15
Display	2-16
Overview Screens	2-17
Commands	2-19
S300-RDR2	2-29
S300-I16	2-30
S300-IO8	2-31
S300-SI08	2-32
S300-SI8	2-33
Enclosures	2-34
S300-DIN-L	2-35
S300-DIN-S	2-36
S300-XL	2-37
S300-XS	2-38
S300-XXS	2-39
Reader Module Comparison	2-40

3: CK722 Architecture

BACnet and Object Engine	3-5
Attributes Common to All Objects	3-5
Attributes Common to Device Objects	3-5
Object Overview	3-6

Object List	3-7
CK722 Device Object	3-8
S300 Trunk Object	3-9
S300 Hardware Module Object	3-10
S300 Reader Terminal Object	3-11
Security Supervised Input Object	3-12
Security Binary Output Object	3-13
Access Control Object	3-14
Door Sequence Object	3-15
Anti-Passback Object	3-16
Occupancy Object	3-17
Anti-Loitering Object	3-18
Intrusion Area Object	3-19
Intrusion Zone Object	3-20
Intrusion Announcer Object	3-21
Intrusion Keypad/Display Object	3-22
Intrusion Detection System	3-23
Interlock Object	3-24
Multiple Command Object	3-25
Controller Event Object	3-26
KONE Integration	3-27
Otis Integration	3-28
KONE IP Integration	3-29
KONE Controller	3-31
Otis Controller	3-32
KONE IP Controller	3-33
KONE Elevator Object	3-34
Otis Elevator Object	3-35
KONE IP COP Object	3-36
KONE IP DOP Object	3-37
Elevator Object	3-38
Site Object	3-39
Folder Engine Object	3-40
Schedule Object	3-41
Calendar Object	3-42
Broadcast Management Object	3-43
Security Notification Class Object	3-44
RS485 Bus	3-45
Ground Wiring on S300 Bus	3-46

4: CK722 Applications

Entity	4-1
Assets	4-2
Entity Category	4-2
Entity Group	4-3
Entity Sponsor/Owner	4-3
Entity Escort	4-5
Organization	4-7
Validation	4-7
Status	4-7
Journal	4-8
User Accounts	4-8

User-Defined Fields	4-8
Identifier	4-8
Identification Badge	4-9
Access Badge	4-10
Personal Identification Number (PIN)	4-11
Radio Frequency Identification (RFID) Tags	4-11
Access Control.....	4-11
Basic Access Control Components	4-12
Host Computer (P2000 Server).....	4-12
Redundant System.....	4-12
Supervisory Controllers (CK722)	4-13
Field Device	4-13
Reader	4-13
Door Hardware	4-13
Door Contacts	4-13
Electric Locks	4-14
Request to Exit (REX) or Egress Devices.....	4-14
Access Control Features and Applications	4-15
Portal Entry (Single Reader/Keypad)	4-15
Portal Entry and Exit (Card-In/Card-Out)	4-16
Video Imaging	4-17
Fail Safe (Fail Unlocked) and Fail Secure (Fail Locked).....	4-18
Visitor Management	4-18
Alarm Monitoring	4-18
Scheduling	4-18
Anti-loitering	4-19
Anti-Passback	4-19
Occupancy	4-21
Mustering	4-21
Man-traps	4-21
Input/Output (I/O)	4-22
System Events	4-23
Controller Events.....	4-23
Guard Tour.....	4-23
Database Partitioning.....	4-24
Intrusion Detection	4-24
Basic Intrusion Components	4-25
Perimeter.....	4-25
Detectors.....	4-25
Keypad/Display Module	4-27
Controller.....	4-27
Reporting / Annunciation	4-27
Paired Sensors or Double Knock Detection	4-27
Zones	4-28
Areas	4-28
Communicating Systems	4-30
Signaling the Central Station.....	4-30
About the Central Station.....	4-30
Proprietary Monitoring Facility.....	4-30
Intrusion Signals	4-30
Alarm Signal.....	4-30
Secure Signal.....	4-30
Abort or Cancel Signal	4-30

Hardware	4-31
Supervised vs. Non-supervised Systems	4-31
Arming, Disarming and Bypassing the Intrusion Detection System	4-31

5: CK722 Commissioning

Installation.....	5-1
Starting a P2000 SCT Project.....	5-1
Once per Project	5-2
Once per Site	5-3
Once per CK722	5-5
Linking Schedule Objects to P2000 Time Zones	5-5
CK722 Installation and Configuration Verification.....	5-7
Downloading the P2000 SCT Object Database to a CK722 Controller for the First Time	5-8
Verifying Online Status of the CK722.....	5-10
Once per Hardware Module	5-11
Option 1: Creating a Hardware Module Using a Basic Hardware Module Template	5-11
Option 2: Creating a Hardware Module Manually	5-12
Once per Door	5-12
Creating a Package for a Door (x-Templates)	5-12
Creating a Package for a Door (Hardware Module Templates).....	5-13
Cursory Test	5-14
Best Practices.....	5-15
Create and Adhere to a Naming Convention	5-15
Link Schedule Objects to Time Zones As Early As Possible	5-15
Create Job-Specific Templates	5-15
Test Job-Specific Templates Before Creating More Packages	5-16
Correctly Map All Points in an x-Template	5-16
Make Adjustments for Doors Without a Door Contact or REX Device	5-16
How to Make Best Use of a Door Opening Device	5-17
How to Make Best Use of a Door Contact of an Inactive Door	5-17
How to Set Up a Card-In/Card-Out (CICO) Door	5-17
How to Make Best Use of a Bond Sensor	5-18
How to Use a "Catch-All" Card Format	5-18
How to Schedule Practically Anything	5-18
How to "Steal" Unused RDR2S-A Field Points	5-19
Creating a Custom Logic Application.....	5-20
Synchronous vs. Asynchronous Operation	5-20
Synchronous Operation	5-21
Asynchronous Operation.....	5-21
Advantages of Synchronous Operation	5-21
Constraints of Synchronous Operation	5-21
Example	5-21
Controller Event	5-23
Multiple Command	5-23
Interlock	5-23
Attribute Prioritization.....	5-24
Examples of Prioritized Attributes	5-25
Recommended Priorities	5-26
Writing and Releasing Prioritized Attributes	5-27
Releasing Multiple Priorities	5-28
P2000 Host Priority	5-28
Hardware Module Number Guidelines.....	5-29

Network Utility Tool (NUT)	5-29
Receiving SNMP Traps.....	5-30
Using the CK722 Command Line Interface	5-31
Writing the CK722 Database to Flash Memory.....	5-34

6: JCI Standard Templates

About P2000 SCT Templates	6-1
Template Types	6-1
JCI Standard Template Naming Convention	6-2
Identifier Format for Access Control Objects	6-2
Card-In-Card-Out Doors	6-3
Copying and Modifying Existing Templates	6-3
Template Information Framework.....	6-3
JCI Standard Templates	6-4
Full-IO Hardware Module Templates	6-4
JCI_I16_Full-IO	6-5
JCI_IO8_Full-IO	6-7
JCI_SI8_Full-IO	6-9
JCI_SIO8_Full-IO.....	6-12
JCI_RDR2SA_Full-IO	6-14
JCI_RDR8S_Full-IO	6-18
Basic Hardware Module Templates	6-21
JCI_RDR2SA_Basic	6-21
JCI_RDR8S_Basic.....	6-23
Door Hardware Module Templates	6-24
JCI_RDR2S_Card-In.....	6-24
JCI_RDR2S_Card-In-Card-Out.....	6-27
JCI_RDR2SA_Card-In	6-33
JCI_RDR2SA_Card-In-Card-Out	6-33
Door x-Templates	6-34
JCI_x_Contact.....	6-34
JCI_x_Contact-w-Alarm	6-36
JCI_x_Card-In	6-40
JCI_x_Card-In_TAMP	6-42
JCI_x_Card-In_IAD	6-42
JCI_x_Card-In_IAD_TAMP	6-43
JCI_x_CICO	6-47
JCI_x_CICO_TAMP	6-51
JCI_x_CICO_IAD	6-52
JCI_x_CICO_IAD_TAMP	6-52
Miscellaneous Templates	6-58
JCI_x_Elevator.....	6-58
JCI_KDM_with-ACO	6-61
Legacy Templates	6-63

7: Creating Job-Specific Templates

Creating Job-Specific Templates	7-1
Copying Existing Templates as Job-Specific Templates	7-2
Adapting the New Template According to the Job	7-3
Prime Candidates for Job-Specific Attributes.....	7-3
Selecting Package Attributes	7-3

Access Control Application Examples	7-4
Card-In Door: Single Reader or Keypad with REX	7-4
Application Description.....	7-4
Using an Existing Template	7-4
Single Portal Entry with Two Readers	7-5
Application Description.....	7-5
Using an Existing Template	7-5
Object Hierarchy	7-6
Application Notes	7-7
Portal Entry: Card Reader and Keypad Combination	7-7
Application Description.....	7-7
Using an Existing Template	7-7
Application Notes	7-7
Portal Entry and Exit (Card-In-Card-Out)	7-8
Application Description.....	7-8
Using an Existing Template	7-8
Emergency Exit Portal	7-8
Application Description.....	7-8
Using an Existing Template	7-9
Object Hierarchy	7-13
Application Notes	7-13
Reader with Tamper Switch	7-14
Application Description.....	7-14
Using an Existing Template	7-14
Object Hierarchy	7-19
Application Notes	7-19
Portal with Timed Anti-Passback	7-19
Using an Existing Template	7-20
Object Hierarchy	7-21
One Portal with Entry/Exit Anti-Passback	7-21
Using an Existing Template	7-22
Object Hierarchy	7-24
Anti-Loitering Area	7-24
Using an Existing Template	7-24
Object Hierarchy	7-26
Occupancy: Parking Lot with "LOT FULL" Sign	7-26
Application Description.....	7-26
Using an Existing Template	7-27
Object Hierarchy	7-36
Application Notes	7-36
Occupancy: Counting People with Turnstiles	7-36
Application Description.....	7-36
Using an Existing Template	7-37
Object Hierarchy	7-42
Application Notes	7-42
Asset Protection	7-43
Application Description.....	7-43
Using an Existing Template	7-44
Object Hierarchy	7-46
Application Notes	7-46
Assisted Access	7-46
Application Description.....	7-46
ADA Relay.....	7-47
Using an Existing Template	7-48

Object Hierarchy	7-49
Elevator Low Level Interface for Floor-by-Floor Control	7-49
Application Description.....	7-49
Using an Existing Template	7-51
Object Hierarchy	7-55
Application Notes	7-56
Intrusion Detection Examples	7-56
One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application	7-57
Application Description.....	7-57
Using an Existing Template	7-57
Object Hierarchy	7-61
Application Notes	7-62
Supervised Alarm, Tamper and Trouble Inputs Application	7-62
Application Description.....	7-62
Using an Existing Template	7-65
Object Hierarchy	7-69
Double Knock and Paired Sensors	7-70
Application Description.....	7-70
Using an Existing Template	7-70
Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application	7-70
Application Description.....	7-71
Using an Existing Template	7-72
Object Hierarchy	7-76
Template Instantiation (Loading).....	7-76
Adding Intrusion Area 5 to the Site	7-78
Lighted Display Signaling Area Arm/Disarm	7-81
Application Description.....	7-81
Using an Existing Template	7-81
One Area, Seven Zones, No Keypad, with Keylock Arming/Disarming	7-82
Application Description.....	7-82
Using an Existing Template	7-82
Object Hierarchy	7-87

8: Scheduling

Schedule	8-1
Using a P2000 Time Zone	8-2
Calendar	8-2
Weekly Schedule	8-3
Exception Schedule	8-3
Time/Value Pairs (Events)	8-5
Scheduled Items	8-5
Dates – Calendar Entry and Exception Schedule.....	8-6
Date/Date Range	8-6
Week and Day	8-7
Wild Cards	8-7
Wild Cards – Date	8-8
Wild Cards – Date Range	8-9
Wild Cards – Week and Day	8-10
Effective Period	8-11
Fast Clock	8-11
Data Consistency Checking.....	8-11
Managing Schedules and Calendars	8-12

Creating a Schedule or Calendar	8-12
Displaying an Existing Schedule or Calendar	8-13
Displaying Scheduled Event (Time/Value Pairs)	8-13
Adding Scheduled Events	8-14
Editing Scheduled Events	8-15
Deleting Scheduled Events	8-15
Copying and Pasting Events from One Day of the Week to Another	8-15
Adding Exception Schedules	8-16
Editing Exception Schedules	8-17
Removing Exception Schedules	8-18
Adding Scheduled Items	8-18
Editing Scheduled Items	8-18
Removing Scheduled Items	8-19
Editing the Effective Period of a Schedule	8-19
Creating a New Calendar Entry	8-20
Editing a Calendar Entry	8-21
Deleting a Calendar Entry	8-21
Toggling Between Calendar Views	8-22
Editing the Attributes of a Schedule or Calendar	8-22
Copying and Pasting a Schedule or Calendar (Offline Mode Only)	8-22
Deleting a Schedule or Calendar	8-23

Appendix A: Secured Premises Notification Settings

Sequence of Events	A-1
P2000 System Configuration Tool (SCT) Procedures	A-2
Access Control Object Configuration	A-2
Intrusion Zone Object Configuration	A-3
Controller Event Object Configuration	A-3
P2000 Host Procedures	A-3

Appendix B: Using Keypad Readers

Invoking Access Requests from a Keypad	B-1
Invoking Common PIN Access Requests from a Keypad	B-2
Invoking Timed Overrides from a Keypad	B-2
Invoking Controller Events from a Keypad	B-5
Quick Guide to Using Keypad Readers	B-8

Appendix C: Configuring Offline Mode Options

No Access in Offline Mode	C-1
Card Access in Offline Mode	C-1
Card and PIN Access in Offline Mode	C-3

Appendix D: Identifier Formats

Index

LIST OF FIGURES

System Overview	2-2
P2000 Host with SCT	2-3
System Components: P2000 Host Software Details.....	2-4
System Components: SCT Details	2-5
System Components: CK722 Controller Details	2-6
CK722 Controller Hardware	2-7
Example of System Configuration.....	2-8
4x5 Rule	2-10
S300 Buses.....	2-11
S300-DIN-RDR2S	2-12
S300-KDM	2-14
S300-RDR2 Field	2-29
S300-I16	2-30
S300-IO8	2-31
S300-SI08	2-32
S300-SI8	2-33
Example of the System Configuration.....	2-34
S300-DIN-L With Installed Components	2-35
S300-DIN-S With Installed Components	2-36
S300-XL	2-37
S300-XS	2-38
S300-XXS	2-39
Example: Three Interlinked Objects	3-1
Example: Adding a Second Reader Terminal Object	3-2
Example: Adding a Schedule Object	3-2
Example: Adding Occupancy, Anti-Passback, and Anti-Loitering Objects	3-3
Example: Basic Motion Detection Application.....	3-4
CK722 Device Object	3-8
S300 Trunk Object	3-9
S300 Hardware Module Object.....	3-10
S300 Reader Terminal Object	3-11
Security Supervised Input Object	3-12
Security Binary Output Object.....	3-13
Access Control Object	3-14
Door Sequence Object.....	3-15
Anti-Passback Object.....	3-16
Occupancy Object.....	3-17
Anti-Loitering Object	3-18
Intrusion Area Object Details	3-19
Intrusion Zone Object Details.....	3-20
Intrusion Annunciator Object Details.....	3-21

Keypad/Display Object Details.....	3-22
Intrusion Detection System: Object Interactions	3-23
Interlock Object	3-24
Multiple Command Object.....	3-25
Controller Event Object.....	3-26
KONE Integration Object	3-27
Otis Integration Object	3-28
KONE IP Integration Object	3-30
KONE Controller Object.....	3-31
Otis Controller Object.....	3-32
KONE IP Controller Object	3-33
KONE Elevator Object	3-34
Otis Elevator	3-35
KONE IP COP Object	3-36
KONE IP DOP Object	3-37
Elevator Object	3-38
Site Object	3-39
Folder Object	3-40
Schedule Object.....	3-41
Calendar Object.....	3-42
Broadcast Management Object	3-43
Security Notification Class Object.....	3-44
RS485 Buses	3-45
Entity Types	4-1
Entity Category Diagram.....	4-2
Entity Group	4-3
Entity Sponsor.....	4-4
Entity Group Sponsor.....	4-4
Entity Group Escort.....	4-6
Identifiers	4-9
Sample Badge Identifier.....	4-10
Portal with Entry Reader	4-15
Portal with Entry and Exit Readers	4-16
Typical P2000 Video Imaging Configuration	4-17
Anti-loitering Feature.....	4-19
Anti-Passback Time Rule.....	4-20
Anti-Passback Entry/Exit Rule	4-20
Man-traps.....	4-22
Paired Sensors	4-27
Double Knock.....	4-28
Zones and Areas.....	4-29
Synchronous vs. Asynchronous Operation.....	5-22
Prioritized Attribute Controlled by Several Different Applications	5-25
PC to CK722 Controller Connections	5-33
Object Diagram for JCI_I16_Full-IO Template.....	6-5
Graphical Representation of JCI_I16_Full-IO Template	6-6
Object Diagram for JCI_IO8_Full-IO Template	6-7
Graphical Representation of JCI_IO8_Full-IO Template	6-8
Object Diagram for the JCI_SI8_Full-IO Template	6-10
Graphical Representation of the JCI_SI8_Full-IO Template	6-10

Object Diagram for the JCI_SIO8_Full-IO Template	6-12
Graphical Representation of the JCI_SIO8_Full-IO Template	6-13
Object Diagram for the JCI_RDR2SA_Full-IO Template	6-15
Graphical Representation of the JCI_RDR2SA_Full-IO Template.....	6-16
Object Diagram for the JCI_RDR2SA_Basic Template	6-21
Graphical Representation of the JCI_RDR2SA_Basic Template	6-22
Object Diagram for the JCI_RDR8S_Basic Template	6-23
Object Diagram for the JCI_RDR2S_Card-In Template	6-24
Graphical Representation of the JCI_RDR2S_Card-In Template.....	6-25
RDR2S Cross-Wiring for Two Readers Sharing a Single Door Strike	6-28
Object Diagram for the JCI_RDR2S_Card-In-Card-Out Template	6-29
Graphical Representation of the JCI_RDR2S_Card-In-Card-Out Template.....	6-29
Object Diagram for the JCI_x_Contact Template	6-34
Graphical Representation of the JCI_x_Contact Template.....	6-35
Object Diagram for the JCI_x_Contact-w-Alarm Template	6-37
Graphical Representation of the JCI_x_Contact-w-Alarm Template	6-37
Object Diagram for the JCI_x_Card-In Template.....	6-40
Graphical Representation of the JCI_x_Card-In Template	6-40
Object Diagram for the JCI_x_Card-In_IAD_TAMP Template.....	6-43
Graphical Representation of the JCI_x_Card-In_IAD_TAMP Template	6-44
Object Diagram for the JCI_x_CICO Template.....	6-48
Graphical Representation of the JCI_x_CICO Template	6-48
Object Diagram for the JCI_x_CICO_IAD_TAMP Template.....	6-53
Graphical Representation of the JCI_x_CICO_IAD_TAMP Template	6-53
Object Diagram for the JCI_x_Elevator Template	6-58
Graphical Representation of the JCI_x_Elevator Template	6-59
Object Diagram for the JCI_KDM_with-ACO Template	6-61
Graphical Representation of the JCI_KDM_with-ACO Template.....	6-62
Object Hierarchy for Single Portal Entry with Two Readers Application.....	7-6
Object Hierarchy for Emergency Exit Portal Application.....	7-13
Object Hierarchy for Reader with Tamper Switch Application	7-19
Object Hierarchy for Portal with Timed Anti-Passback Application.....	7-21
Object Hierarchy for One Portal with Entry/Exit Anti-Passback Application.....	7-24
Object Hierarchy for Anti-Loitering Area Application.....	7-26
Object Hierarchy for Occupancy: Parking Lot with "LOT FULL" Sign Application.....	7-36
Object Hierarchy for Occupancy: Counting People with Turnstiles Application.....	7-42
Object Hierarchy for Asset Protection Application	7-46
Alternate Access Timing Diagram.....	7-47
Object Hierarchy for the Assisted Access Application	7-49
Object Hierarchy for Elevator Low Level Interface for Floor-by-Floor Control Application.....	7-55
Object Hierarchy for One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application.....	7-61
Supervised Alarm, Tamper and Trouble Inputs Application.....	7-64
Object Hierarchy for Supervised Alarm, Tamper and Trouble Inputs Application.....	7-69
Controlling Multiple Intrusion Areas with a Single Area Across Multiple CK722 Controllers.....	7-71
Object Hierarchy for Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application.....	7-76
Object Hierarchy for One Area, Seven Zones, No Keypad, with Keylock Arming/Disarming Application.....	7-87

INTRODUCTION

The CK722 is a powerful and flexible network controller for the P2000AE Security Management System (SMS) that is configured using the P2000AE System Configuration Tool (SCT). Configuring the CK722 consists of inserting and interconnecting BACnet objects to create a unified access control or intrusion detection application. Applications can range from basic access control functions, such as configuring an entry reader for a particular door, to more advanced functions, such as assigning anti-passback, occupancy and anti-loitering rules to a controlled area of the facility.

This document describes the hardware and software components of the P2000AE SMS, including brief descriptions of the host, SCT, CK722, hardware modules, and objects. It also covers many access control and intrusion detection applications associated with each object, and explains how to interconnect objects to build several common applications.

Many standard applications are provided as templates, but different applications can also be developed or modified using the template/package feature.

This document **does not** describe how to:

- Mount, wire, or power on a CK722.
Refer to the *CK722 Network Controller Hardware Installation Manual* for information.
- Configure legacy controllers, such as the CK721-A, CK721, CK720/705, etc.
Refer to the *P2000AE Software User's Manual*.

Information in this document on how to use the P2000AE SCT is limited. For detailed use instructions, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

NOTES

- *The screen captures shown in this manual may differ slightly, depending on the installation media and the software version you are using.*
 - *"P2000AE" is also referred to as "P2000" throughout this manual.*
 - *Although the RDR8S hardware module is mentioned in this document, it is currently not available for purchase.*
-

CHAPTER SUMMARIES

This guide is divided into the following chapters:

- **Chapter 1: Introduction** provides a brief description of this guide and each chapter, and defines the key terms and conventions used throughout the guide.
- **Chapter 2: System Components** presents a system overview followed by detailed descriptions of each system component.
- **Chapter 3: CK722 Architecture** includes a brief description of the objects used by the CK722 as well as the RS485 bus.
- **Chapter 4: CK722 Applications** provides detailed information about various CK722 applications, such as entities and identifiers, access control, and intrusion detection.
- **Chapter 5: CK722 Commissioning** outlines the process of commissioning the CK722 network controller for use in the P2000 Security Management System.
- **Chapter 6: JCI Standard Templates** explains each of the predefined templates provided with the P2000 SCT and how to use them to build customized applications.
- **Chapter 7: Creating Job-Specific Templates** describes, in detail, how to define various security logic functions using the P2000 SCT. Included are examples for defining access control and intrusion detection functions.
- **Chapter 8: Scheduling** describes how to use the scheduling feature to automate certain functions, such as unlocking a door for a defined amount of time during defined days of the week.
- **Appendix A: Secured Premises Notification Settings** explains how to configure the P2000 SMS to enable entities to suppress and unsuppress alarms using a Controller Event object in accordance with UL 1076.
- **Appendix B: Using Keypad Readers** describes how to invoke access requests, Common PIN requests, Card ID requests, Timed Overrides, and Controller Events using a keypad reader.
- **Appendix C: Configuring Offline Mode Options** describes the Offline Mode feature options and how to configure them. Offline Mode occurs when a hardware module becomes disconnected from its supervisory controller.
- **Appendix D: Identifier Formats** lists and describes all of the official card formats included as part of the P2000 installation.

MANUAL CONVENTIONS

The following items are used throughout this installation manual to indicate special circumstances, exceptions, important points regarding the equipment or personal safety, or to emphasize a particular point.

NOTE

Notes indicate important points or exceptions to the information provided in the main text.



Cautions remind you that certain actions, if not performed exactly as stated, can cause damage to equipment.

KEY TERMS

The following terms are used throughout this manual:

Archive – The archive database holds the configuration information for the CK722 supervisory devices, field devices, and field points that make up one or more sites.

BACnet (Building Automation and Control network) – A standard protocol from the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). This protocol provides a standard for allowing computers and equipment controllers to transfer data between the devices in an object-oriented fashion. The BACnet standard defines the types of information and attributes that any device must maintain, and defines how BACnet messages are communicated between the various devices.

CK722 – An advanced, intelligent, network controller capable of handling high volume, high-speed traffic with host security management systems, such as the P2000 SMS. The CK722 uses the Metasys Control Engine (MCE), an object-based operating system that resides on the controller, to perform various functions based on the objects defined in the P2000 SCT. The CK722 interacts with the P2000 host software and the field hardware (e.g. readers, doors, and I/O devices) to provide SMS controls. Each CK722 supports up to 64 readers/doors.

Entity – A person, asset or system account entered in the P2000 SMS. See “Entity” on page 4-1 for more information.

Field devices – These devices consists of reader interfaces, keypad/display modules, input points, or output relays.

Field points – In the P2000 SCT, a field point represents a peripheral device, such as a keypad, reader, sensor, or relay, which is added to field devices or supervisory devices.

Identifier – Enables the P2000 SMS to identify (authenticate) and track an entity in the system. Also known as “credentials.” See “Identifier” on page 4-8.

Legacy Panels – Controllers developed and available prior to the CK722 controller. Legacy controllers include the CK721-A, CK721, CK720, and CK705.

MCE (Metasys Control Engine) – An object-based operating system that resides on a controller and performs control functions with BACnet objects. MCE interfaces with external control hardware to perform various control functions.

Object – Objects are self-contained functional items in the P2000 SMS that contain processes to manage security components. Each object that exists in a system is based on a specific object type. There is an object type to manage the functions of the site, object types to manage the operation of the various device types installed on the site, object types to manage the physical input and output points of each field device, and others.

Objects communicate with the rest of the system and with the user through their attributes. The object type defines the basic function of the object, but the actual behavior of each object also depends on the values assigned to its configuration attributes and received by its input attributes from other objects. The object writes its status conditions and the results of internal processes to its output attributes.

P2000 SCT – P2000 System Configuration Tool used to configure CK722 controllers for use with the P2000 SMS. The P2000 SCT is a browser-based application provided as part of the P2000 SMS software package. Users can access the P2000 SCT directly from the P2000 SMS software or via their browser on any computer with a LAN/WAN connection to the P2000 server. When the P2000 SCT's configuration settings are modified, these changes can be downloaded to the CK722 controller, and the P2000 SMS host software is updated simultaneously to reflect the changes.

P2000 SMS – P2000 Security Management System, which consists of the host, SCT, controllers, field devices (e.g. S300 modules), terminals, etc. The P2000 host system software is a Windows®-based application that resides on P2000 servers and workstations and oversees the operation of the complete security system.

Package (or Template Instance) – A copy of objects and connections from an existing template that is loaded into a CK722 supervisory device. A package has the same objects, attributes and connections as its parent template; however, object names are assigned upon package instantiation according to the tags of the objects being loaded (instantiated).

Package Tag – Identified as text surrounded by curly brackets, a tag is appended to the name of an object in a template, allowing you to assign a more representative name to the object when loading the template as a package.

Portal – An access point in the P2000 SMS, such as a door or gate.

Site – A site is a logical grouping of CK722 devices and their field level devices that are on the same Local Area Network (LAN).

Template – “Rubber stamps” of pre-defined applications that can be used to rapidly populate the P2000 SCT hardware configuration database. Templates are used to create *Packages*, which contain all of the components for a single application, such as a door.

SYSTEM COMPONENTS

This chapter presents a system overview followed by detailed descriptions of each system component.

INTRODUCTION

The Security Management System consists of several pieces of hardware that can be divided into four different levels. Each level is responsible for a different part of your access control application. Some applications run entirely within a single level, whereas others stretch out across several levels.

- I. The top level (Host Level) comprises a network of PCs with Johnson Controls' software enabling host level applications, as well as providing a user interface to configure and operate applications in the levels underneath.
- II. The second level (Supervisory Level) comprises Johnson Controls' CK722 supervisory controllers. This level accommodates the majority of the real-time applications.
- III. The third level (Field Device Level) comprises Johnson Controls' line of field devices, offering electrical inputs and outputs to interface to the fourth level. Only very basic access control applications are handled in this level.
- IV. The lowest level (Peripheral Devices) comprises third party equipment such as readers, keypads, relays, acoustical and visual indicators, and sensors.

Refer to Figure 2-1 for a graphic representation of the system levels.

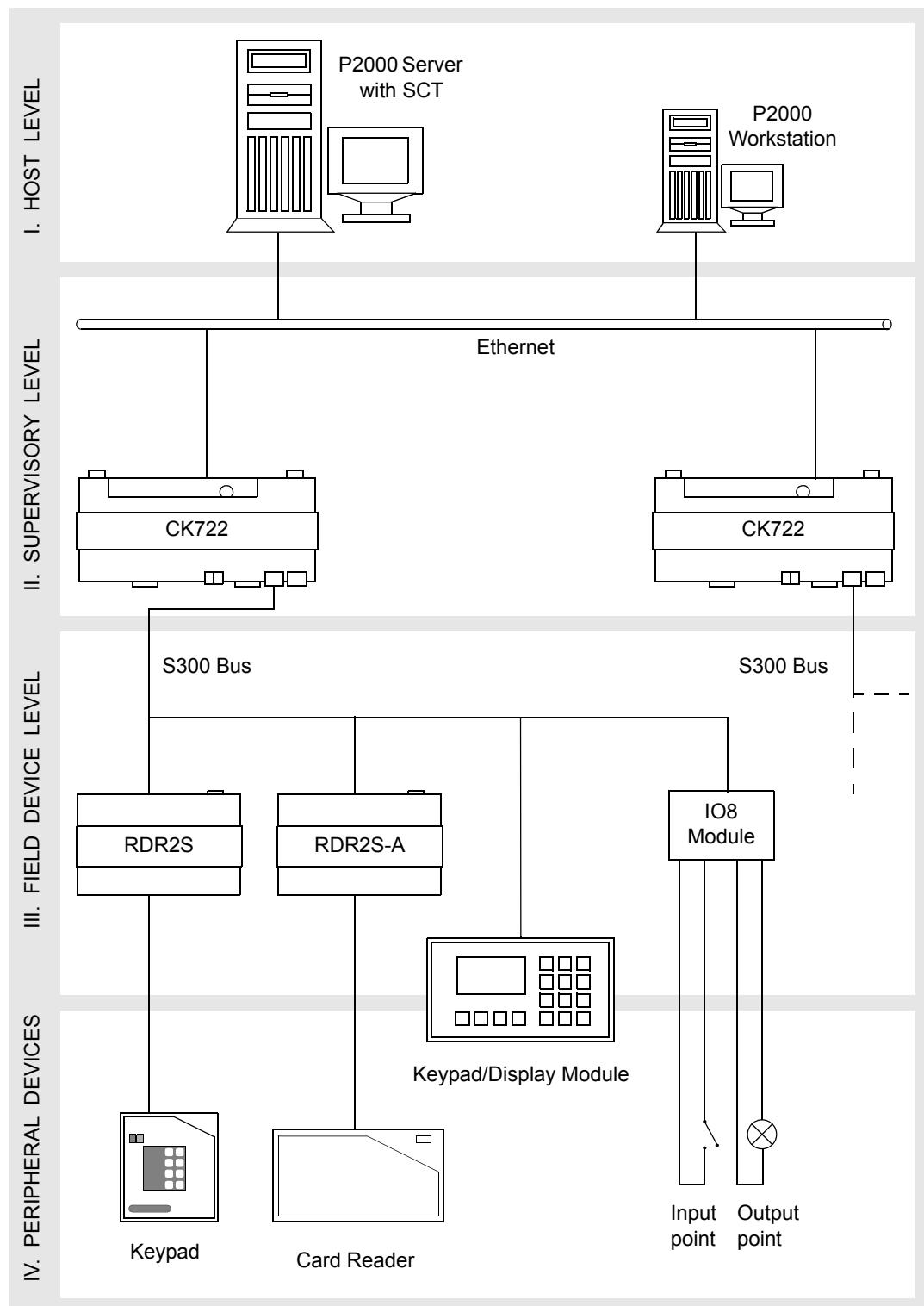


Figure 2-1: System Overview

P2000 SERVER AND WORKSTATIONS

P2000 server (also called “host” in this manual) provides the overall access control capabilities required to operate and monitor the operation of an access control system.

The P2000 server has the following functions:

- Runs the P2000 applications
- Provides management of entities and identifiers
- Stores database information
- Communicates with the P2000 workstations
- Communicates with the controllers
- Allows for configuration of devices (legacy panels, network panels prior to CK722, etc.)

The System Configuration Tool (SCT) is installed as a part of the P2000 server installation. For details on SCT refer to page 2-5.

P2000 workstations run the P2000 workstation software and allow additional users to monitor and configure the P2000 system. Workstations communicate with the server via an Ethernet TCP/IP local area network (LAN).

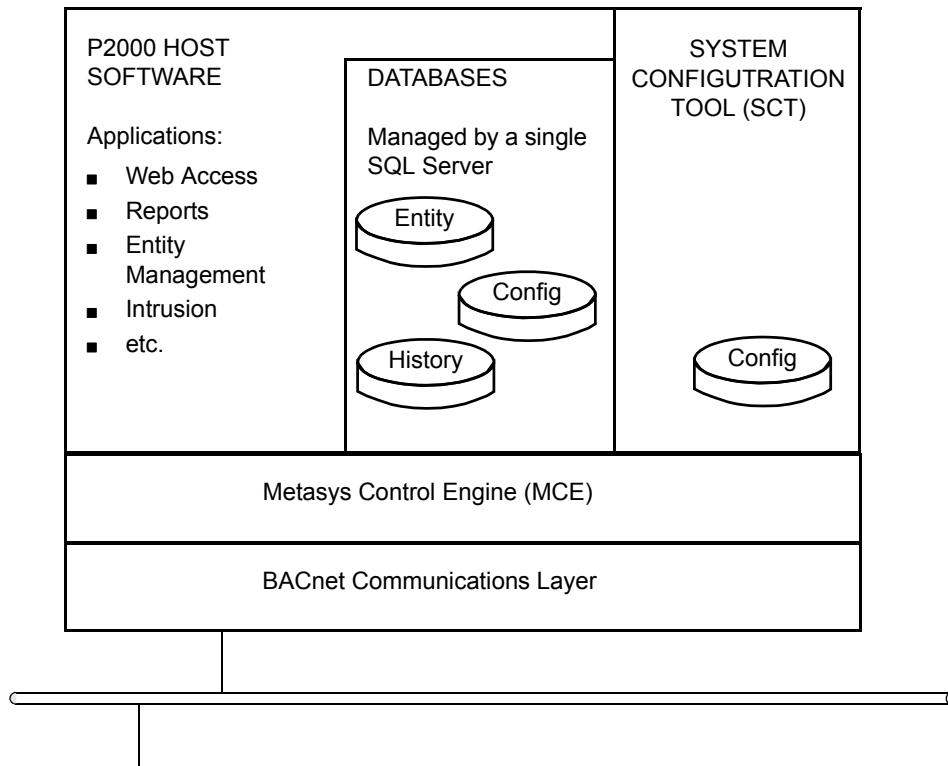


Figure 2-2: P2000 Host with SCT

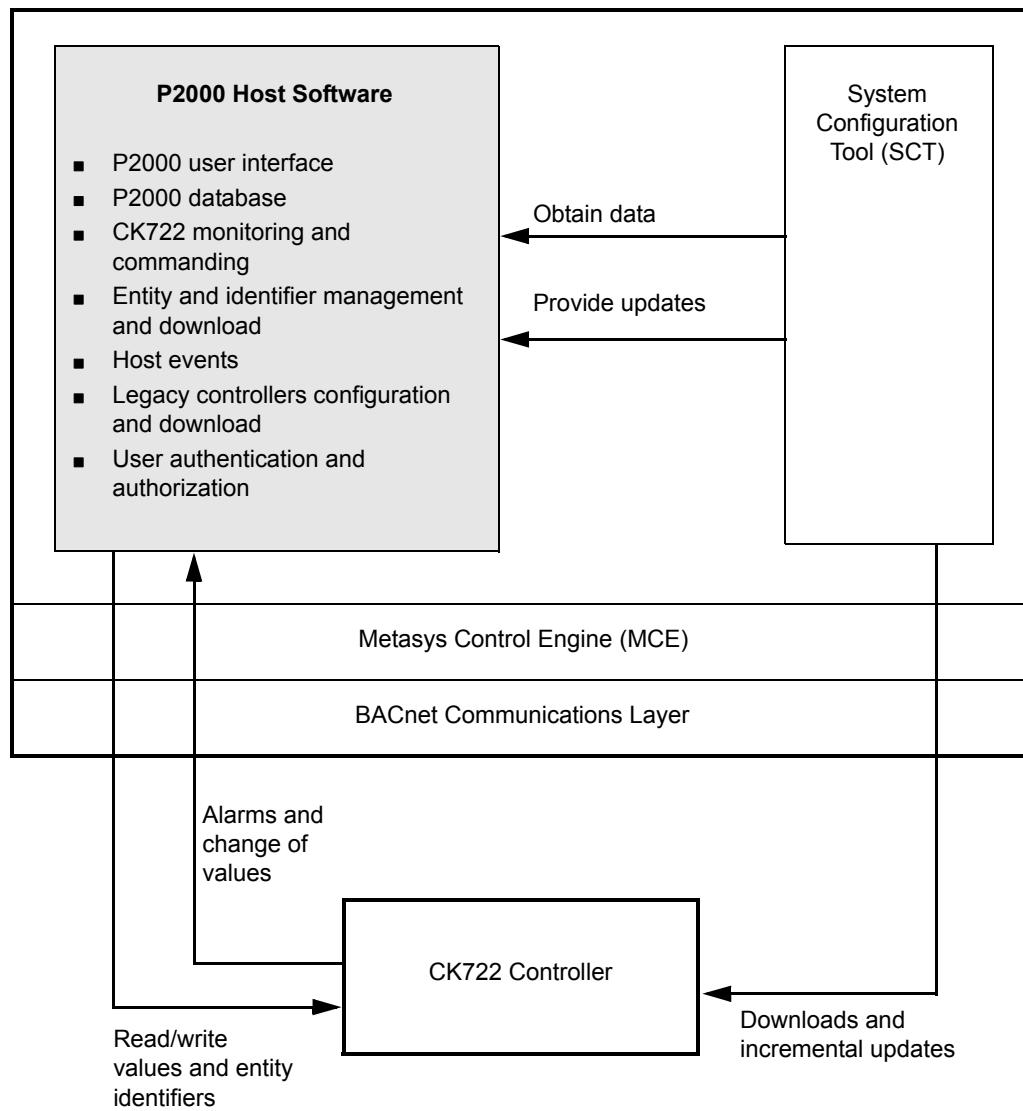


Figure 2-3: System Components: P2000 Host Software Details

SYSTEM CONFIGURATION TOOL (SCT)

The P2000 SCT is a browser-based application provided as part of the P2000 SMS software package and installed on the P2000 server. Users can access the P2000 SCT directly from the P2000 SMS software or via their browser on any computer with a LAN/WAN connection to the P2000 server. When the P2000 SCT's configuration settings are modified, these changes are downloaded to or synchronized with the CK722 controller, and the P2000 SMS host software is updated to reflect the changes.

The SCT allows you to:

- Configure CK722 controllers
- Create and modify BACnet objects that will be downloaded to one or more CK722 controllers
- Create archive (configuration) databases
- Synchronize the CK722 controller with the archive database
- Updates the P2000 host software with the changes made to the SCT

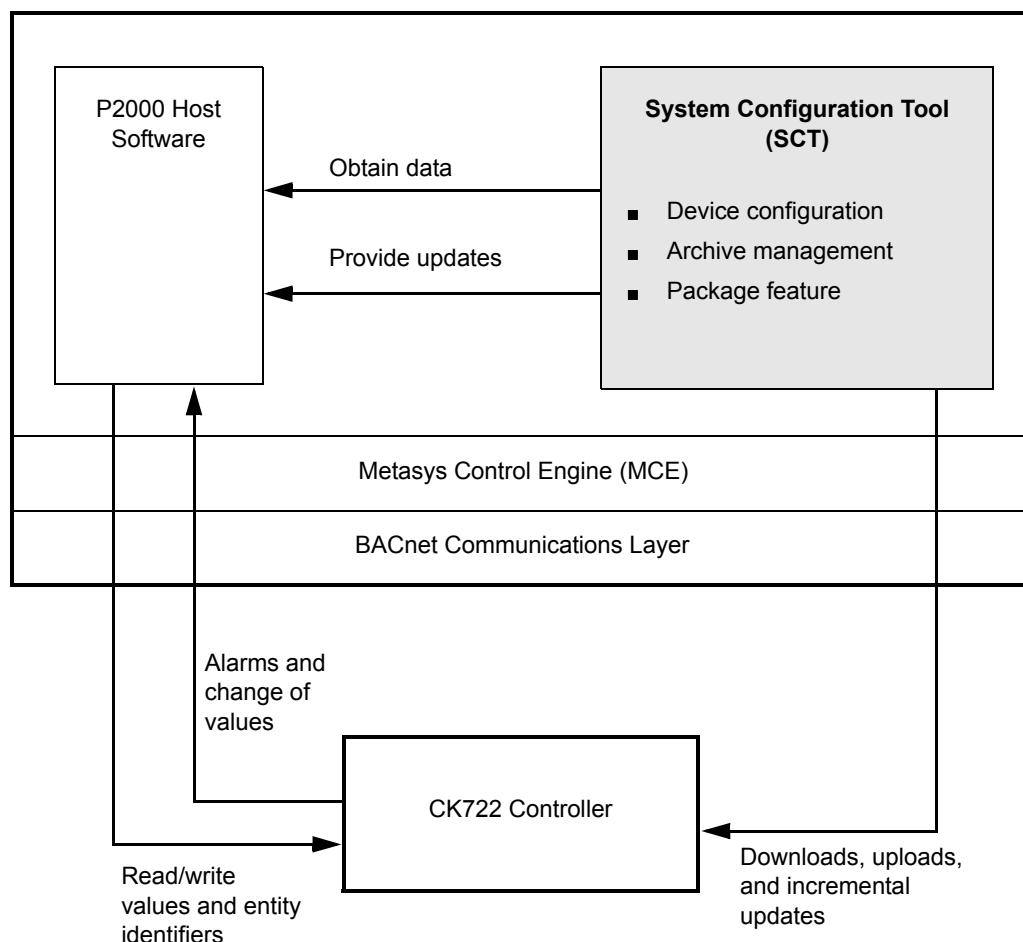


Figure 2-4: System Components: SCT Details

CK722

The CK722 is an advanced, intelligent network controller which utilizes the Metasys® Control Engine (MCE) to provide highly configurable control applications.

The CK722 controller:

- Performs control applications based on the objects defined in MCE.
- Handles high volume, high-speed traffic and communicates with the system host (P2000 server).
- Uses the S300 field devices to provide controls.

The CK722 is intended to be mounted on a wall, DIN rail, or in an S300-DIN enclosure.

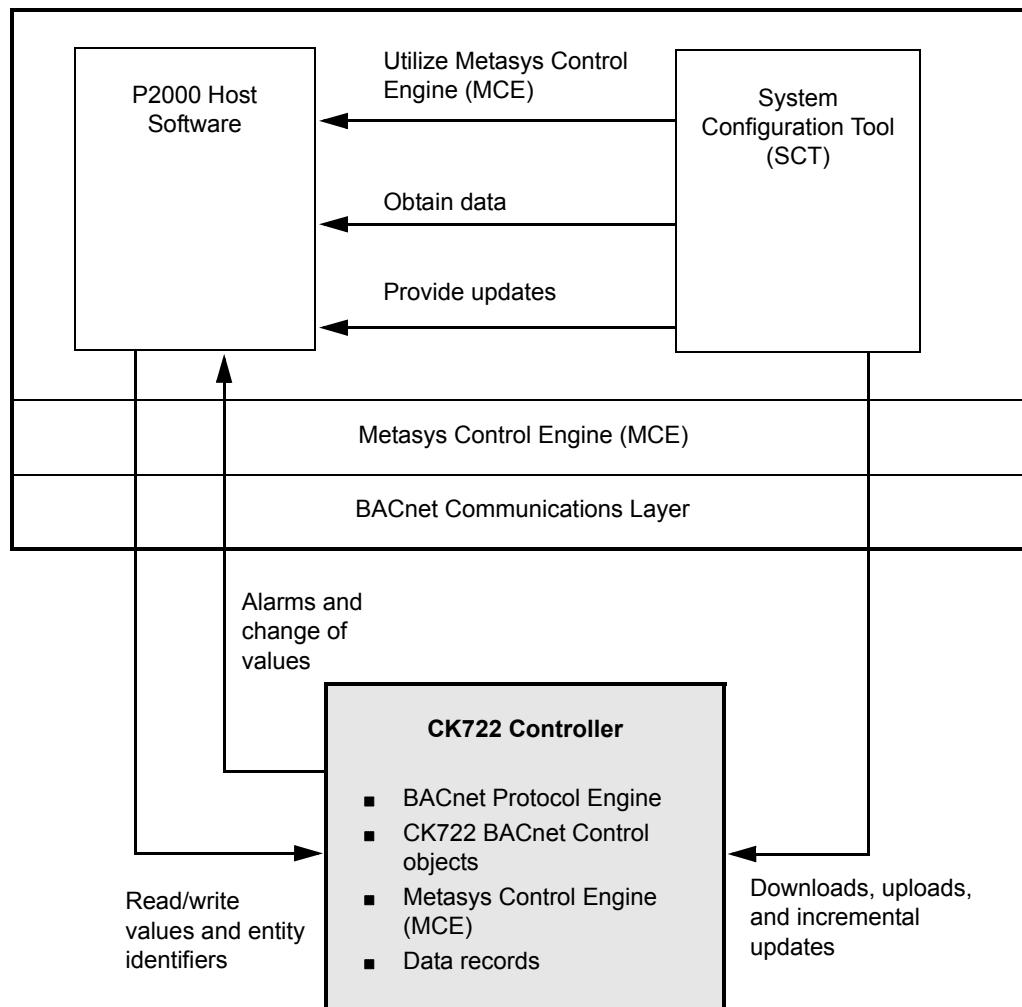


Figure 2-5: System Components: CK722 Controller Details

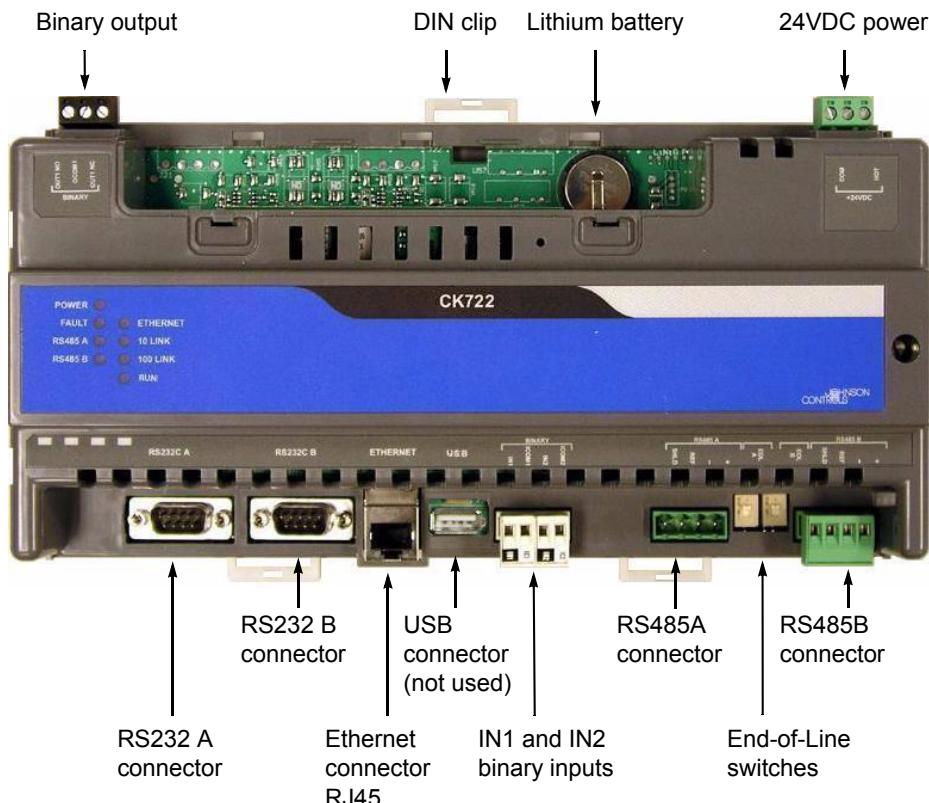


Figure 2-6: CK722 Controller Hardware

The major functional components of the CK722 are:

- Embedded 32-bit processor
- 128 MB onboard flash memory (for the operating system and database)
- 3V lithium battery
- IN1 and IN2 - Binary inputs, unsupervised
- Binary output - Form C Relay, SPDT, 30VDC/VAC maximum
- LED indicators (POWER, FAULT, RS485 A, RS485 B, ETHERNET, 10/LINK, 100/LINK, and RUN)
- Connectors:
 - RS232C A - RS-232 Serial Interface, DB9 port for setting the static IP address
 - RS232C B - RS-232 Serial Interface, DB9 port for high level elevator integration
 - RS485A - For field device communication
 - RS485B - For field device communication
 - RJ45 - 10/100Base-T network port for host communication
 - USB - Not used

System Configuration with CK722 Controller

Figure 2-1 illustrates a simple system configuration, including CK722 controllers, DIN enclosures, and expansion enclosures. Those components are described later in this chapter.

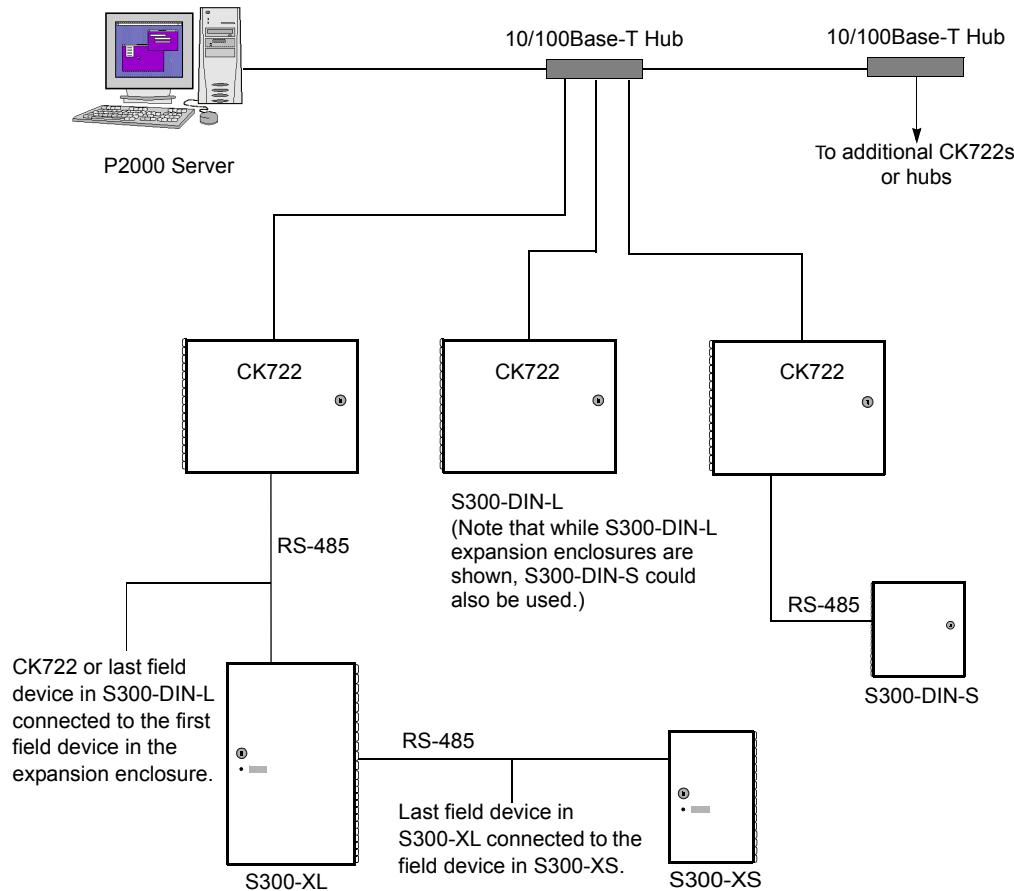


Figure 2-7: Example of System Configuration

Network Communication

Communication with Host

The CK722 panels communicate with the P2000 SMS via 10/100Base-T Ethernet network, using a standard 10/100Base-T cabling and 10/100Base-T hubs. The communication protocol used is BACnet. Cabling of the system must comply with the industry-standard network guidelines.

10/100Base-T Ethernet (also referred to simply as 10/100Base-T) is the physical network connecting the P2000 SMS to the CK722 panels. 10/100Base-T provides reliable connections using a series of hubs to lengthen a network's distance at a local level. Bridges, routers, and network switches increase a network's size to greater distances across states or over continents.

The basic unit of 10/100Base-T networks (and others as well) is the LAN (Local Area Network). Johnson Controls recommends the P2000 SMS be on its own LAN, meaning a single self-contained network not connected to any other network. This will allow you to maintain security and implement a simple IP addressing scheme.

Communication with Field Devices

You can add field devices to connect readers, monitor 2 or 4-state input points, and adding output relays to perform manual or automatic control functions. In addition, input points can be linked to output relays. Communication between the CK722 and the field devices is accomplished via the S300 protocol over RS-485.

10/100Base-T Networking Guidelines (specific to the CK722)

As a network device, the CK722 can be installed in a variety of configurations based on the needs of your sites. However, the installation must follow the standard 10/100Base-T four by five (annotated 4x5) rule. The rule states that:

- The 10/100Base-T network may contain a maximum of **four** hubs and **five** segments. Another explanation: a maximum of four hubs can be installed between the P2000 SMS and the last CK722 panel in the network.
- The maximum **segment length** is 328 ft (100 m). This is the distance between two hubs, or the distance between a hub and a network device such as the CK722.
- Wiring from a CK722 to a hub is straight through (CAT-5, 8 conductor cable, RJ45 connectors).

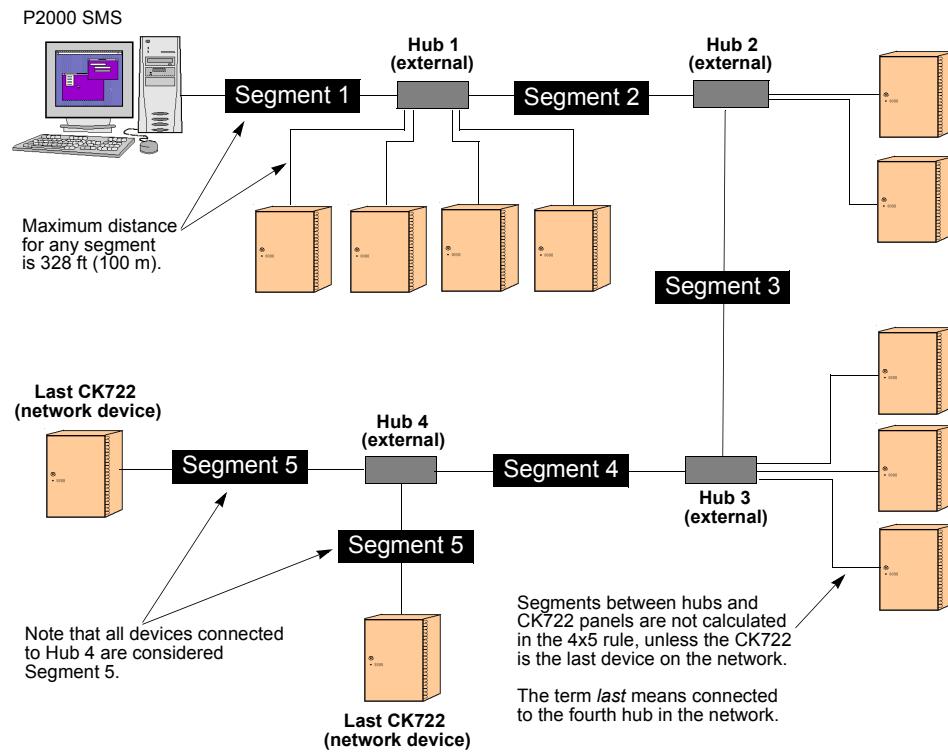


Figure 2-8: 4x5 Rule

S300 I/O MODULES

The CK722 controller has two RS485 ports: RS485A and RS485B. They are used to communicate with field devices on the S300 bus. The controller can use *either* of the following communication settings:

- 19200 bps, no parity, 8 bits per character, and one stop bit
- 9600 bps, even parity, 8 bit per character, and one stop bit

Each CK722 can have two S300 buses and supports up to 32 field devices on each bus.

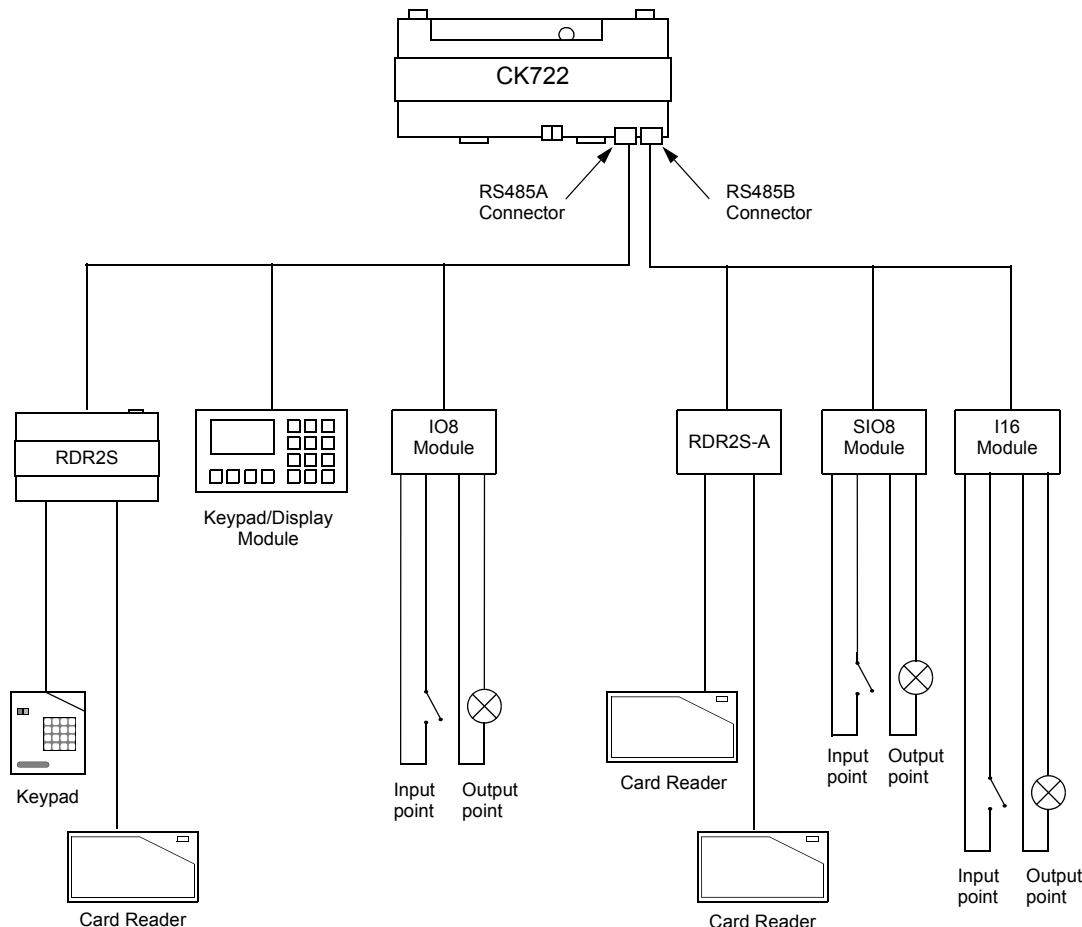


Figure 2-9: S300 Buses

For more information on the CK722 controller refer to the *CK722 Network Controller Hardware Installation Manual*.

S300-DIN-RDR2S

The S300-DIN-RDR2S module (also called “RDR2S” in this manual) provides two door access control input/output interface, each consisting of:

- Supervised door monitor switch input (normally open or normally closed, based on the external network configuration)
- Supervised auxiliary access or exit request switch input (normally open)
- Wiegand Data0 and Data1 interface
- Door strike relay (SPDT)
- Alarm shunt relay driver (open collector)
- Red lamp driver (open collector)
- Green lamp driver (open collector)

The RDR2S module supports S300 bus communications (RS-485). Auto baud rate detection is 9600/19200 baud.

The RDR2S module can be mounted on a DIN rail or on a flat surface.

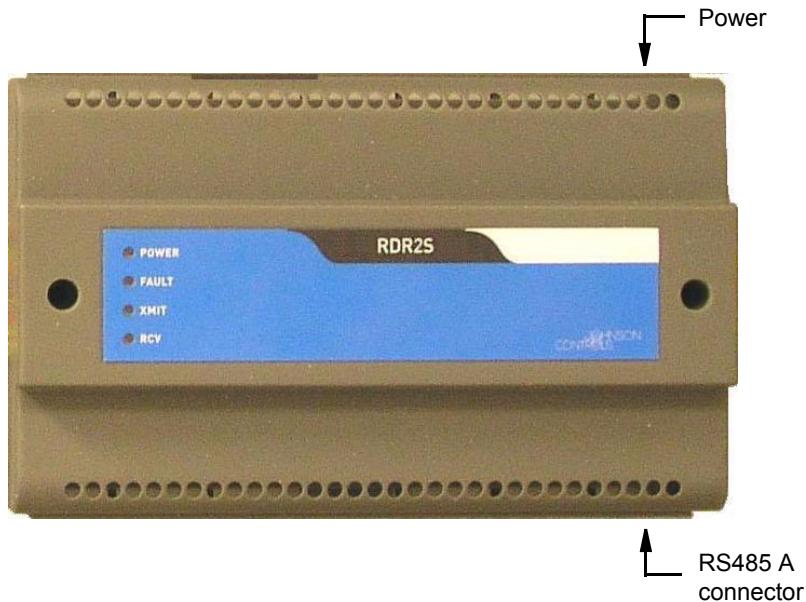


Figure 2-10: S300-DIN-RDR2S

See “Reader Module Comparison” on page 2-40 for a quick reference table listing features of reader modules.

S300-DIN-RDR2SA

The S300-DIN-RDR2SA module (also called “RDR2S-A” in this manual) provides interface control for access and security devices associated with a door. It supports up to two doors per unit. When interfacing to a single door, the unused points can be configured as general purpose input/output (I/O) points.

Each door access control input/output interface consists of:

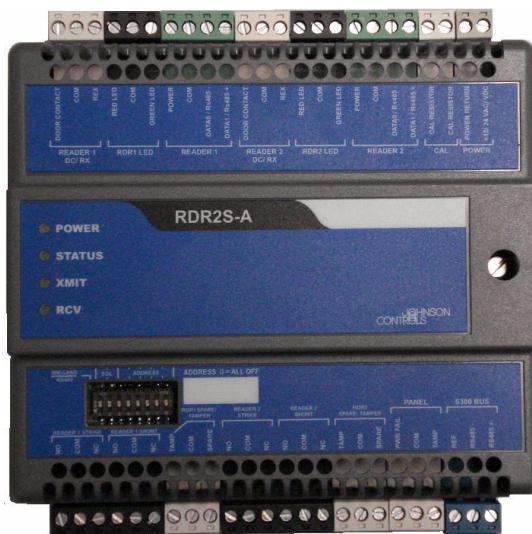
- Supervised door monitor switch input, normally open or normally closed, based on wired configuration
- Supervised auxiliary access or exit request switch input, normally open
- Supervised tamper and spare inputs
- Wiegand Data0 and Data1 interface
- Door strike relay, SPDT (Single Pole Double Throw)
- Alarm shunt relay, SPDT
- Red lamp driver and green lamp driver (open collectors)
- +12VDC 250mA reader power supply

Also, the following inputs are shared by both interfaces (one per unit):

- Calibration resistor input
- Supervised tamper and power fail inputs

The RDR2S-A module supports S300 bus communications (RS-485). Auto baud rate detection is 9600/19200 baud.

The RDR2S-A module can be mounted on a DIN rail or on a flat surface.



See “Reader Module Comparison” on page 2-40 for a quick reference table listing features of reader modules.

S300-KDM

The S300-KDM (also called the “Keypad/Display module” in this manual) is a keypad with an LCD display that connects to the CK722 panel and functions as the user interface for the intrusion detection system.

An authorized user can utilize the Keypad/Display module to:

- Arm or disarm an area
- Bypass or activate a zone
- Silence an annunciator
- Examine status of the areas, zones, annunciators, and active alarms
- Acknowledge intrusion alarms
- Reset or test a sensor

The display part of the module provides information necessary to guide the authorized user through the above actions. The S300-KDM reports the keys entered into the keypad to the CK722 panel as the authorized user navigates through the menus associated with the above functions.

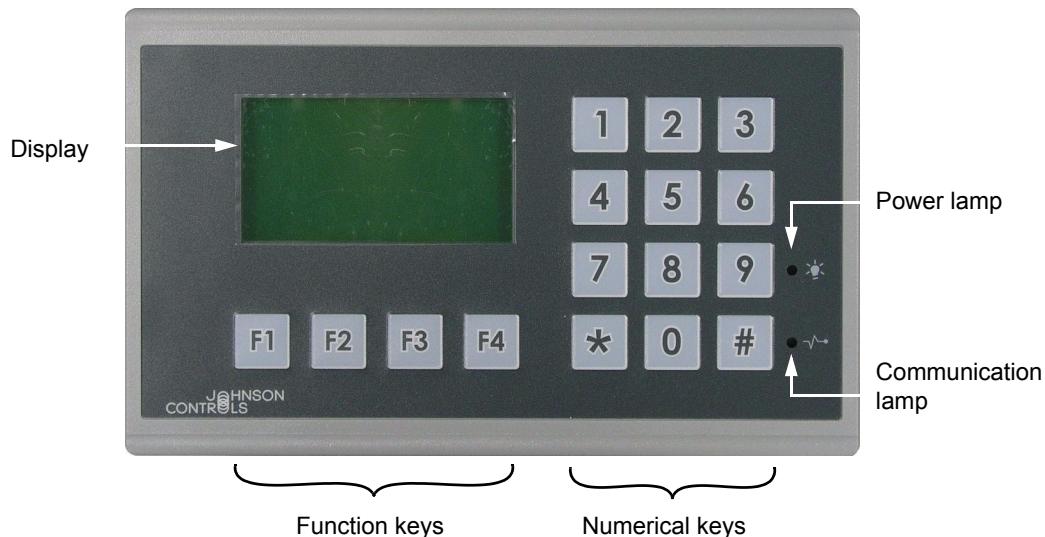


Figure 2-11: S300-KDM

Operation of the S300-KDM module is performed through the use of a Human Machine Interface (HMI) which consists of a keypad with 16 keys and a display area.

Keypad

The keys in the Keypad/Display module can be categorized into functional keys and numerical keys.

The functional keys take on different functions at different times. For example, the **F1** key can, at different occasions, arm an area, bypass a zone, silence an annunciator, etc. Current function of the key is indicated by the display above it.

Throughout this manual the functional keys are referred to by their function name rather by their **F1-F4** names. For example, an **F1** key which has been assigned the “arm” function will be called the **ARM** key.

The numerical keys are used to specify particular areas, zones, and annunciators.

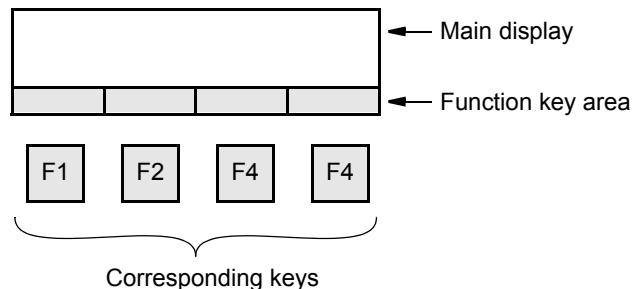
When a key is pressed, the Keypad/Display module makes a “beep” sound to signal that a command is pressed.

Function	Command Description
ARM	Access the menu to arm areas or arm the selected area
DSRM	Access the menu to disarm areas or disarm the selected area
STAT	Access the Status menu
EXIT	Exit the Main menu and return to the custom logo display
AREA	Access the list of areas
ZONE	Access the list of zones
ANNU	Access the list of annunciators
PREV	Display the previous item on the list (of areas, zones, or annunciators)
NEXT	Display the next item on the list (of areas, zones, or annunciators)
SLNC	Silence the annunciator
BYPS	Bypass the selected zone
ACTV	Activate the selected zone
ACK	Acknowledge an alarm
RST	Reset sensor
TEST	Test sensor (activate)
STOP	Stop sensor test (de-activate)
MORE	Scroll through the function key menu

Display

The display is divided into two sections:

- The main display area, showing the current menu item, status messages, and error messages.
- The function key area, showing the currently available user functions.

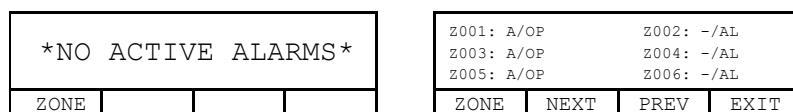


The display may be in one of the following modes:

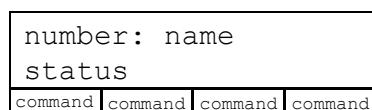
- Offline mode: The Keypad/Display module is powered up and the communication with the controller is not yet established or has been lost. The Keypad/Display module displays the custom logo



- Inactive mode: The display depends on the Display_Overview attribute configuration and whether there are any active alarms at the time. Possible displays: custom logo, "No Active Alarms" screen, or an overview screen



- Command mode: Depending on the access rights, various menu items and commands are accessible for viewing and executing.



NOTE

*Press ** to exit the command mode.
Press *7 to go to the overview mode.*

Overview Screens

The display of the overview screens depends on the configuration of the Display_Overview attribute. The following overview modes are available:

- “All Alarm and Status”
- “All Alarms Only”
- “Select Alarm and Status”
- “Select Alarms Only”

In each mode, up to 12 entries can be displayed at a time.

Press the **EXIT** key in any of the status screens to enter the command mode.

If the overview mode is disabled, when you swipe a valid badge or enter the card identification number the command screen will be displayed.

“All Alarm and Status” Mode

This mode does not require access rights and consists of four overview screens:

- Active Alarms/Bypass - Displays zones with active alarm/bypass status.
- Zones - Displays zones with any status.
- Areas - Displays areas with any status.
- Annunciators - Displays annunciators with any status.

➤ To browse the “All Alarm and Status” screens:

1. The first screen, Active Alarms/Bypass, is displayed when the Keypad/Display Module is idle. Use the **NEXT** and **PREV** keys to view all zones in the list. Press the **ZONE** key to go to Zones screen.

Active Alarms/Bypass			
Z001: A/OP	Z002: A/AL		
Z003: A/SH	Z004: A/OP		
Z005: A/TA	Z006: A/AL		
Z008: A/TR	Z018: A/UN		
Z019: A/AL	Z020: A/AL		
Z031: B/AL	Z032: B/AL		
ZONE	NEXT	PREV	EXIT

Arm status:		Alarm status:	
A	- armed	AL	- in alarm
-	- disarmed	-	- not in alarm
B	- bypassed	OP	- alarm open
F	- faulted	SH	- alarm short
		TR	- trouble alarm
		TA	- tamper alarm
		UN	- unknown

2. In the Zones screen, use the **NEXT** and **PREV** keys to view all zones in the list. Press the **AREA** key to go to Areas screen.

Zones:			
Z001: A/OP	Z002: -/AL		
Z003: A/SH	Z004: -/--		
Z005: A/TA	Z006: B/OP		
Z007: A/TR	Z009: A/--		
Z009: A/AL	Z010: A/AL		
Z011: -/UN	Z012: B/--		
AREA	NEXT	PREV	EXIT

Arm status:		Alarm status:	
A	- armed	AL	- in alarm
-	- disarmed	-	- not in alarm
B	- bypassed	OP	- alarm open
F	- faulted	SH	- alarm short
		TR	- trouble alarm
		TA	- tamper alarm
		UN	- unknown

3. In the Areas screen, use the **NEXT** and **PREV** keys to view all areas in the list. Press the **ANNC** key to go to Annunciators screen.

Areas:	
A001: A	A002: M
A003: A	A004: -
A005: F	A006: -
A007: A	A009: A
A009: A	A010: A
A011: -	A012: M

ANNC NEXT PREV EXIT

Arm status:
 A - armed
 - - disarmed
 U - arming/disarming
 F - faulted

- In the Annunciators screen, use the **NEXT** and **PREV** keys to view all annunciators in the list. Press the **ALRM** key to return to the “Active Alarms/Bypass” screen or press the **EXIT** key to enter the command mode.

Annunciators:	
S001: A	S002: -
S003: A	S004: -
S005: -	S006: -
S007: A	S009: A
S009: A	S010: A
S011: -	S012: -

ALRM NEXT PREV EXIT

Status:
 A - active
 - - inactive

“All Alarms Only” Mode

This mode does not require access rights and consists of the following screen:

- Active Alarms/Bypass screen - Lists zones with active alarm/bypass status.

► To use the Active Alarms/Bypass screen:

- The following screen is displayed when the Keypad/Display Module is idle. Use the **NEXT** and **PREV** keys to view all zones in the list. Press the **EXIT** key to enter the command mode.

Active Alarms/Bypass	
Z001: A/OP	Z002: A/AL
Z003: A/SH	Z004: A/OP
Z005: A/TA	Z006: A/AL
Z008: A/TR	Z018: A/UN
Z019: A/AL	Z020: A/AL
Z031: B/AL	Z032: B/AL

NEXT PREV EXIT

Arm status:	Alarm status:
A - armed	AL - in alarm
- - disarmed	- - not in alarm
B - bypassed	OP - alarm open
F - faulted	SH - alarm short
	TR - trouble alarm
	TA - tamper alarm
	UN - unknown

“Select Alarm and Status” Mode

This mode is similar to the “All Alarm and Status” mode, but the overview screens are displayed only after you swipe a valid badge or enter the card identification number. The items displayed depend on the access rights.

“Select Alarms Only” Mode

This mode is similar to the “All Alarms Only” mode, but the overview screen is displayed only after you swipe a valid badge or enter the card identification number. The items displayed depend on the access rights.

Commands

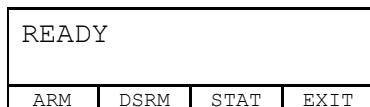
You can perform the following functions using the Keypad/Display module (a valid badge or valid key sequence is required to access the menu):

AREA	ZONE	ANNUNCIATOR
— ARM	— BYPASS	— SILENCE
— DISARM	— ACTIVATE	
— BYPASS zone*	— ACKNOWLEDGE alarm	
	— RESET sensor	
	— TEST sensor	
	— STOP sensor test	

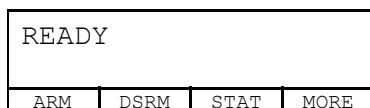
* Available only if arming of area is faulted

► To access the command mode:

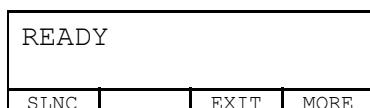
1. Swipe a valid badge or enter the card identification number.
2. If the overview mode is enabled, press the **EXIT** key in the overview screen.
3. Verify that the display reads: “READY” and the following keys appear: **ARM**, **DSRM**, **STAT**, and **EXIT**.



During an alarm condition, the **MORE** key will appear on the Main menu. Press this key to access the rest of the function key descriptions.



The next screen displays the **SLNC** key as well as **EXIT** and **MORE**.

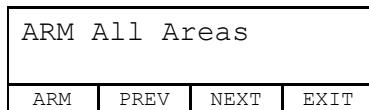


4. Depending on the menu displayed, you have the following choices:
 - Press the **ARM** key to access the menu to arm area(s).
 - Press the **DSRM** key to access the menu to disarm area(s).
 - Press the **STAT** key to access the Status menu.
 - Press the **EXIT** key to exit the Main menu and return to the custom logo display.
 - Press the **MORE** key to scroll through the function key menu (during alarm conditions only).
 - Press the **SLNC** key to go to the screen where you can select active annunciator and silence it (during alarm conditions only).

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the custom logo display.

► **To arm area(s):**

1. Swipe a valid badge or enter the card identification number.
2. *If the overview mode is enabled*, press the **EXIT** key in the overview screen.
3. To arm all areas, press **ARM**.

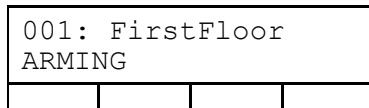


Alternatively, to browse for a specific area, use the **PREV** and **NEXT** keys or type the area number and press the # key.

When the correct area is displayed, press the **ARM** key.



4. Wait till arming is complete.



5. Verify that the status of the area(s).



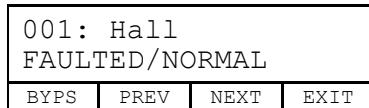
6. From here you have the following choices:

- Wait for the operation to time out. The default time out period is 15 seconds. After that time the Keypad/Display module will display the custom logo.
- To access another area to arm/disarm, use the **PREV** and **NEXT** keys or type the area number and press the # key.
- Press the **EXIT** key to return to the Main menu.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► **Special handling if arming of an area is faulted:**

1. If any zone is faulted during arming of an area, you are directed to separate screen that allows you to bypass faulted zone and retry arming:



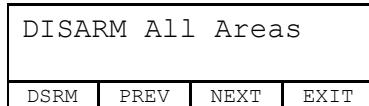
2. From here you have the following choices:

- Press **BYPAS** to bypass the zone.
- If more than one zone is faulted, use the then **PREV** and **NEXT** keys to access other faulted zones.
- Press the **EXIT** key to return to the screen menu which allows you to arm area(s).

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► **To disarm area(s):**

1. Swipe a valid badge or enter the card identification number.
2. *If the overview mode is enabled*, press the **EXIT** key in the overview screen.
3. To disarm all areas, press **DSRM**.



Alternatively, to browse for a specific area, use the **PREV** and **NEXT** keys or type the area number and press the # key.

When the correct area is displayed, press the **DSRM** key.



4. Wait till disarming is complete.



5. Verify that the status of the area(s).



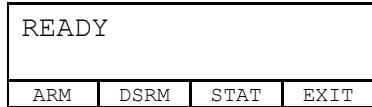
6. From here you have the following choices:

- Wait for the operation to time out. The default time out period is 15 seconds. After that time the Keypad/Display module will display the custom logo.
- To access another area to arm/disarm, use the **PREV** and **NEXT** keys or type the area number and press the # key.
- Press the **EXIT** key to return to the Main menu.

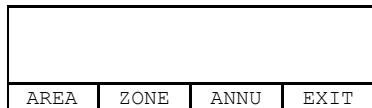
If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► To bypass a zone:

1. Swipe a valid badge or enter the card identification number.
2. If the overview mode is enabled, press the **EXIT** key in the overview screen.
3. Press the **STAT** key.



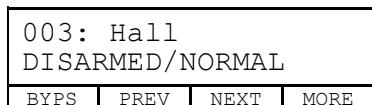
4. Press the **ZONE** key.



5. The display will show the first zone in the list and its status.

To browse for a different zone, use the **PREV** and **NEXT** keys or type the zone number and press the # key.

When the correct zone is displayed, press the **BYPS** key.



6. Verify that the zone has been bypassed.

003: Hall BYPASSED/NORMAL			
ACTV	PREV	NEXT	EXIT

7. From here you have the following choices:

- To access another zone to bypass/activate, use the **PREV** and **NEXT** keys or type the zone number and press the # key.
- Press the **EXIT** key to return to the Status menu.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► **To activate a zone:**

1. Swipe a valid badge or enter the card identification number.
2. *If the overview mode is enabled*, press the **EXIT** key in the overview screen.
3. Press the **STAT** key.

READY			
ARM	DSRM	STAT	EXIT

4. Press the **ZONE** key.

AREA	ZONE	ANNU	EXIT

5. The display will show the first zone in the list and its status.

To browse for a different zone, use the **PREV** and **NEXT** keys or type the zone number and press the # key.

When the correct zone is displayed, press the **ACTV** key.

003: Hall BYPASSED/NORMAL			
ACTV	PREV	NEXT	MORE

6. Verify that the zone has been activated.

003: Hall DISARMED/NORMAL			
BYPS	PREV	NEXT	EXIT

7. From here you have the following choices:

- To access another zone to bypass/activate, use the **PREV** and **NEXT** keys or type the zone number and press the # key.
- Press the **EXIT** key to return to the Status menu.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► **To acknowledge an alarm:**

1. Swipe a valid badge or enter the card identification number.
2. *If the overview mode is enabled*, press the **EXIT** key in the overview screen.
3. Press the **STAT** key.

READY			
ARM	DSRM	STAT	EXIT

4. Press the **ZONE** key.

AREA	ZONE	ANNU	EXIT

5. The display will show the first zone in the list and its status.

To browse for a different zone, use the **PREV** and **NEXT** keys or type the zone number and press the # key.

When the correct zone is displayed, press the **MORE** key.

003: Hall ARMED/ALARMED			
ACTV	PREV	NEXT	MORE

6. Press the **ACK** key.

003: Hall ARMED/ALARMED			
ACK	RST	EXIT	MORE

7. Verify that the alarm has been acknowledged.

003: Hall *ALARM ACKNOWLEDGED*			
ACK	RST	EXIT	MORE

8. “ALARM ACKNOWLEDGED” will flash for a few seconds on the status line of the display, then the status will return back to display current zone status.

003: Hall ARMED/NORMAL			
ACK	RST	EXIT	MORE

9. From here you have the following choices:
- Use the **MORE** key to scroll the menu.
 - Press the **RST** key to reset sensor(s) associated with this zone.
 - Type the zone number and press the # key to access a specific zone.
 - Press the **EXIT** key to return to the Status menu.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► **To reset sensor(s) associated with a zone:**

1. Swipe a valid badge or enter the card identification number.
2. *If the overview mode is enabled*, press the **EXIT** key in the overview screen.
3. Press the **STAT** key.

READY			
ARM	DSRM	STAT	EXIT

4. Press the **ZONE** key.

AREA	ZONE	ANNU	EXIT

5. The display will show the first zone in the list and its status.

To browse for a different zone, use the **PREV** and **NEXT** keys or type the zone number and press the # key.

When the correct zone is displayed, press the **MORE** key.

003: Hall ARMED/ALARMED			
ACTV	PREV	NEXT	MORE

6. Press the **RST** key. All the outputs listed in the zone Reset_Output_Attribute_List will be activated.

003: Hall ARMED/ALARMED			
ACK	RST	EXIT	MORE

7. Verify that the sensor(s) have been reset.

003: Hall *SENSOR RESET*			
ACK	RST	EXIT	MORE

8. “SENSOR RESET” will flash for a few seconds on the status line of the display, then the status will return back to display current zone status.

003: Hall ARMED/NORMAL			
ACK	RST	EXIT	MORE

9. From here you have the following choices:

- Use the **MORE** key to scroll the menu.
- Type the zone number and press the # key to access a specific zone.
- Press the **EXIT** key to return to the Status menu.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

► To test sensor(s) associated with a zone:

1. Swipe a valid badge or enter the card identification number.
2. *If the overview mode is enabled*, press the **EXIT** key in the overview screen.
3. Press the **STAT** key.

READY			
ARM	DSRM	STAT	EXIT

4. Press the **ZONE** key.

AREA	ZONE	ANNU	EXIT

5. The display will show the first zone in the list and its status.

To browse for a different zone, use the **PREV** and **NEXT** keys or type the zone number and press the # key.

When the correct zone is displayed, press the **MORE** key.

003: Hall ARMED/ALARMED			
ACTV	PREV	NEXT	MORE

6. Press the **MORE** key again.

003: Hall ARMED/ALARMED			
ACK	RST	EXIT	MORE

7. Press the **TEST** key. All the outputs listed in the zone Test_Ouput_Attribute_List will be activated.

003: Hall *ZONE TEST START*			
TEST	STOP		MORE

8. Press the **STOP** key. All the outputs listed in the zone Test_Ouput_Attribute_List and the Reset_Ouput_Attribute_List will be de-activated.

003: Hall *TEST/RESET STOP*			
TEST	STOP		MORE

9. From here you have the following choices:

- Use the **MORE** key to scroll the menu.
- Type the zone number and press the # key to access a specific zone.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

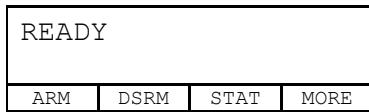
► To silence an annunciator:

NOTE

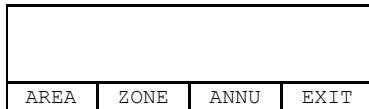
*The simplest way to silence an annunciator is to use the **SLNC** key that appears on the command menu during an alarm condition. See page 2-19 for details. The steps below describe how to silence an annunciator using the Status menu.*

1. Swipe a valid badge or enter the card identification number.
2. If the overview mode is enabled, press the **EXIT** key in the overview screen.

3. Press the **STAT** key.



4. Press the **ANNU** key.



5. The display will show the first annunciator in the list and its status.
To browse for a different annunciator, use the **PREV** and **NEXT** keys or type in the zone number and press the # key.
When the correct annunciator is displayed, press the **SLCN** key.



6. Verify that the annunciator has been silenced.



7. From here you have the following choices:
 - To access another annunciator to bypass/activate, use the **PREV** and **NEXT** keys or type the annunciator number and press the # key.
 - Press the **EXIT** key to return to the Status menu.

If no action is taken, after the time out period (default value is 15 seconds) the Keypad/Display module will return to the inactive mode display.

S300-RDR2

The S300-RDR2 module (also called “RDR2” in this manual) is a reader terminal with an interface for two readers. Each interface provides:

- Cardkey/Wiegand, one wire data interface
- Sensor Wiegand, two wire data interface
- Four rows by four columns, 16 button keypad interface
- Red LED or incandescent bulb reader lamp drivers
- Green LED or incandescent bulb reader lamp drivers
- +5 VDC reader power output at 20 milliamperes
- +12 VDC reader power output at 150 milliamperes

The RDR2 also provides two door input/output interfaces each consisting of:

- A two-state door monitor switch input (secure = normally closed)
- A two-state auxiliary access or exit request switch input (normally open)
- A door-strike relay (SPDT)
- An alarm shunt relay driver (open collector)

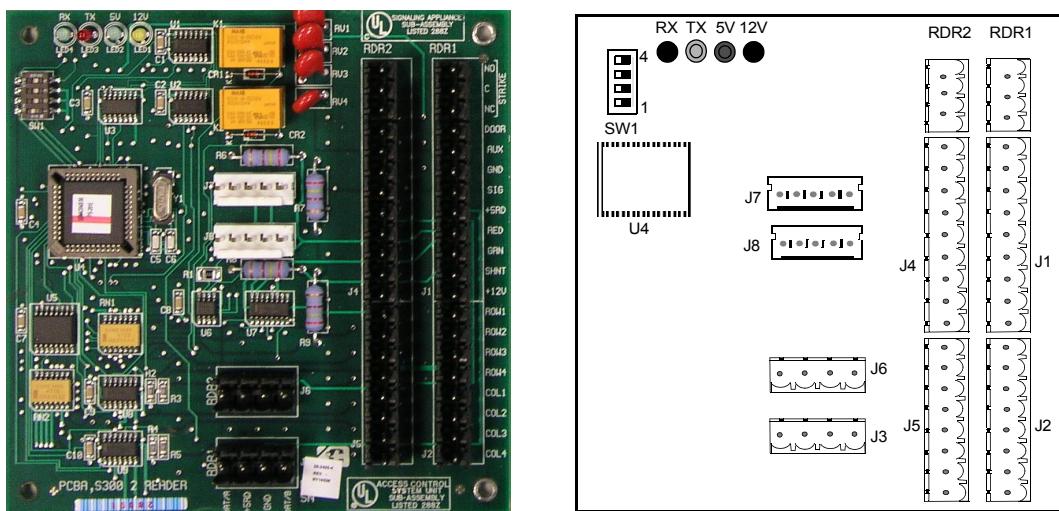


Figure 2-12: S300-RDR2 Field

See “Reader Module Comparison” on page 2-40 for a quick reference table listing features of reader modules.

S300-I16

The S300-I16 field device (also called “I16” in this manual) is an unsupervised input terminal board. It provides:

- 16 two-state input points
- 16 red LED indicators which illuminate when each input is in alarm

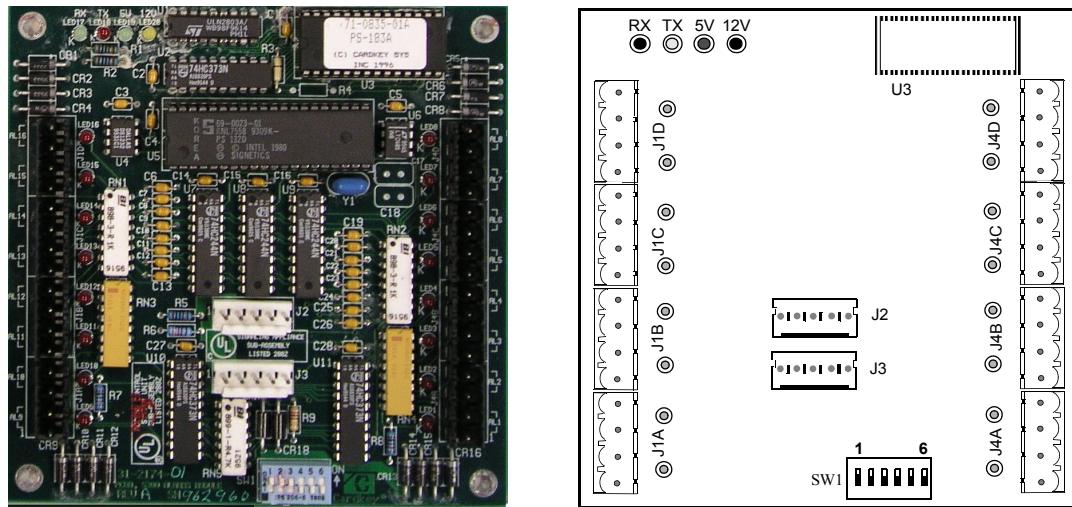


Figure 2-13: S300-II16

S300-IO8

The S300-IO8 field device (also called “IO8” in this manual) is an unsupervised input/output terminal board. It provides:

- 8 two-state input points
- 8 general purpose SPDT output relays
- 8 input LEDs which illuminate when each input is in alarm
- 8 output LEDs showing when each relay is energized

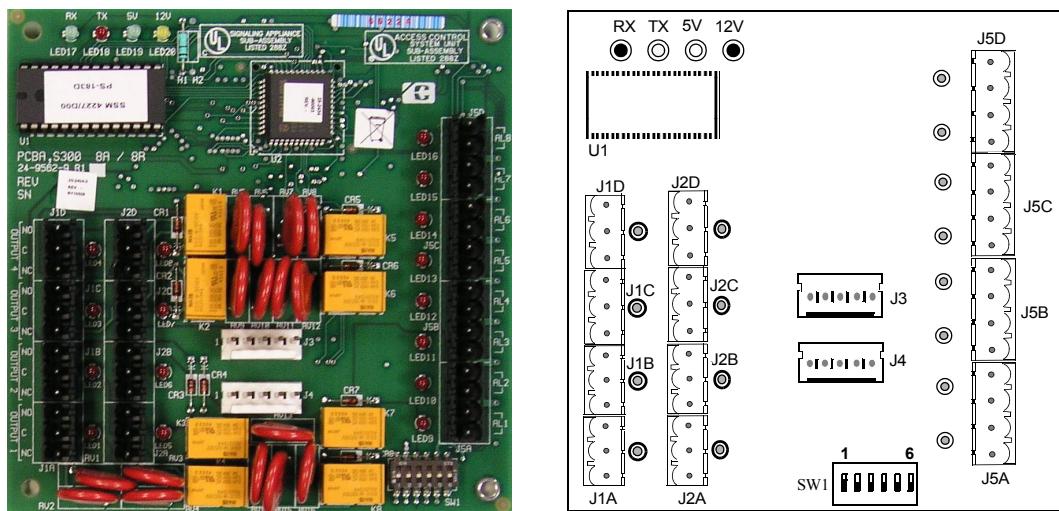


Figure 2-14: S300-IO8

S300-SIO8

The S300-SIO8 field device (also called “SIO8” in this manual) is a supervised input/output terminal board. It provides:

- 8 four-state alarm inputs, which monitor open or short circuit, alarm, and secure states.

The eight alarm input LEDs are three-color indicators showing:

Off	-	Open
Green	-	Secure
Yellow	-	Short
Red	-	Alarm

- 8 general purpose SPDT output relays
- 8 input LEDs which illuminate when each input is in alarm
- 8 output LEDs showing when each relay is energized

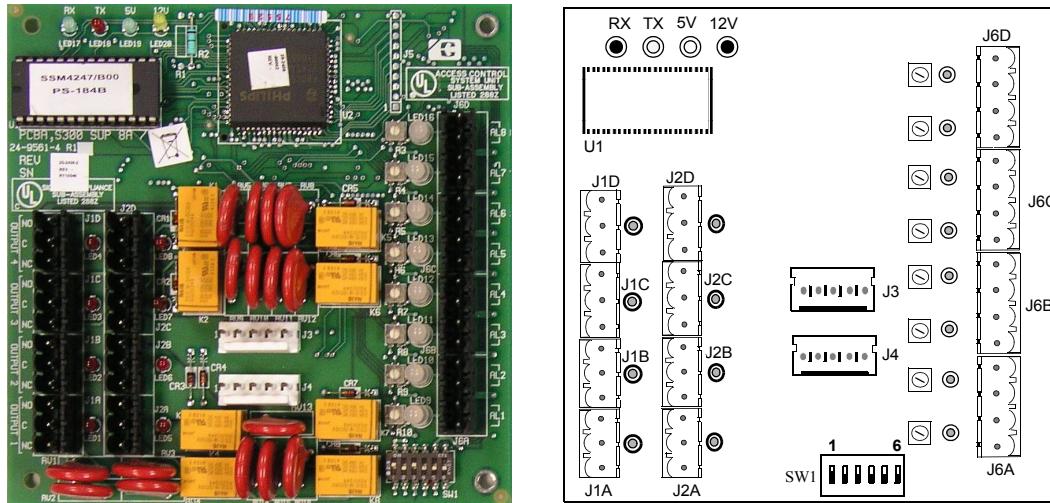


Figure 2-15: S300-SIO8

S300-SI8

The S300-SI8 field device (also called “SI8” in this manual) is a supervised input terminal board. It provides:

- 8 four-state (open or short circuit, alarm, and secure) alarm inputs.

The eight alarm input LEDs are three-color indicators showing:

Off	-	Open
Green	-	Secure
Yellow	-	Short
Red	-	Alarm

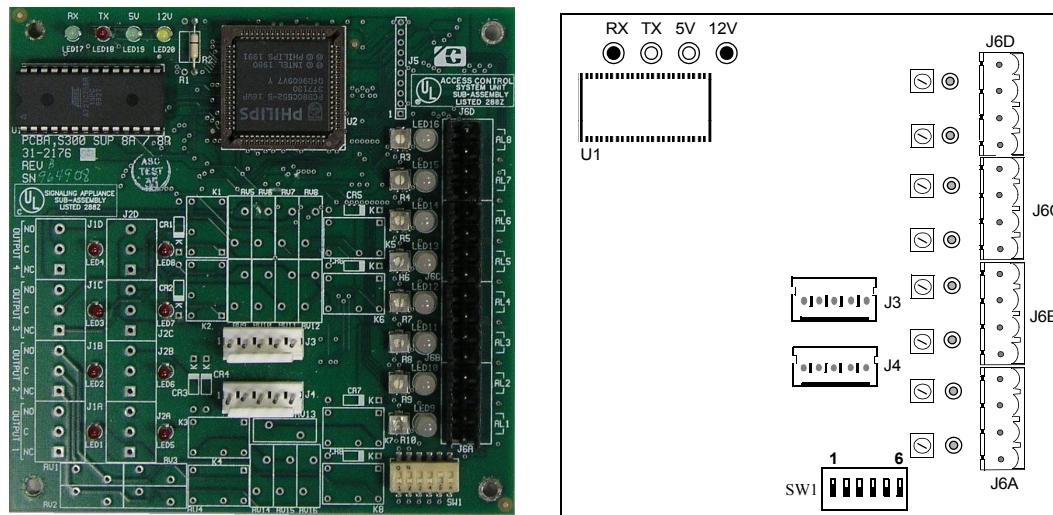


Figure 2-16: S300-SI8

ENCLOSURES

There are two types of enclosures used with the CK722 controller and field devices:

- S300-DIN enclosures for housing DIN-mountable devices like the CK722 itself. These include S300-DIN-L and S300-DIN-S.
- S300 expansion enclosures for housing I/O field devices and S300-RDR2. These include S300-XL, S300-XS, and S300-XXS.

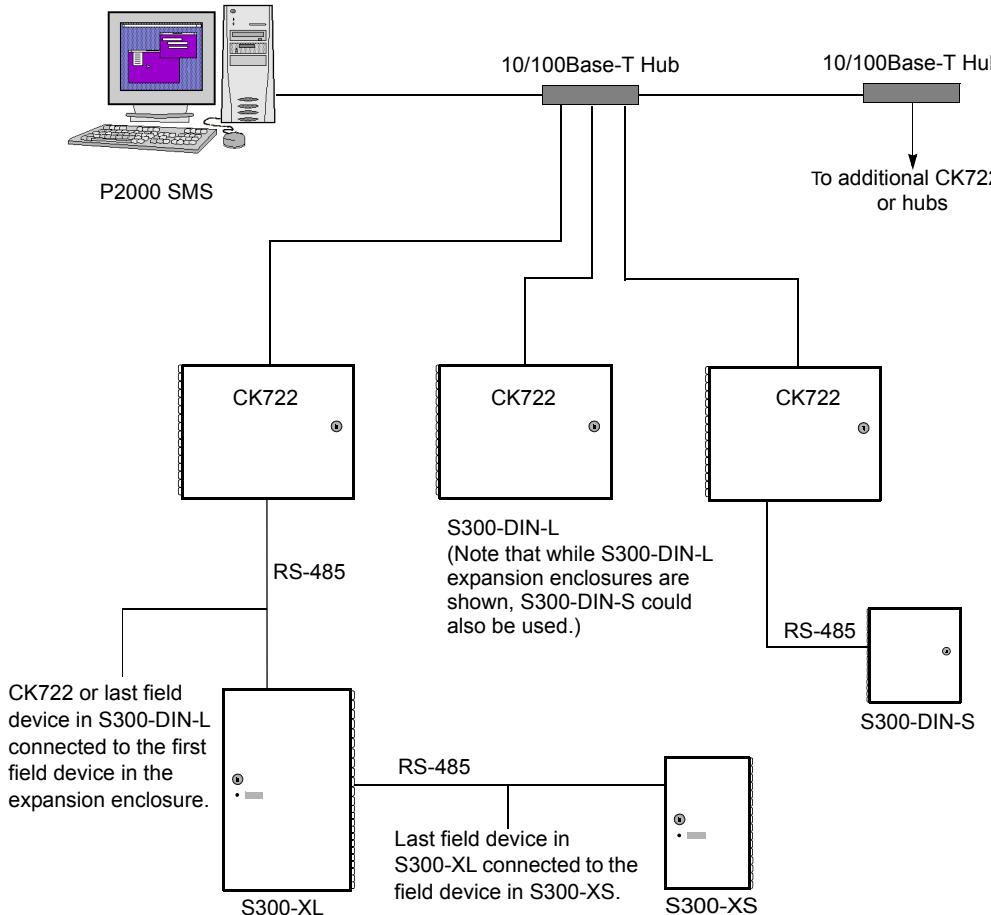


Figure 2-17: Example of the System Configuration

S300-DIN-L

The S300-DIN-L is a large enclosure used to house DIN-mountable field devices in the CK722 system. It can also be used to house the CK722 controller itself.

The S300-DIN-L comes with a backplate, a tamper switch, a lock, and a ground strap kit that have to be installed. The backplate contains a power supply and DIN rails for module mounting.

Up to three modules can be mounted on the DIN rails (three S300-DIN-RDR2S modules, or one CK722 module and two S300-DIN-RDR2S modules).

The enclosure can also hold a backup battery unit composed of two 12V lead-acid batteries in two battery brackets.

Figure 2-18 gives you an overview of the large enclosure with all components installed. The modules shown here are S300-DIN-RDR2S.

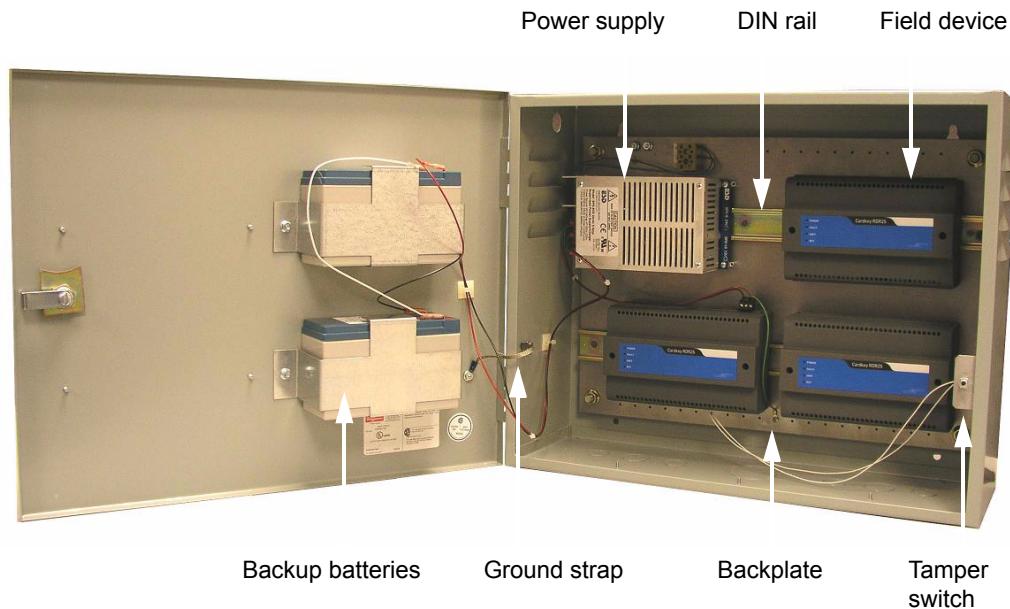


Figure 2-18: S300-DIN-L With Installed Components

S300-DIN-S

The S300-DIN-S is a small enclosure used to house DIN-mountable field devices in the CK722 system. It can also be used to house the CK722 controller itself.

The S300-DIN-S comes with a backplate, a tamper switch, a lock, and a ground strap kit that have to be installed. The backplate contains a power supply and a DIN rail for mounting of one S300-DIN-RDR2S, or CK722 module.

The enclosure can also hold a backup battery unit composed of two 12V lead-acid batteries in one battery bracket.

Figure 2-19 gives you an overview of the small enclosure with all components installed. The module shown here is S300-DIN-RDR2S.

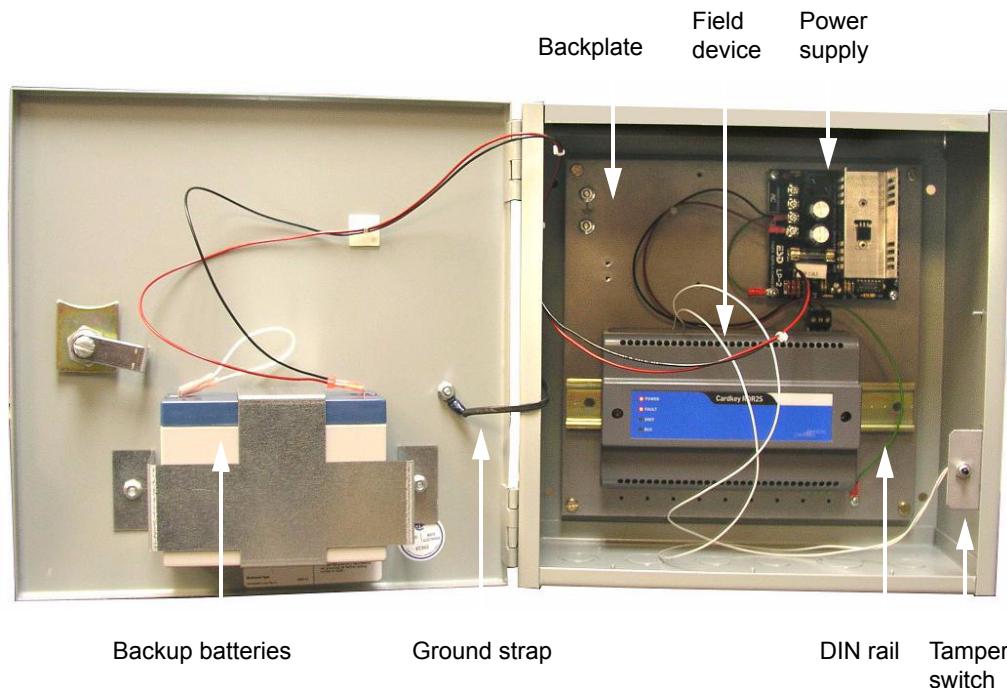


Figure 2-19: S300-DIN-S With Installed Components

S300-XL

S300-XL is a large expansion enclosure that contains a power supply, a tamper switch, a power indicator light, and a lock. It has room for nine additional field devices and for a battery back-up unit.

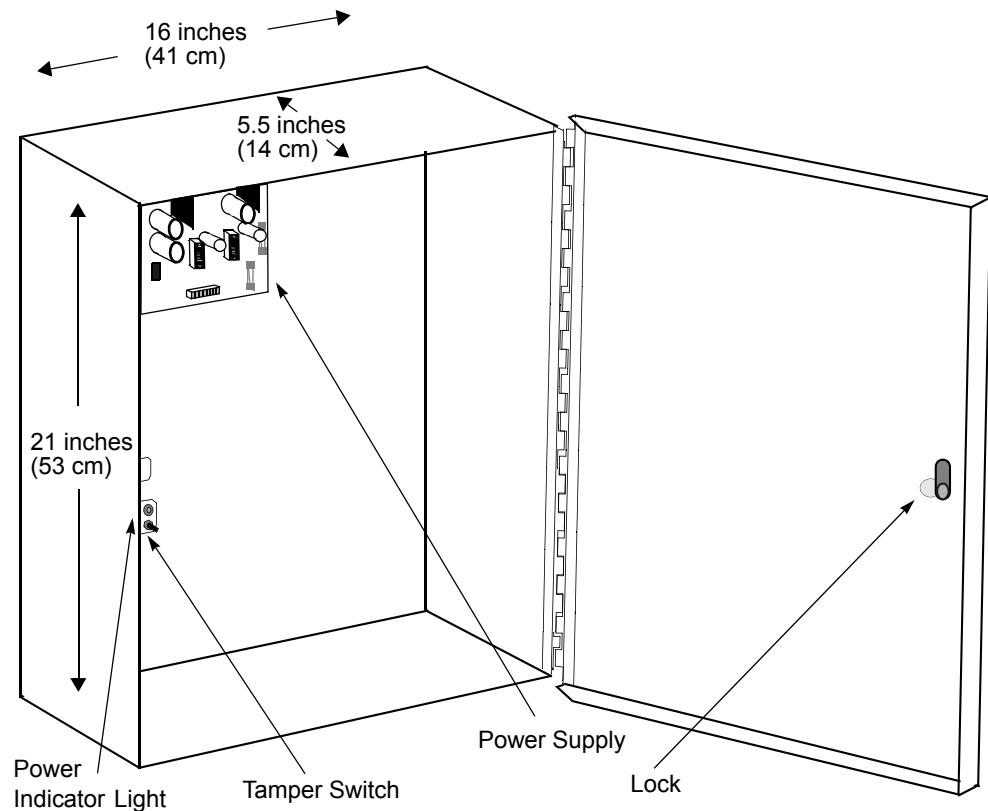


Figure 2-20: S300-XL

S300-XS

S300-XS is a small expansion enclosure that contains a power supply, a tamper switch, a power indicator light, and a lock. It has room for five additional field devices and for a battery back-up unit.

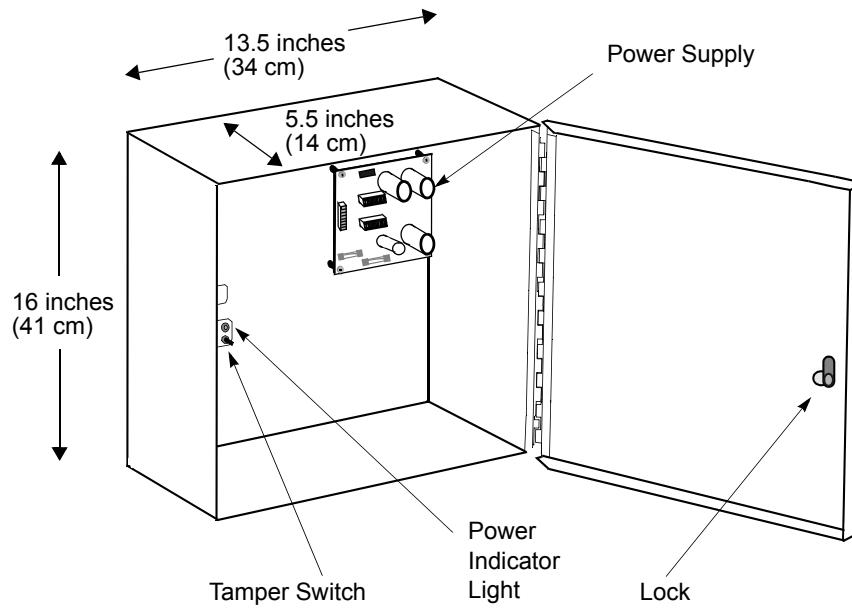


Figure 2-21: S300-XS

S300-XXS

S300-XXS is an extra small expansion enclosure that contains a power supply, a tamper switch, a power indicator light, and a lock. It has room for two additional field devices and for a battery backup unit.

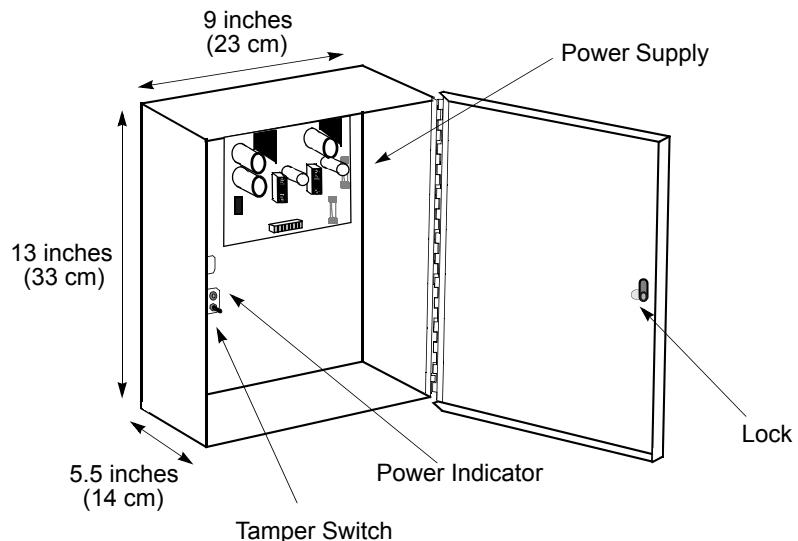


Figure 2-22: S300-XXS

READER MODULE COMPARISON

Table 2-1: Reader Module Comparison Chart

Feature	S300-RDR2	S300-DIN-RDR2S Dual Reader Mode with I/O Enabled	S300-DIN-RDR2SA Dual Reader Mode	S300-DIN-RDR2S Single Reader Mode with I/O Enabled	S300-DIN-RDR2SA Single Reader Mode
Power	5+12+ VDC	24 VDC	12-24 AC or DC	24 VDC	12-24 AC or DC
Reader Power	5 or 12 VDC	5, 12, 24 VDC	12 VDC	5, 12, 24 VDC	12 VDC
Door Power	External	12 or 24 VDC (300 mA)	External	12 or 24 VDC	External
Supervised Inputs	No	Yes (150 to 2.4 kOhm)	Yes (150 to 3.6 kOhm)	Yes (150 to 2.4 kOhm)	Yes (150 to 3.6 kOhm)
Door Input	2	2	2	1	1
Power Fail	0	0	1	0	1
Low Battery	0	0	1 ¹	0	1 ¹
Request to Exit/AUX Input	2	2	2	1	1
Tamper	0	0	2 Reader 1 Enclosure	0	1 Reader 1 Enclosure
General Purpose Inputs	0	2	2	4	5
Door Strike Relays	2	2	2	1	1
General Purpose Outputs	0	2 Open Collector Outputs	0	1 Relay 5 Open Collector	2 Relay 2 Open Collector
Alarm Shunt Driver	2 ²	2 ²	2 (Relay)	1	1 (Relay)
Alarm Driver	0	0	0	0	0
Wiegand Input	1 and 2 Wire	2 Wire	2 Wire	2 Wire	2 Wire
Keypad	Column and Row/ Wiegand	Wiegand	Wiegand	Wiegand	Wiegand

1.Internal indicator built in to the reader module – no separate physical connection required for low battery indication.

2.Open Collector

CK722 ARCHITECTURE

This chapter gives you a brief description of the objects used by CK722 as well as the RS485 bus.

Figures 3-1 to 3-4 give you examples how objects interact with each other and how you can use them to address the needs of a particular Security Management System. This section is **not intended** to guide you through the configuration process, but rather to illustrate the idea of translating your system design and intended functionality into a map of cross-linked objects.

For detailed information on how to add and configure objects, refer to the *P2000 System Configuration Tool (SCT) Manual*.

Figure 3-1 shows a combination of three objects comprising a simple access control solution.

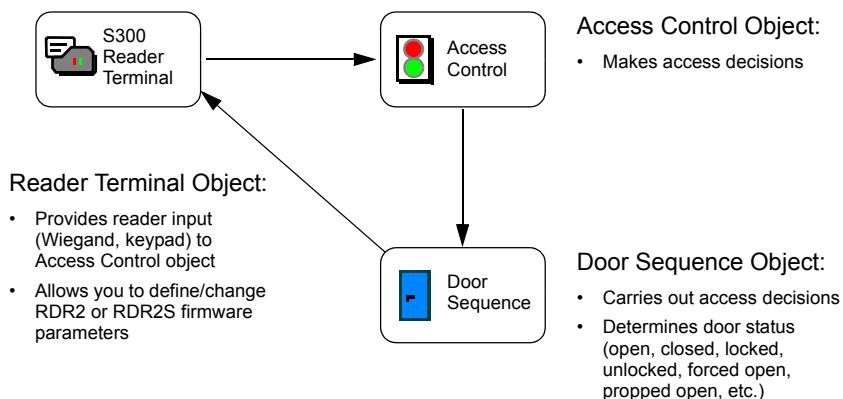


Figure 3-1: Example: Three Interlinked Objects

Figure 3-2 illustrates adding a second Reader Terminal object to a single Access Control object. This allows you to use the same set of access rules for two separate readers.

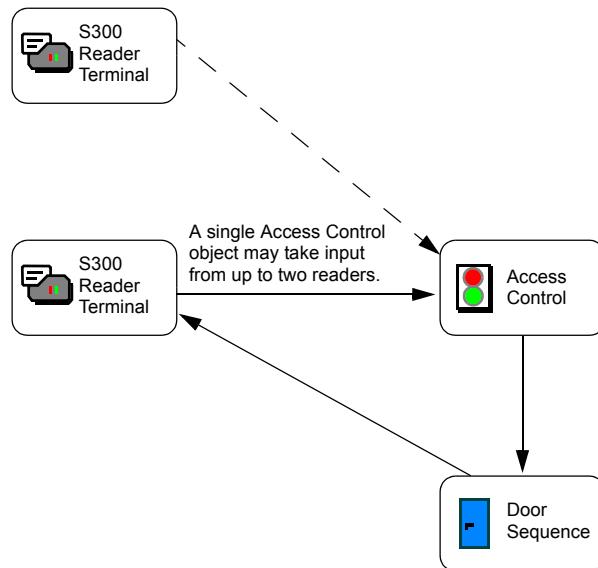


Figure 3-2: Example: Adding a Second Reader Terminal Object

Figure 3-3 illustrates adding a Schedule object to the system. You can use this object to modify functions of other objects based on time of day.

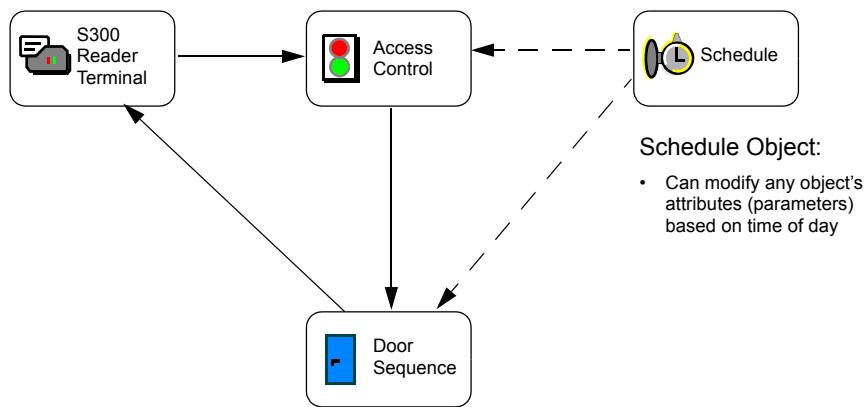


Figure 3-3: Example: Adding a Schedule Object

Figure 3-4 illustrates how to incorporate additional functionality into your system by adding Occupancy, Anti-Passback, and Anti-Loitering objects.

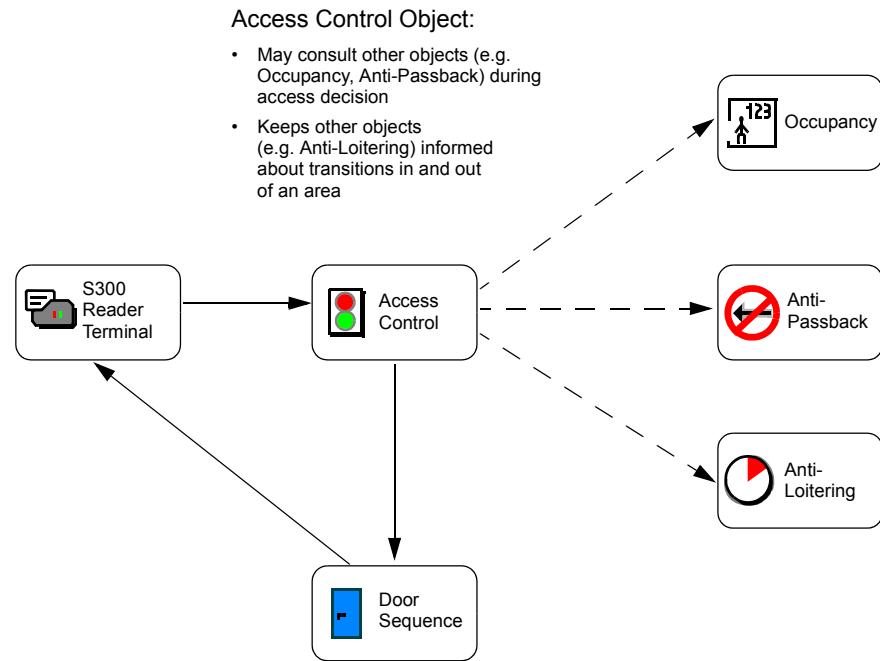


Figure 3-4: Example: Adding Occupancy, Anti-Passback, and Anti-Loitering Objects

Figure 3-5 illustrates basic motion detection Application.

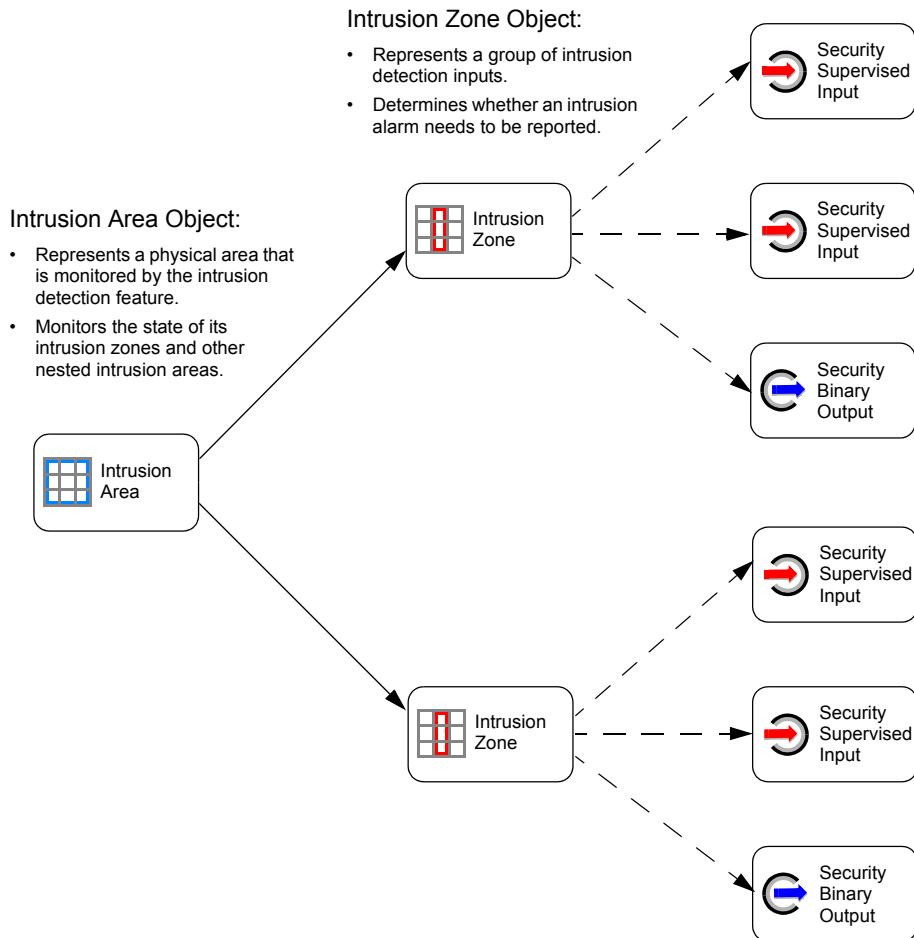


Figure 3-5: Example: Basic Motion Detection Application.

BACNET AND OBJECT ENGINE

The CK722's applications are made available as functional blocks called *objects*.

Each object allows you to configure the parameters of its function, for example, which facility codes are allowed at a door, or how long the door shall be unlocked upon an access grant. These parameters are called *configurable attributes*.

Each object also has *read only attributes* from which the operational state of the object can be read, for example, whether a door is open or closed, locked or unlocked.

Applications are built by selecting the objects necessary to achieve the desired functionality, and adjusting their configurable attributes with the System Configuration Tool (SCT).

Objects can be interconnected to create highly customized applications that even span multiple CK722 controllers, using Johnson Controls' *MCE Peer-to-Peer* technology.

Applications can also be created by selecting pre-defined *templates* at the SCT. Templates are "rubber-stamps" for common applications and have most of their configurable attributes already filled in. You can also create new templates with the SCT.

Attributes Common to All Objects

All objects inherit a set of attributes called "common attributes." Each object type also has specific attributes that apply only to that object type.

For detailed information refer to the *General Object Information* manual.

Attributes Common to Device Objects

In addition to "common attributes," all device objects inherit a set of attributes called "common device attributes." Each object type also has specific attributes that apply only to that object type.

The common device attributes represent the externally visible characteristics of a controller. It allows you to see data about the controller, such as the IP address and the current date and time. One Device object exists per controller.

Currently these attributes are only used by the CK722 controller. For detailed information refer to the *CK722 Device Object* manual.

Object Overview

Based on their function, objects can be divided into the following groups:

- Hardware-related objects
- Basic application objects
- Advanced application objects
- Intrusion detection objects
- Custom logic objects
- Elevator related objects
- Miscellaneous objects
- Internal and diagnostic objects

Also, at times some object types may be referred to as Sites, Supervisory Devices, Integrations, Field Devices, or Field Points.

“Object List” on page 3-7 contains an overview of all object types, with the respect to the aforementioned categories. The list is followed by object descriptions and illustrations of object interactions.

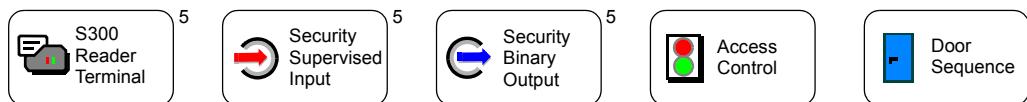
For even more detailed information, including object attributes, refer to the object manuals.

Object List

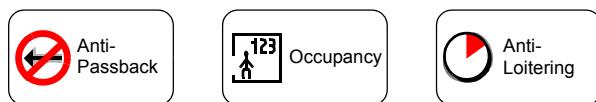
Hardware-related objects:



Basic application objects:



Advanced application objects:



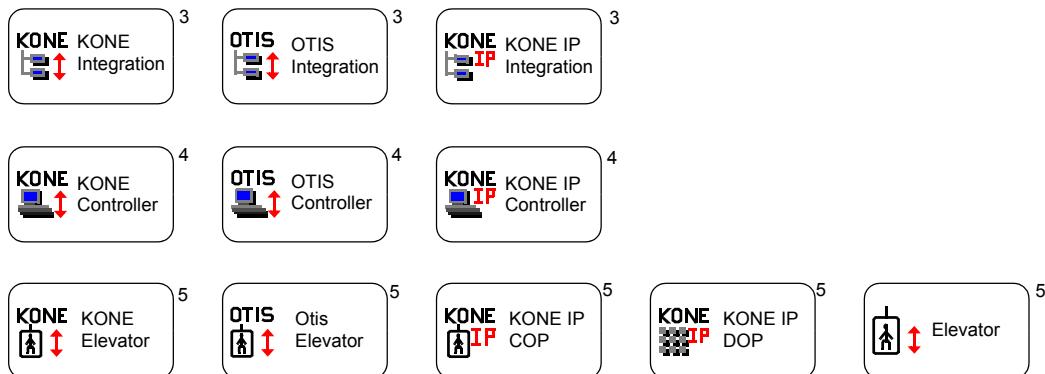
Intrusion detection objects:



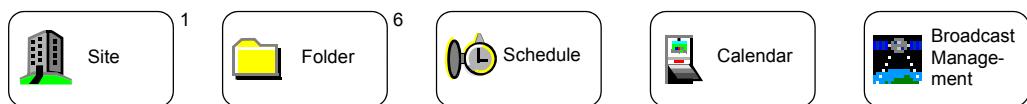
Custom logic objects:



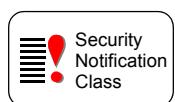
Elevator-related objects:



Miscellaneous objects:



Internal and diagnostic object:



Notes: ¹ Site, ² Supervisory Device, ³ Integration, ⁴ Field Device, ⁵ Field Point, ⁶ Folder

CK722 Device Object

The CK722 Device Object defines the attributes that represent the externally visible characteristics of the CK722 controller.

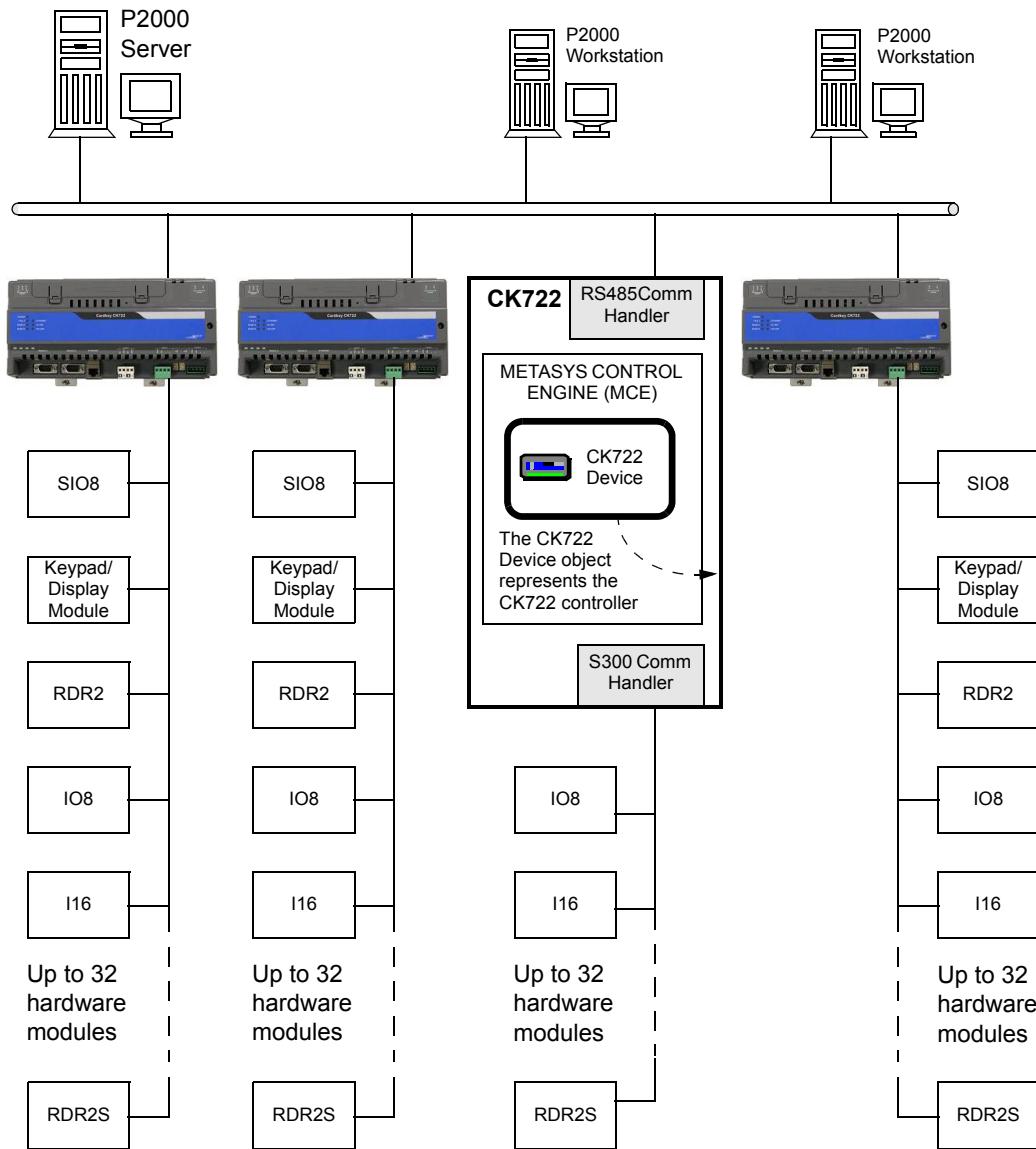


Figure 3-6: CK722 Device Object

For detailed information refer to the *CK722 Device Object* manual.

S300 Trunk Object

The S300 Trunk object is an S300 integration object that combines all bus-wide settings and diagnostic information.

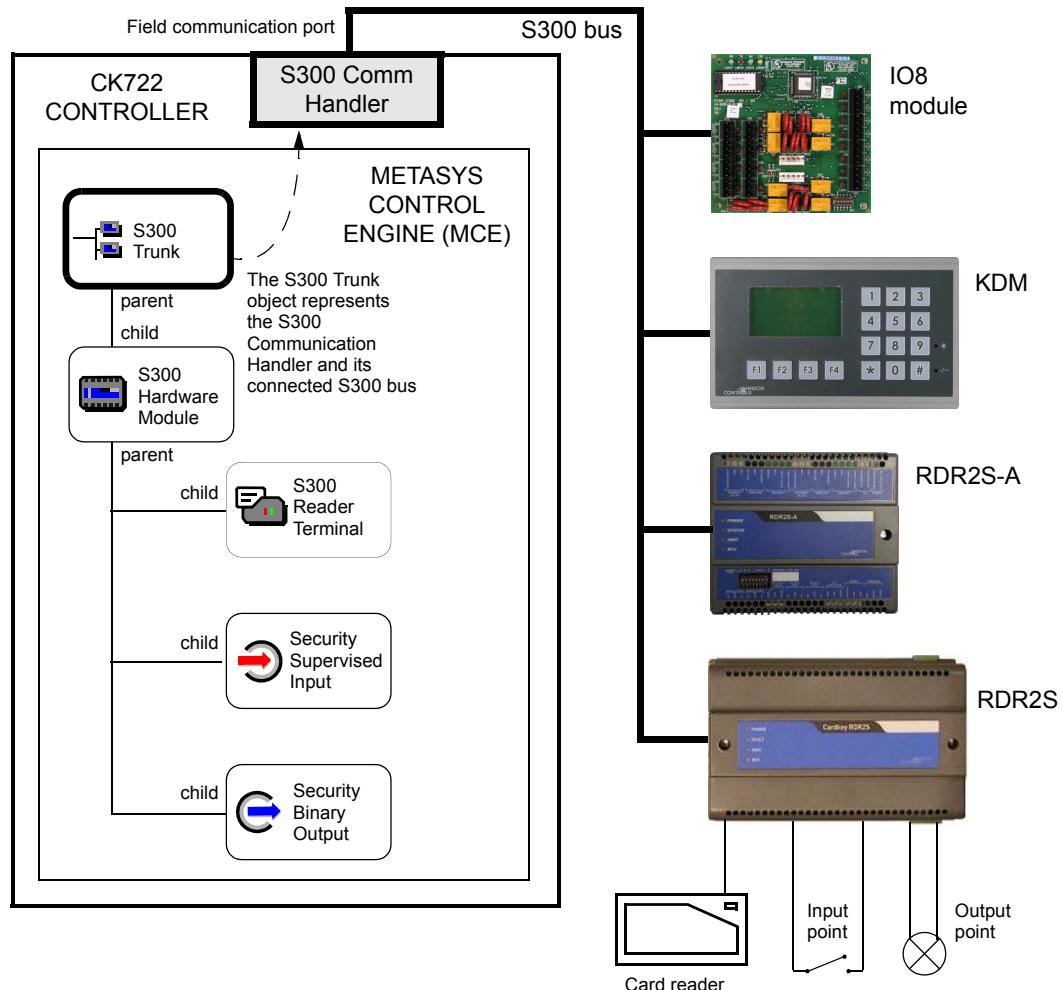


Figure 3-7: S300 Trunk Object

For detailed information refer to the *S300 Trunk Object* manual.

S300 Hardware Module Object

The S300 Hardware Module object combines all S300 hardware module-wide settings and diagnostic information. Each separate piece of hardware other than the supervisory controller on the S300 bus is represented by one S300 Hardware Module object.

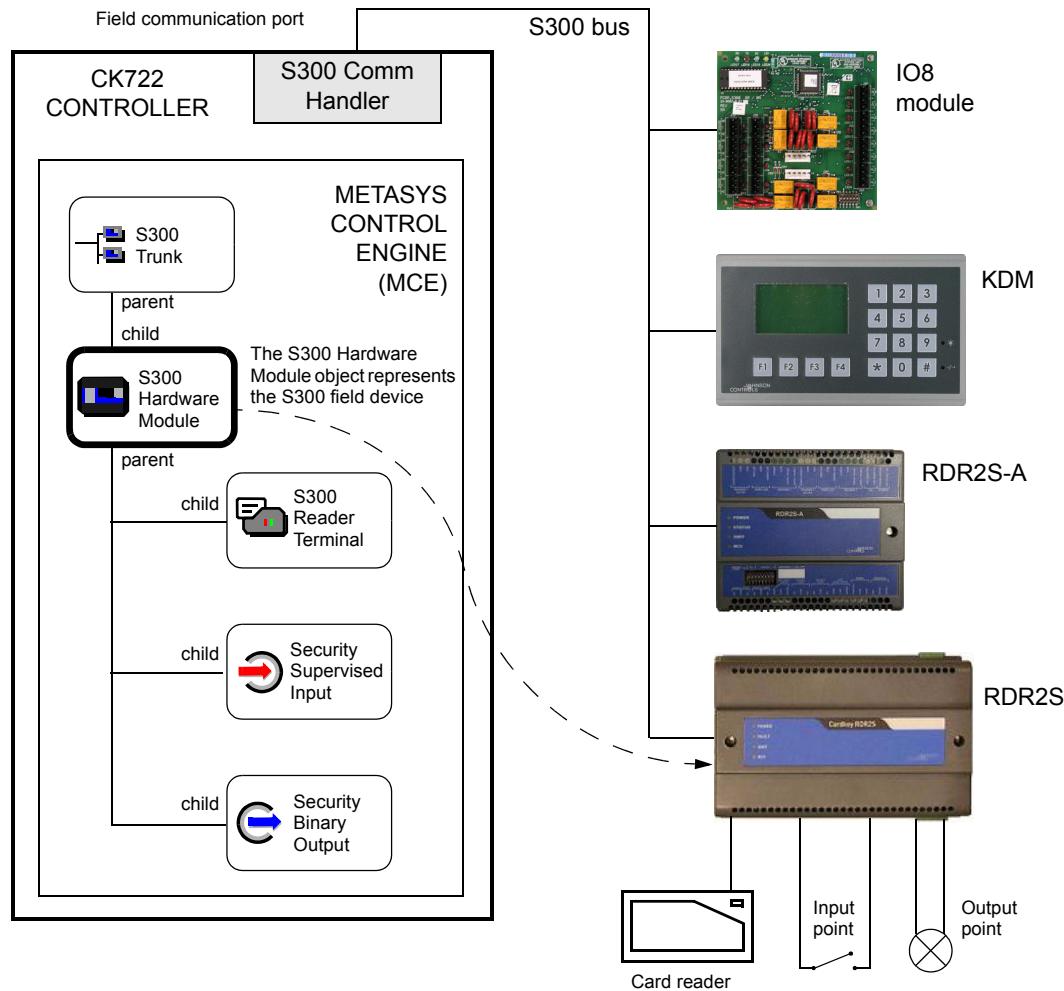


Figure 3-8: S300 Hardware Module Object

For detailed information refer to the *S300 Hardware Module Object* manual.

S300 Reader Terminal Object

The S300 Reader Terminal object has the following functions:

- Forwards the card and keypad information received from the S300 task to the Access Control object.
- Provides portal contact and auxiliary input information to the Door Sequence object.
- Carries out the access decisions handed down from the Door Sequence object in accordance with the capabilities of the S300 hardware module.

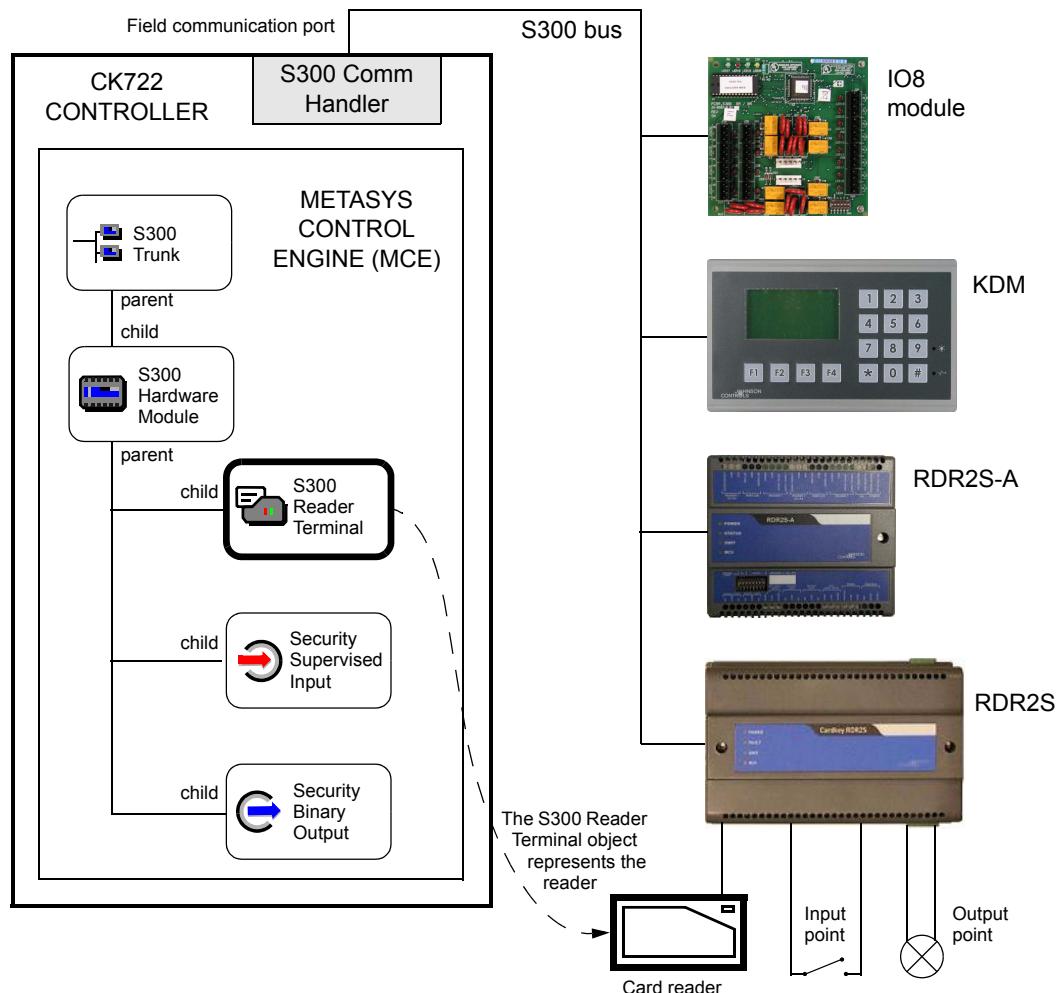


Figure 3-9: S300 Reader Terminal Object

For detailed information refer to the *S300 Reader Terminal Object* manual.

Security Supervised Input Object

The Security Supervised Input object defines Johnson Controls' mechanism to monitor the state of a physical supervised input point.

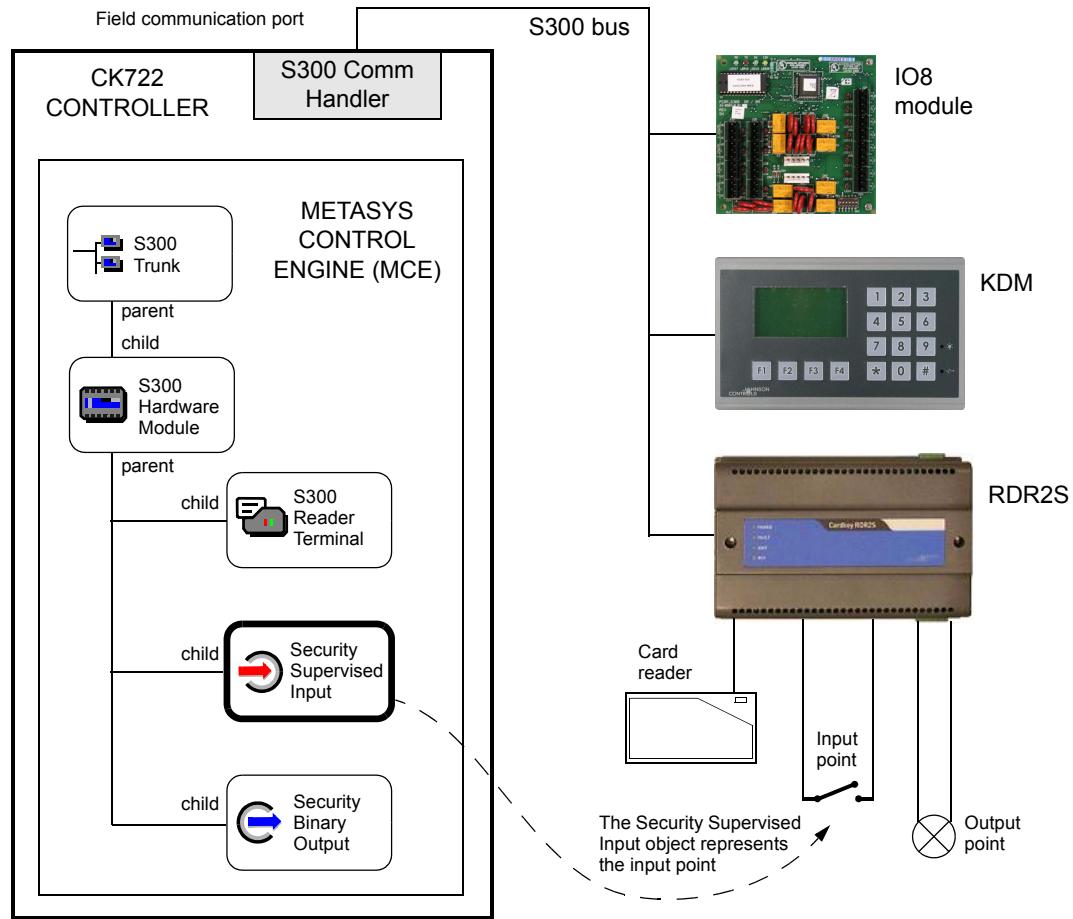


Figure 3-10: Security Supervised Input Object

For detailed information refer to the *Security Supervised Input Object* manual.

Security Binary Output Object

The Security Binary Output object defines Johnson Controls mechanism to control the state of a physical two-state output point for security applications.

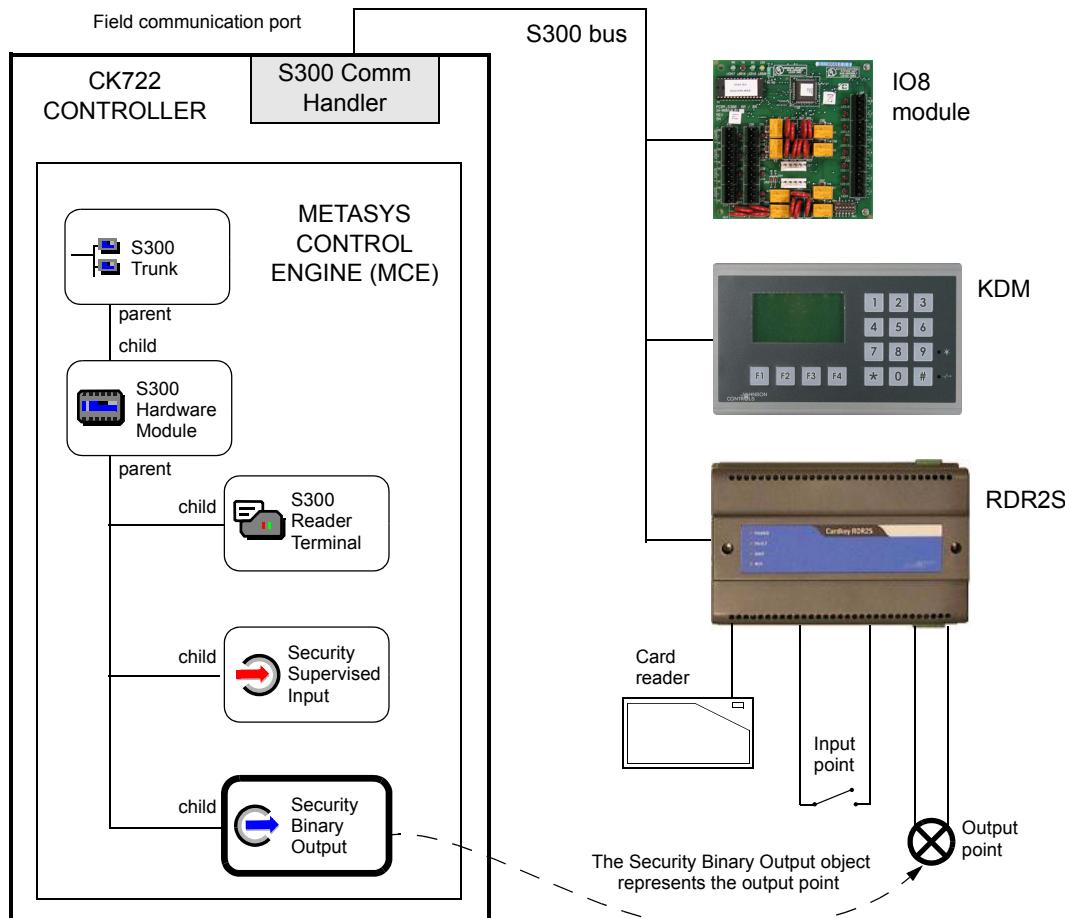


Figure 3-11: Security Binary Output Object

For detailed information refer to the *Security Binary Output Object* manual.

Access Control Object

The Access Control object determines whether an access request made at an access controller is granted or denied.

The access decision of the Access Control object is the first, but not the ultimate step in determining whether or not any physical action is taken, such as unlocking the door strike.

Based on the application, the Access Control object interacts with a variety of different objects. Figure 3-12 illustrates basic interactions.

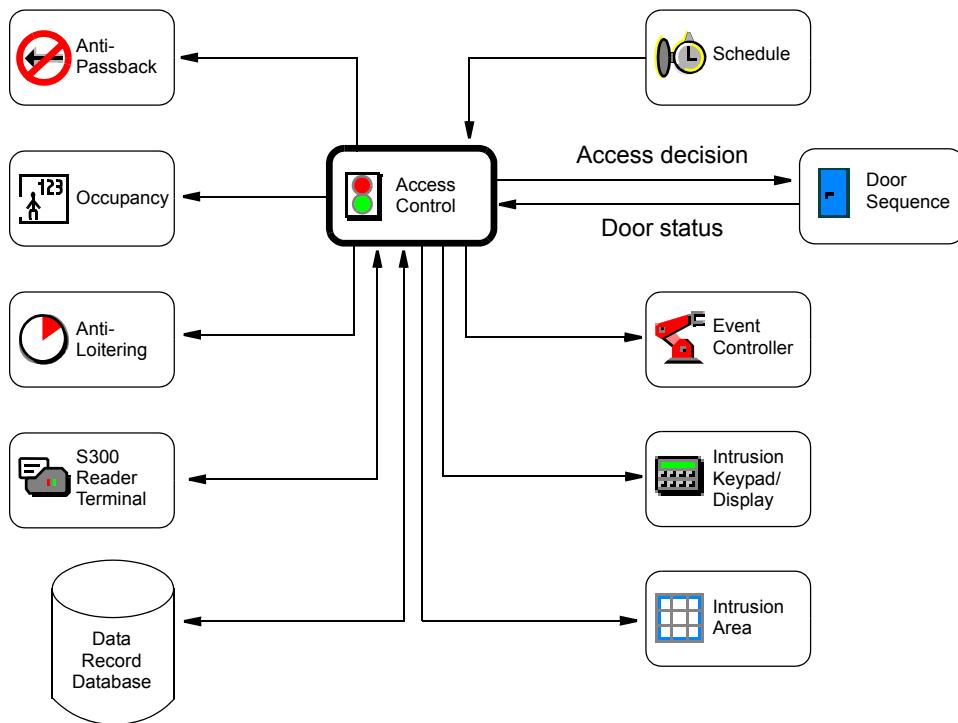


Figure 3-12: Access Control Object

For detailed information refer to the *Access Control Object* manual.

Door Sequence Object

The Door Sequence object provides a mechanism to operate the basic portal-related inputs and outputs.

Depending on the application, the Door Sequence object interacts with a variety of different objects. Figure 3-13 illustrates basic interactions.

When the Access Control object makes an access decision, it writes to the *Decision Category* attribute of the Door Sequence object. The Door Sequence object then takes the appropriate actions to operate any outputs, and possibly generate notifications.

Also, the Door Sequence object can activate Controller Event objects when it is about to unlock a portal, or after the portal is determined to be locked and closed.

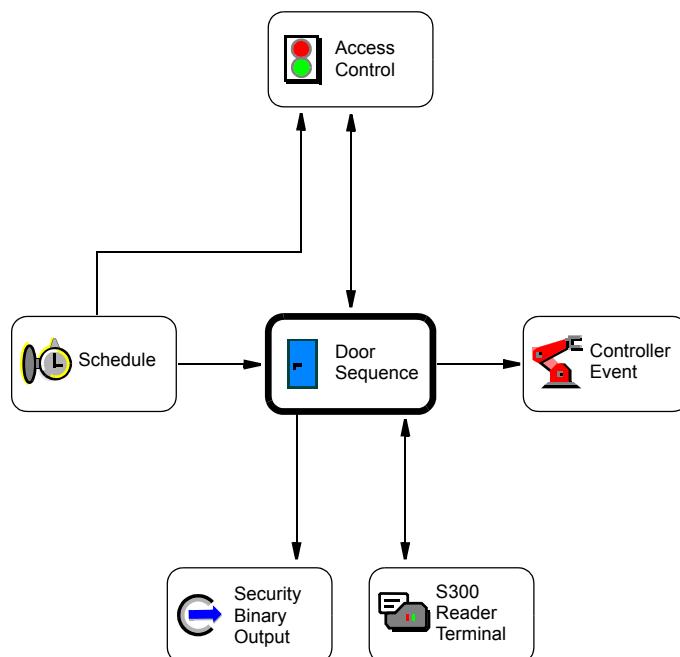


Figure 3-13: Door Sequence Object

For detailed information refer to the *Door Sequence Object* manual.

Anti-Passback Object

The Anti-Passback object is used in monitoring and enforcing the use of entry and exit readers in accordance with the anti-passback rule by time, and the anti-passback rule by location (also known as the entry-exit rule).

Violations of any of the anti-passback rules can be reported to the P2000 server and may cause access to be denied for the entity who is violating the rules.

The anti-passback feature can also be used locally (without the P2000 server).

An anti-passback application managing a single anti-passback area consists of one or more Anti-Passback objects cooperating with one or more Access Control objects.

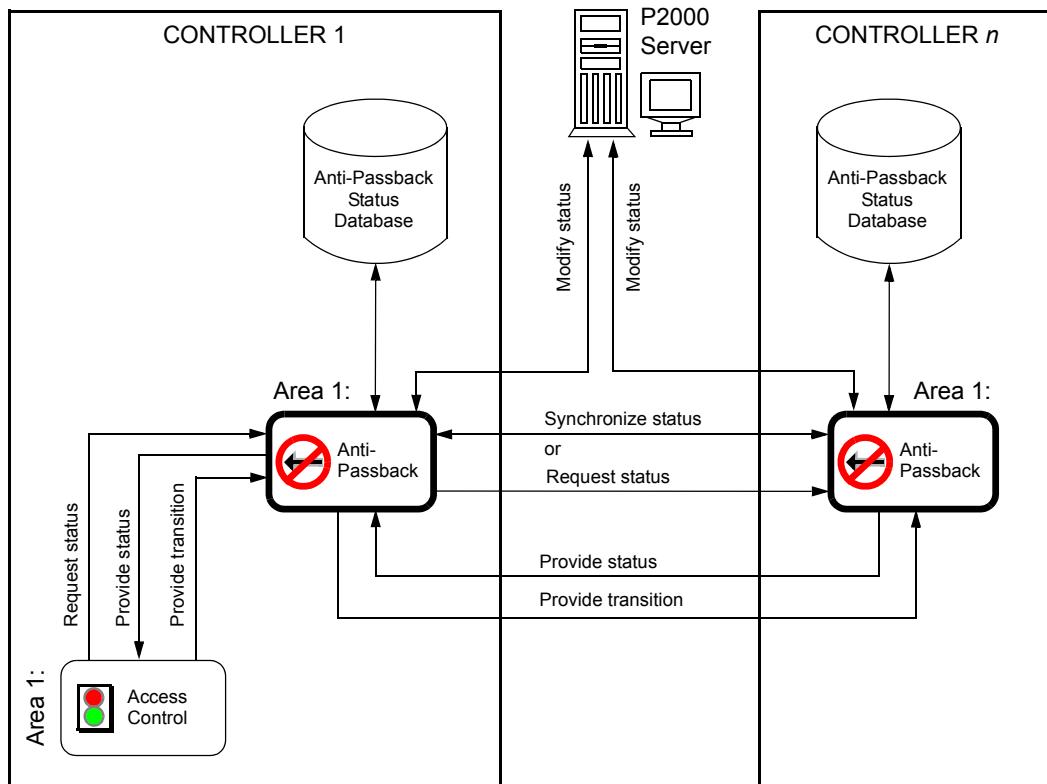


Figure 3-14: Anti-Passback Object

For detailed information refer to the *Anti-Passback Object* manual.

Occupancy Object

The Occupancy object monitors the number of entities in an occupancy space.

The Occupancy object can work in an anonymous mode, in which occupancy is determined by balancing anonymous in versus out transitions, or it can determine the number of occupants by keeping track of all occupants by their entity ID.

The Occupancy object interacts with the Access Control object in the following way: An Access Control object consults its local Occupancy object, which immediately returns an occupancy decision. The Access Control objects also keep their local Occupancy objects informed about any transitions into or out of the occupancy space.

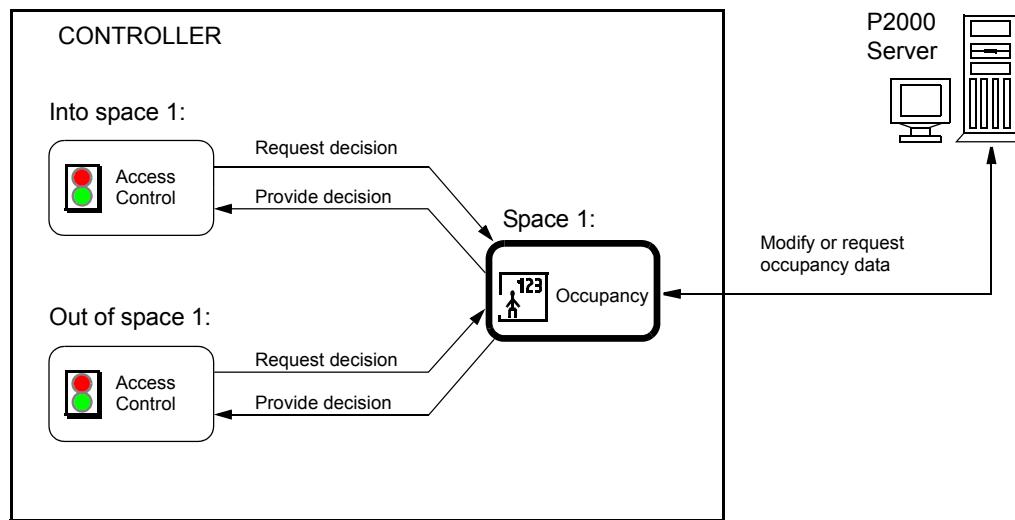


Figure 3-15: Occupancy Object

For detailed information refer to the *Occupancy Object* manual.

Anti-Loitering Object

The Anti-Loitering object tracks the date and time of transition of up to 1000 different entities into a specific area.

The Anti-Loitering object is informed about any transitions into or out of the area by the Access Control objects.

The use of the anti-loitering feature is to monitor the time individual entities spend in an anti-loitering area. If an entity exceeds the area's anti-loitering time, an anti-loitering notification is generated.

An anti-loitering area may be accessible through several portals, and may span controllers. Also, a single controller may hold several anti-loitering areas.

The following diagram shows the major blocks the Anti-Loitering object interacts with.

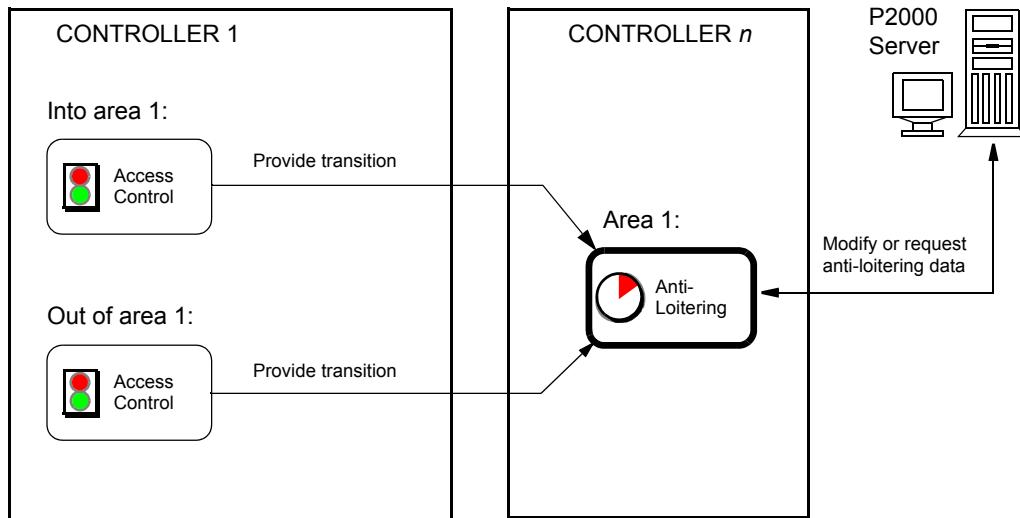


Figure 3-16: Anti-Loitering Object

For detailed information refer to the *Anti-Loitering Object* manual.

Intrusion Area Object

The Intrusion Area object arms and disarms its associated Intrusion Zone objects. The Intrusion Area object itself may be armed or disarmed automatically by an associated Schedule object or by authorized users via the Intrusion Keypad/Display module.

An Intrusion Area object can contain other Intrusion Area objects.

The user may perform the following via the host or via the Intrusion Keypad/Display object:

- Arm the Intrusion Area object
- Disarm the Intrusion Area object
- View the status of the Intrusion Area object

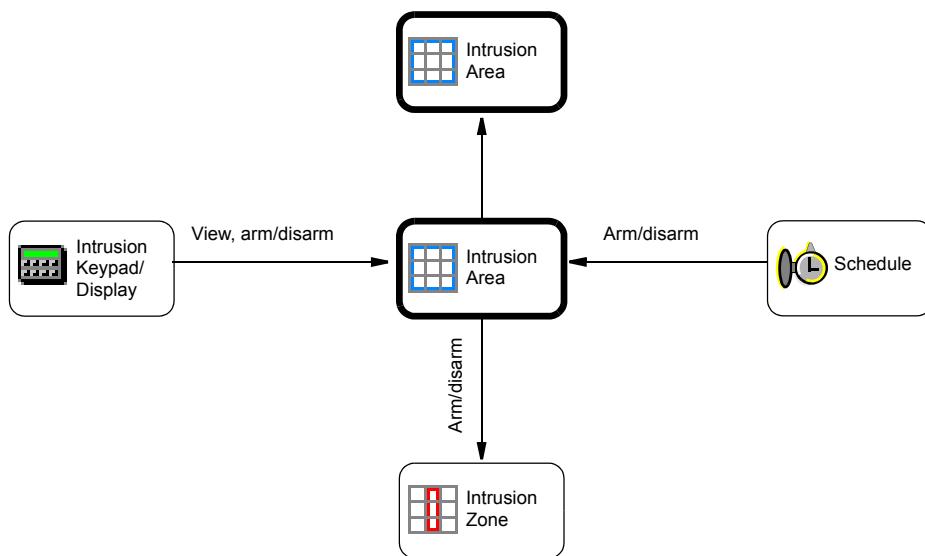


Figure 3-17: Intrusion Area Object Details

See also “Intrusion Detection System” on page 3-23 for a depiction on how the Intrusion objects interact with each other.

For detailed information refer to the *Intrusion Area Object* manual.

Intrusion Zone Object

The Intrusion Zone object monitors and controls a group of sensors. It receives alarm states from its associated Security Supervised Input objects and sets an associated Security Binary Output object. It also generates notifications when alarm conditions occur.

The Intrusion Zone object may be armed or disarmed by the Intrusion Area object.

The user may perform the following via the host or via the Intrusion Keypad/Display object:

- Bypass the Intrusion Zone object
- Activate the Intrusion Zone object
- Acknowledge the Intrusion Zone object
- View the status of the Intrusion Zone object

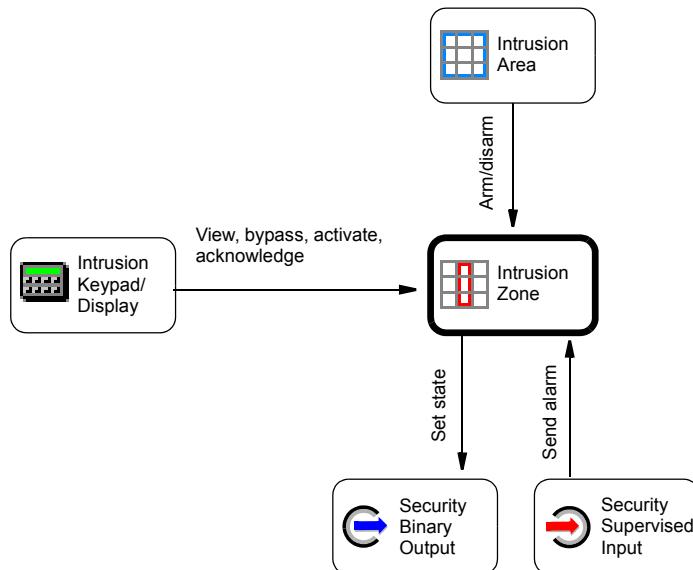


Figure 3-18: Intrusion Zone Object Details

See also “Intrusion Detection System” on page 3-23 for a depiction on how the Intrusion objects interact with each other.

For detailed information refer to the *Intrusion Zone Object* manual.

Intrusion Announcer Object

The Intrusion Announcer object resets associated output points. It may be silenced by authorized users via the Keypad/Display module or by users via the host (P2000 server).

The Intrusion Announcer object may generate notifications when a change occurs in output point status or when an error occurs while writing to other objects.

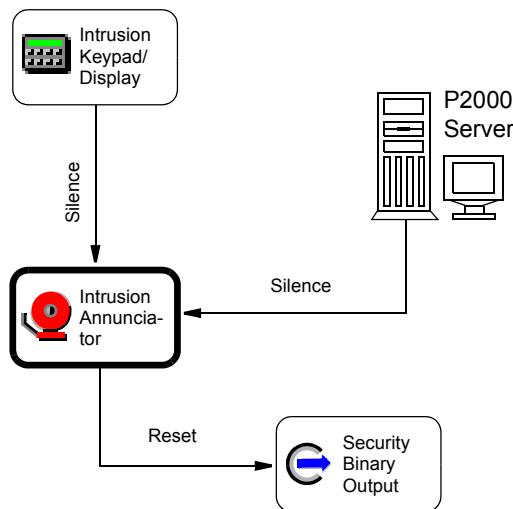


Figure 3-19: Intrusion Announcer Object Details

See also “Intrusion Detection System” on page 3-23 for a depiction on how the Intrusion objects interact with each other.

For detailed information refer to the *Intrusion Announcer Object* manual.

Intrusion Keypad/Display Object

The Intrusion Keypad/Display object interfaces to the Intrusion Keypad/Display module and allows authorized users to control the Intrusion Area, Intrusion Zone, and Intrusion Annunciator objects.

The Intrusion Keypad/Display object may generate notifications when an error occurs while writing to other objects.

Use the Intrusion Keypad/Display object to perform the following functions on other objects:

- The Intrusion Area object: view status, arm, and disarm
- The Intrusion Annunciator object: view status, silence
- The Intrusion Zone object: view status, bypass, acknowledge alarms, and activate

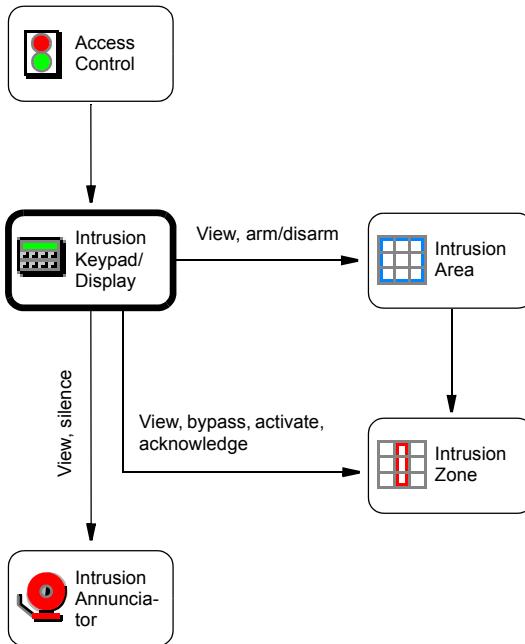


Figure 3-20: Keypad/Display Object Details

See also “Intrusion Detection System” on page 3-23 for a depiction on how the Intrusion objects interact with each other.

For information on the operation of S300-KDM refer to “S300-KDM” on page 2-14.

For detailed information on the object refer to the *Intrusion Keypad/Display Object* manual.

Intrusion Detection System

The Intrusion Area, Intrusion Zone, Intrusion Annunciator, and Intrusion Keypad/Display objects define Johnson Controls' mechanism of providing intrusion detection capability.

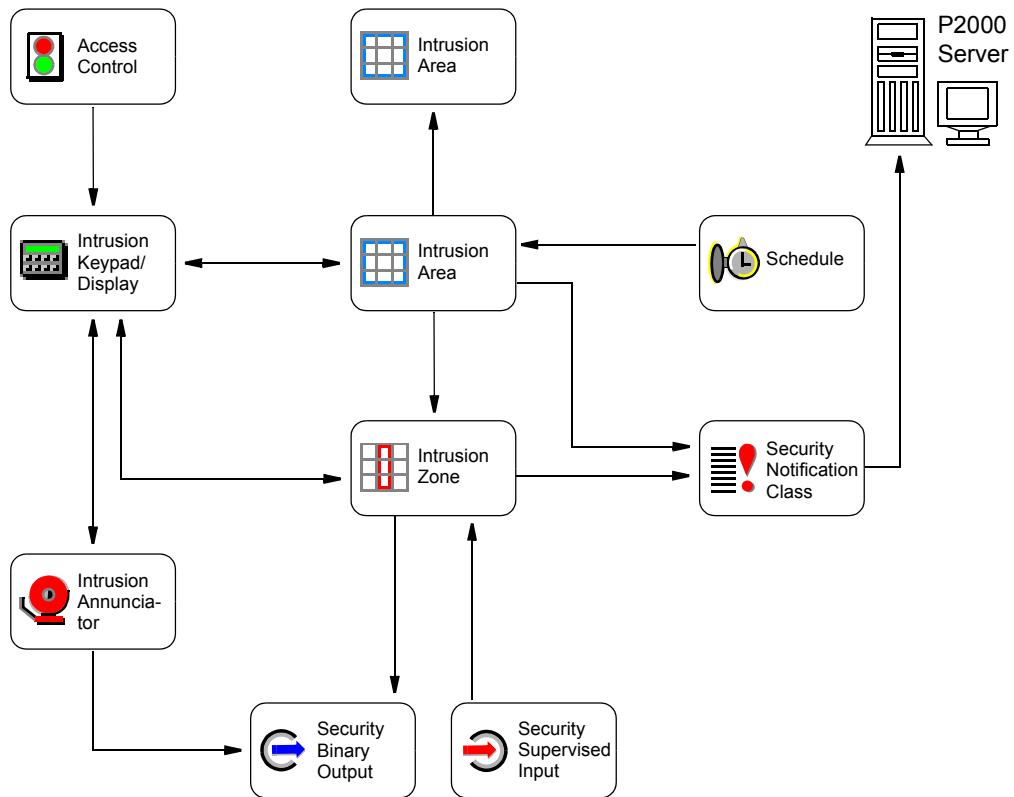


Figure 3-21: Intrusion Detection System: Object Interactions

Interlock Object

The Interlock object provides a means to establish conditional control over one or more other objects. It consists of an IF conditional statement, True command statements, and False command statements. Through these statements, the user specifies a set of conditional checks (using one or more points) for which a series of commands is used to control a collection of one or more other objects.

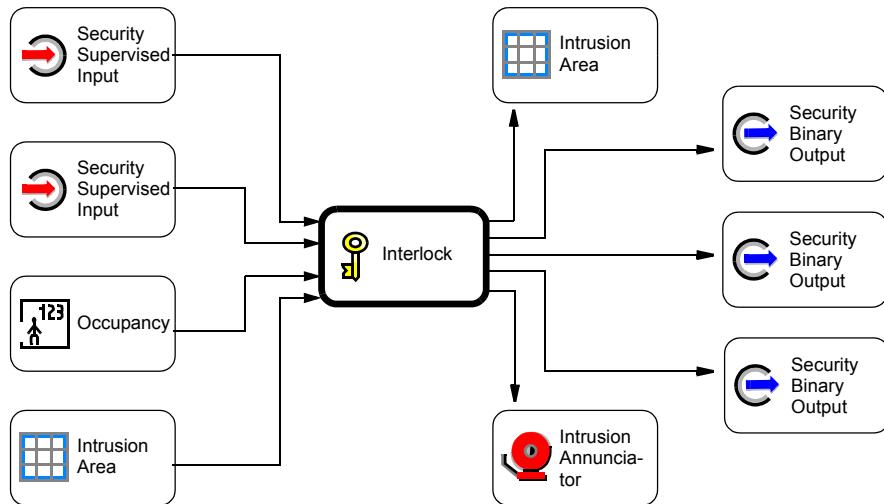


Figure 3-22: Interlock Object

For detailed information refer to the *Interlock Object* manual.

Multiple Command Object

The Multiple Command object issues a series of commands to multiple objects by means of single command actions.

The objects assigned to the Multiple Command object are called *slaves*. Whenever the Multiple Command object is set to a new state, the commands associated with the state will be sent to the slave objects after any specified delays.

For each of the slaves a command and delay can be defined for each state. An object does not need to be sent a command for every state. On the other hand, it may be sent more than one command per state by repeating it as a slave object and specifying another command.

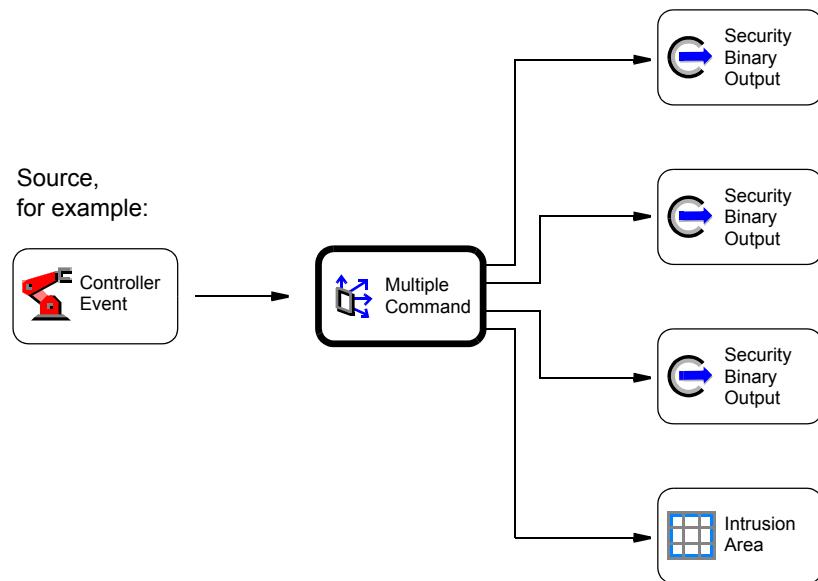


Figure 3-23: Multiple Command Object

For detailed information refer to the *Multiple Command Object* manual.

Controller Event Object

The Controller Event object defines the conditions upon which a controller event is triggered, and the actions that are taken when the controller event is activated or deactivated. The Controller Event Object is also responsible for reporting the activation or deactivation of the controller event to the host (P2000 server).

Controller events can either write a fixed value or a value read from another object into the target object, or they can toggle the current target attribute's value.

The following diagram shows the major blocks the Controller Event Object interacts with.

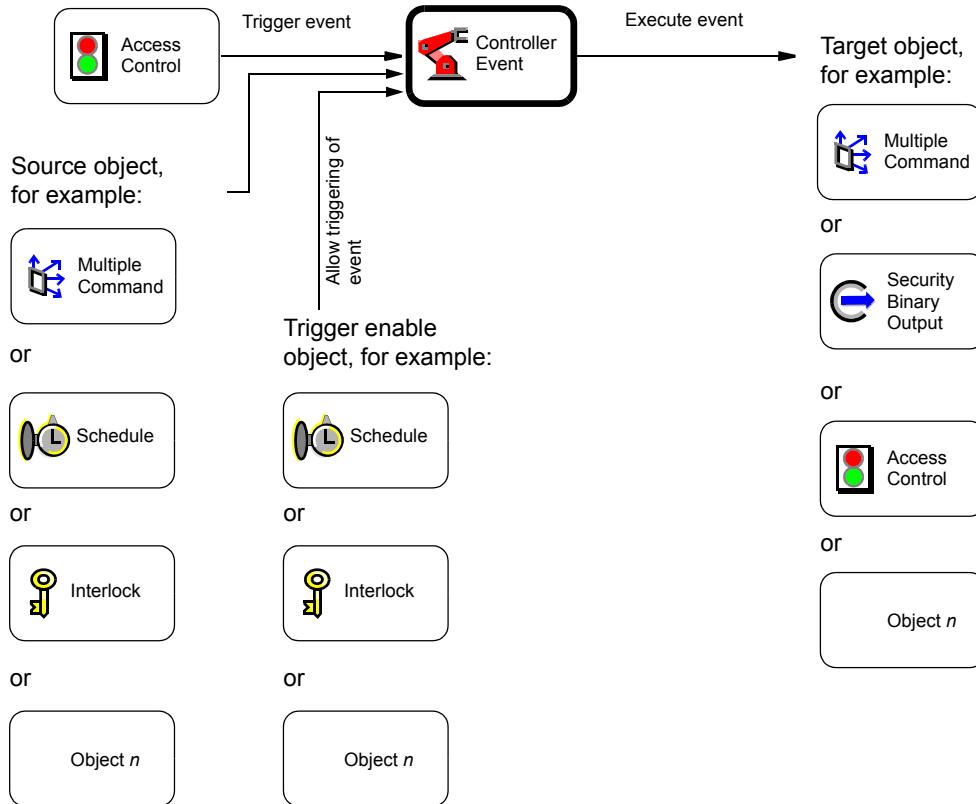


Figure 3-24: Controller Event Object

For detailed information refer to the *Controller Event Object* manual.

KONE Integration

The KONE Integration object is used to configure KONE integration-wide settings. It also provides KONE integration-wide communication statistics.

The KONE Integration object supports the KONE HLI V2.3 serial protocol.

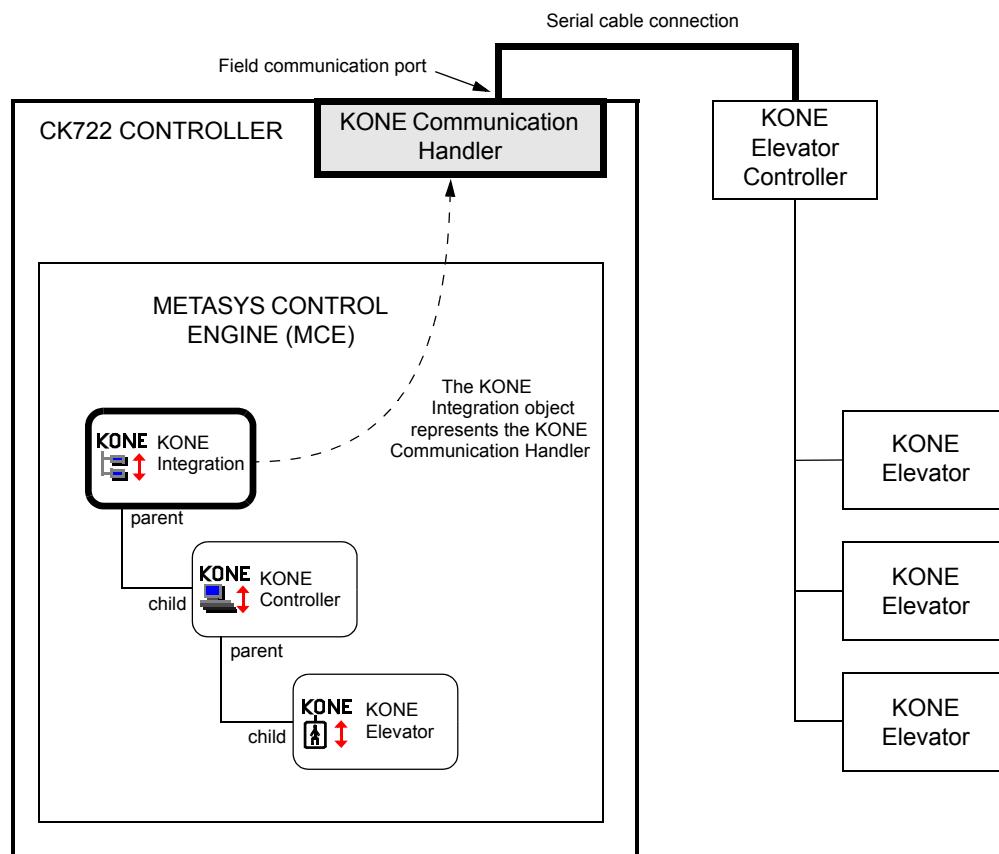


Figure 3-25: KONE Integration Object

For detailed information refer to the *KONE Integration Object* manual.

Otis Integration

The Otis Integration object provides integration-wide communication statistics.

The Otis Integration object supports the E.M.S.- Security / B.M.S.serial protocol, documented in *OTIS - Ref No. 51646B*.

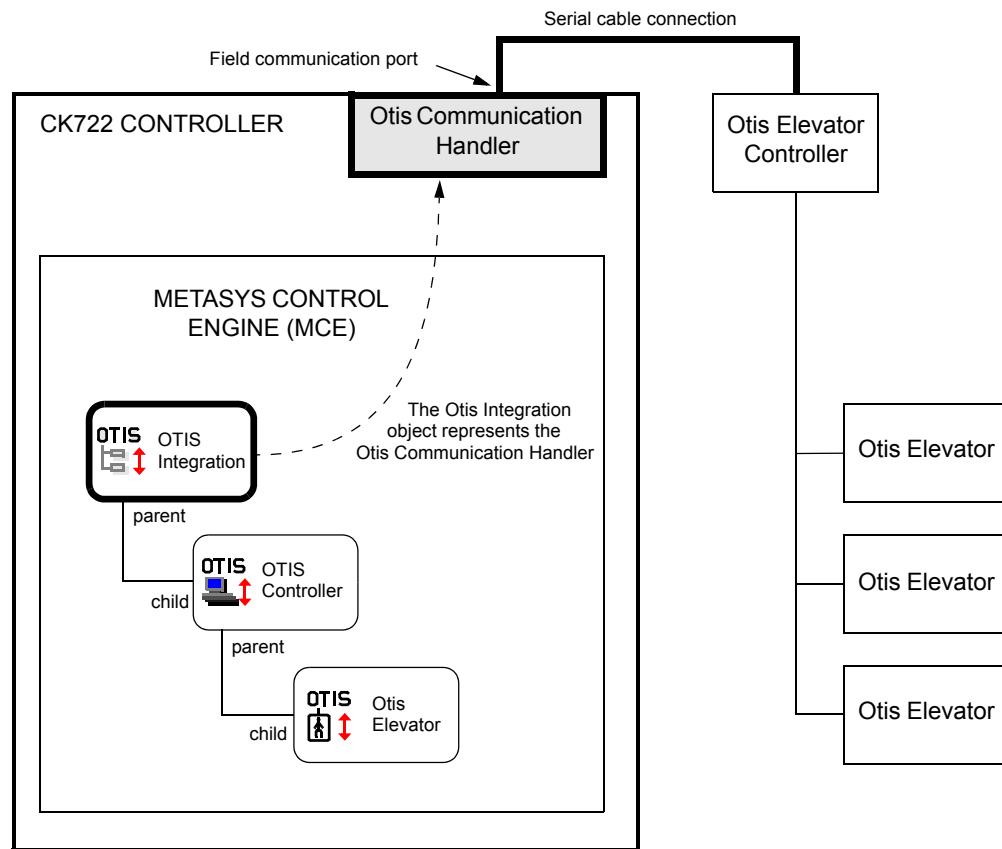


Figure 3-26: Otis Integration Object

For detailed information refer to the *Otis Integration Object* manual.

KONE IP Integration

The KONE IP Integration object is used to configure KONE IP integration-wide settings, and provides KONE IP integration-wide communication statistics.

The network protocol supported by the KONE IP Integration object is documented in the *KONE Group Controller Access Control Interface Specification Rev 1.5*.

There are different rules when interfacing to a KONE KIC as opposed to a Primary or Back-up PC Group Controller. KONE KIC controllers do not support Destination Operation Panels (DOPs).

Also, KONE KIC controllers may control elevator groups with different layouts of KONE levels. As the mapping of P2000 floor numbers to KONE levels is done inside the KONE-IP Integration object, it may be necessary to use several supervisory controllers to communicate with the same KIC.

A single KONE-IP Integration object can be used for all groups that map the same P2000 floor to KONE level and vice versa. Groups are allowed to cover a subset of that mapping scheme.

To allow for a greater number of groups to be handled by a single supervisory controller, some attribute values of the KONE-IP Integration object are ignored when the KONE-IP Controller object's *Type* attribute is set to KIC.

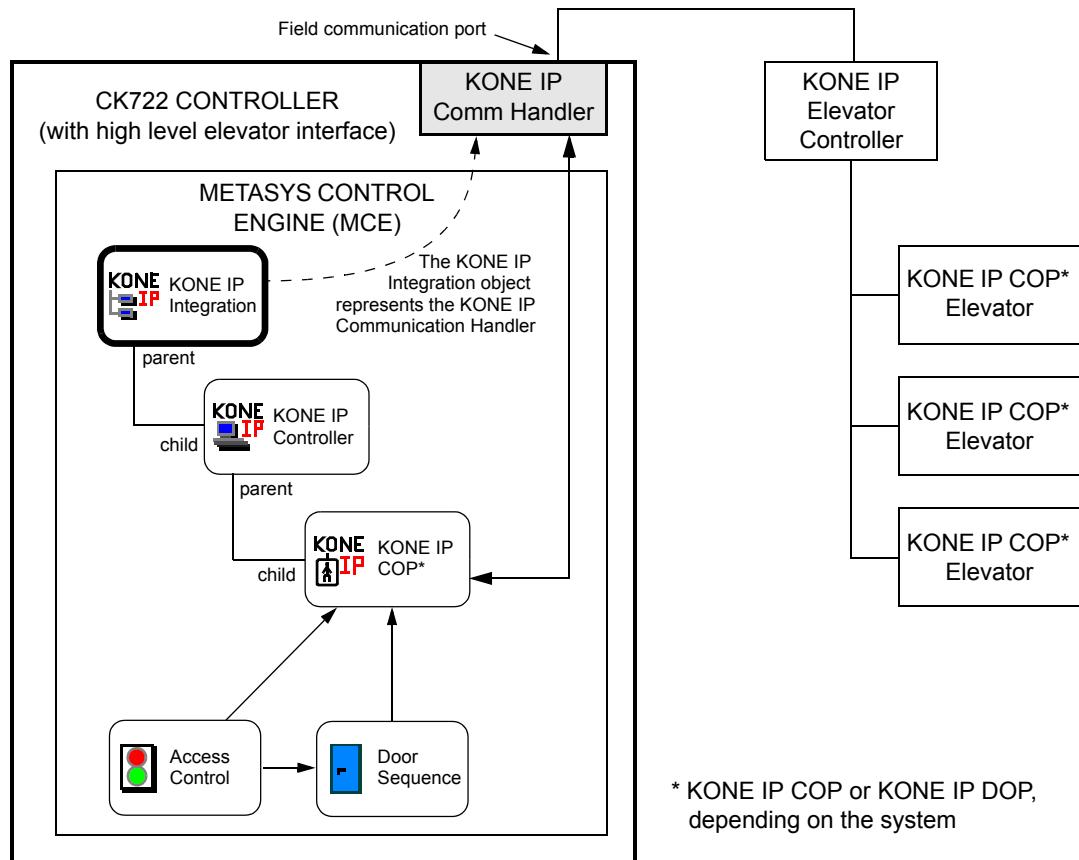


Figure 3-27: KONE IP Integration Object

For detailed information refer to the *KONE IP Integration Object* manual.

KONE Controller

The KONE Controller object represents the KONE elevator controller inside the Metasys Control Engine (MCE). The object serves as the interface to set configuration parameters related to the elevator controller, as well as the interface to monitor the status of the elevator controller and its communication with the CK722 supervisory controller.

The KONE Controller object must be a child of a KONE Integration object. Each KONE Controller object represents a single elevator controller that is connected to the CK722 supervisory controller. The limit is 1 elevator controller per CK722.

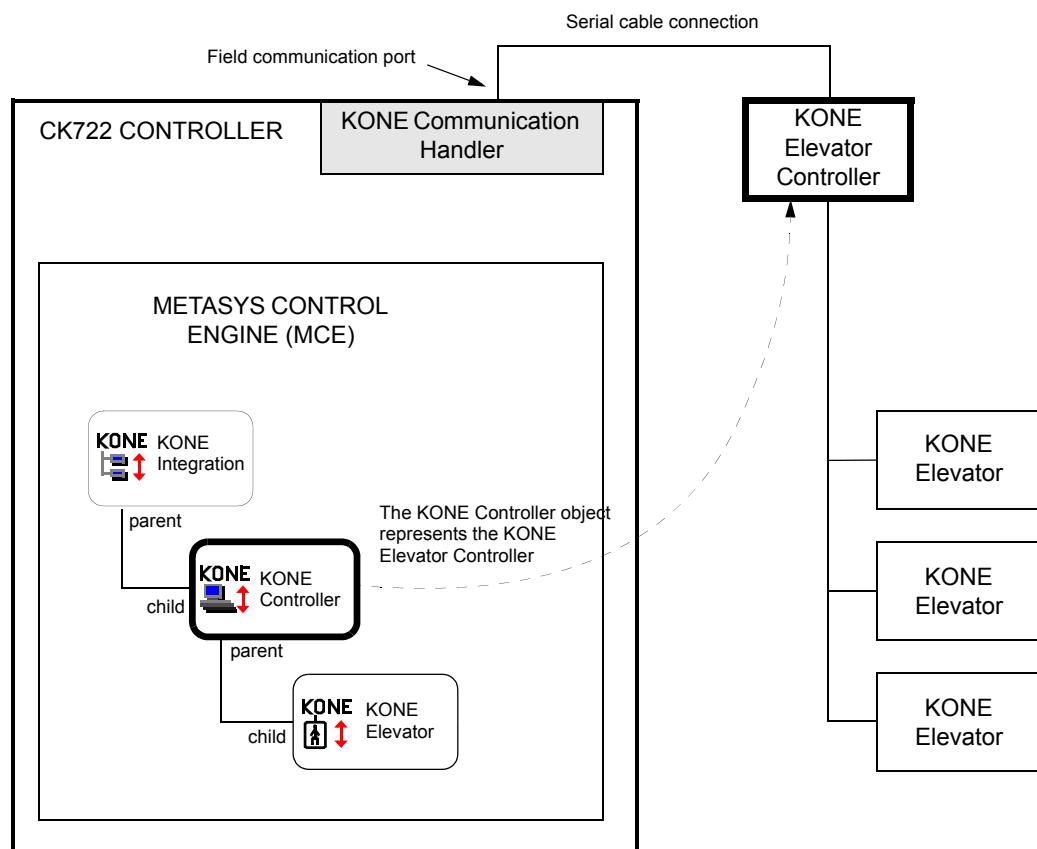


Figure 3-28: KONE Controller Object

For detailed information refer to the *KONE Controller Object* manual.

Otis Controller

The Otis Controller object represents the Otis elevator controller inside the Metasys Control Engine (MCE). The object serves as the interface to set configuration parameters related to the elevator controller, as well as the interface to monitor the status of the elevator controller and its communication with the CK722 supervisory controller.

The Otis Controller object is a child of the Otis Integration object. Each Otis Controller object represents a single elevator controller that is connected to the CK722 supervisory controller. The limit is 1 elevator controller per CK722.

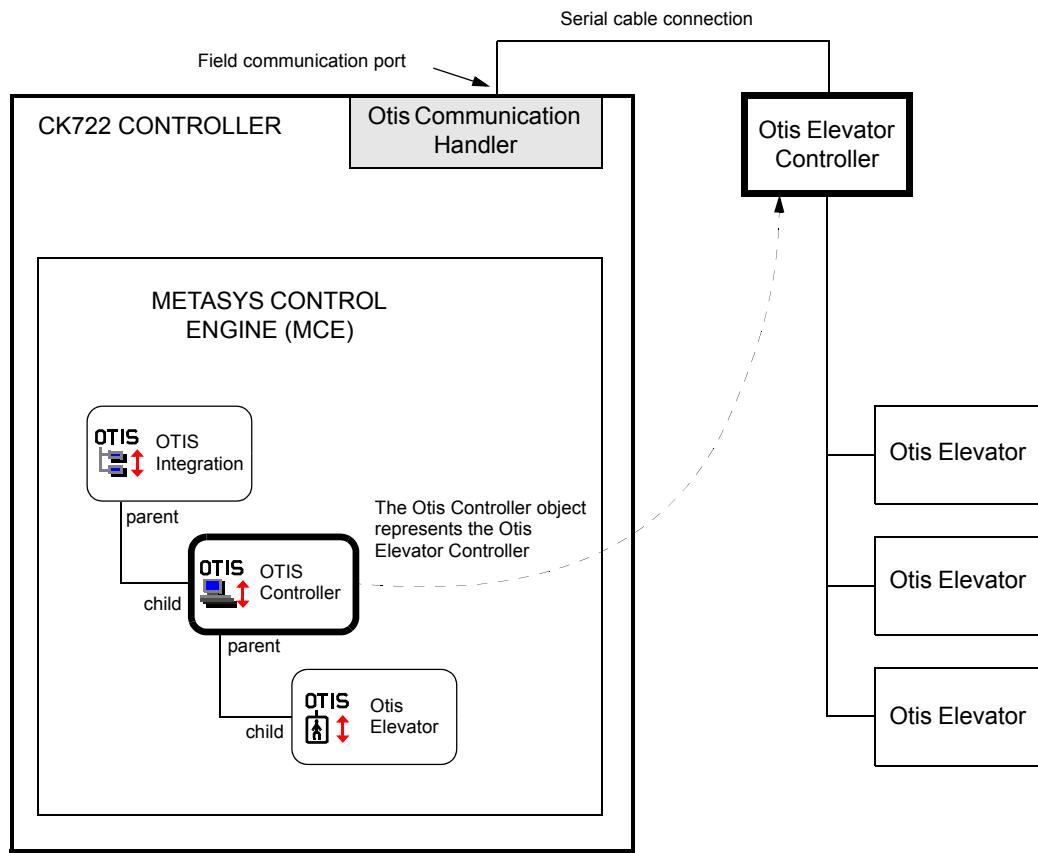


Figure 3-29: Otis Controller Object

For detailed information refer to the *Otis Controller Object* manual.

KONE IP Controller

The KONE IP Controller object represents a single controller (a KONE PC group controller or a KONE KIC).

The KONE IP Controller object serves as the interface to set the configuration parameters related to the elevator controller, as well as the interface to monitor the status of the elevator controller and its communication with the CK722 controller.

The KONE IP Controller object must be a child of a KONE IP Integration object.

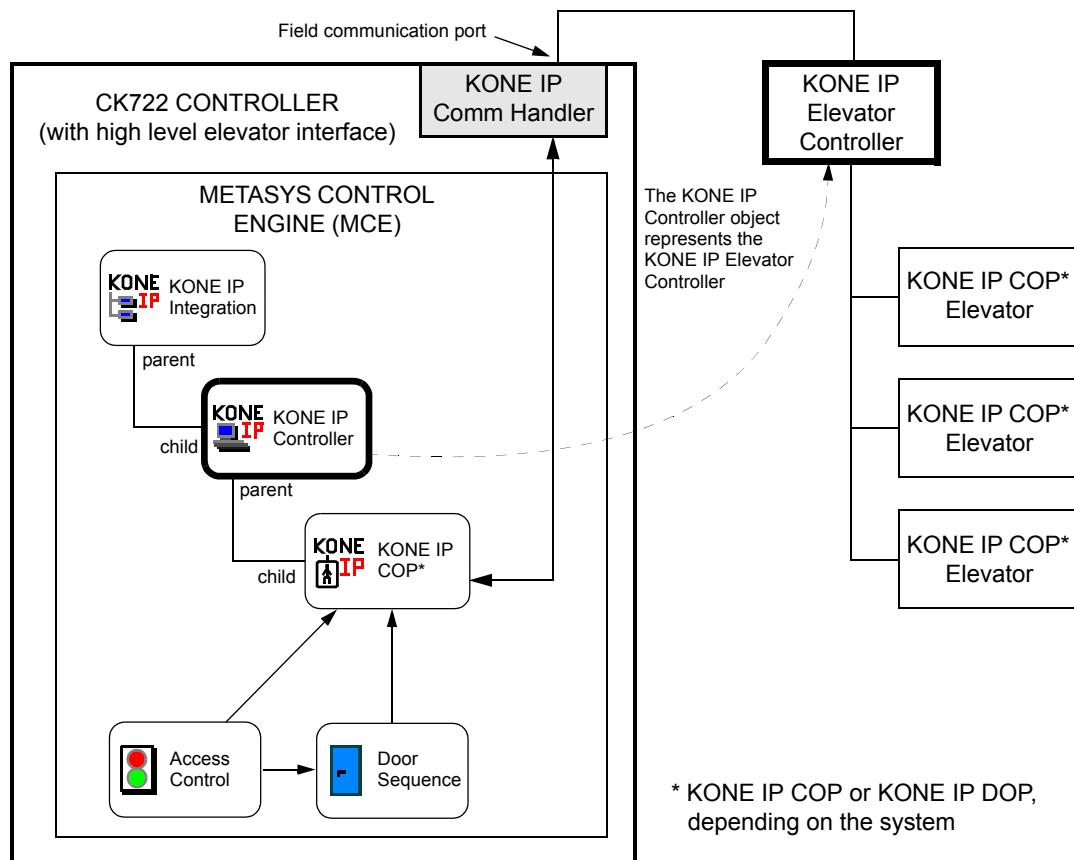


Figure 3-30: KONE IP Controller Object

For detailed information refer to the *KONE IP Controller Object* manual.

KONE Elevator Object

The KONE Elevator object represents KONE elevator in a KONE high level elevator integration. The KONE Elevator object must be a child of the KONE Controller object.

In a high level elevator integration, the access control system interfaces with the elevator control system through a communications protocol. Granting access to floors is achieved by sending telegrams to the elevator controller; reporting the pressed floor buttons is achieved by receiving telegrams from the elevator controller.

The arrows in the diagram show the message flow between the different components of the elevator interface.

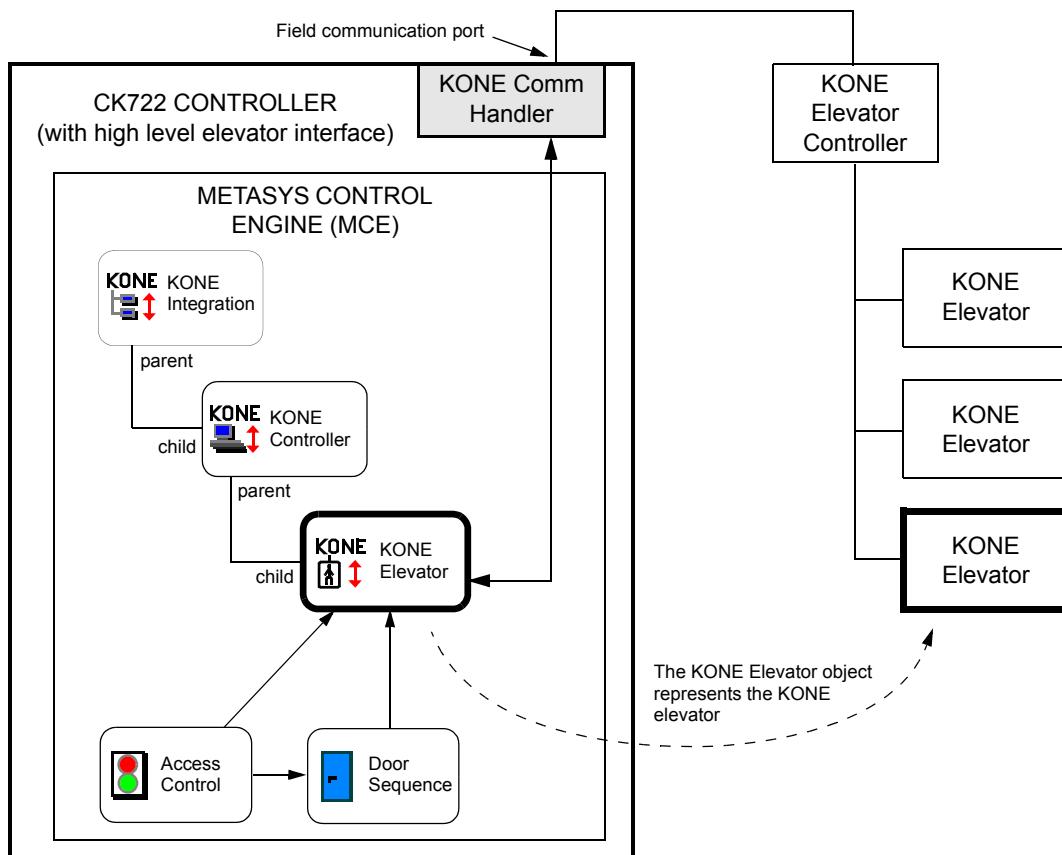


Figure 3-31: KONE Elevator Object

For detailed information refer to the *KONE Elevator Object* manual.

Otis Elevator Object

The Otis Elevator object represents Otis elevator in an Otis high level elevator integration. The Otis Elevator object must be a child of the Otis Controller object.

In a high level elevator integration, the access control system interfaces with the elevator control system through a communications protocol. Granting access to floors is achieved by sending telegrams to the elevator controller; reporting the pressed floor buttons is achieved by receiving telegrams from the elevator controller.

The arrows in the diagram show the message flow between the different components of the elevator interface.

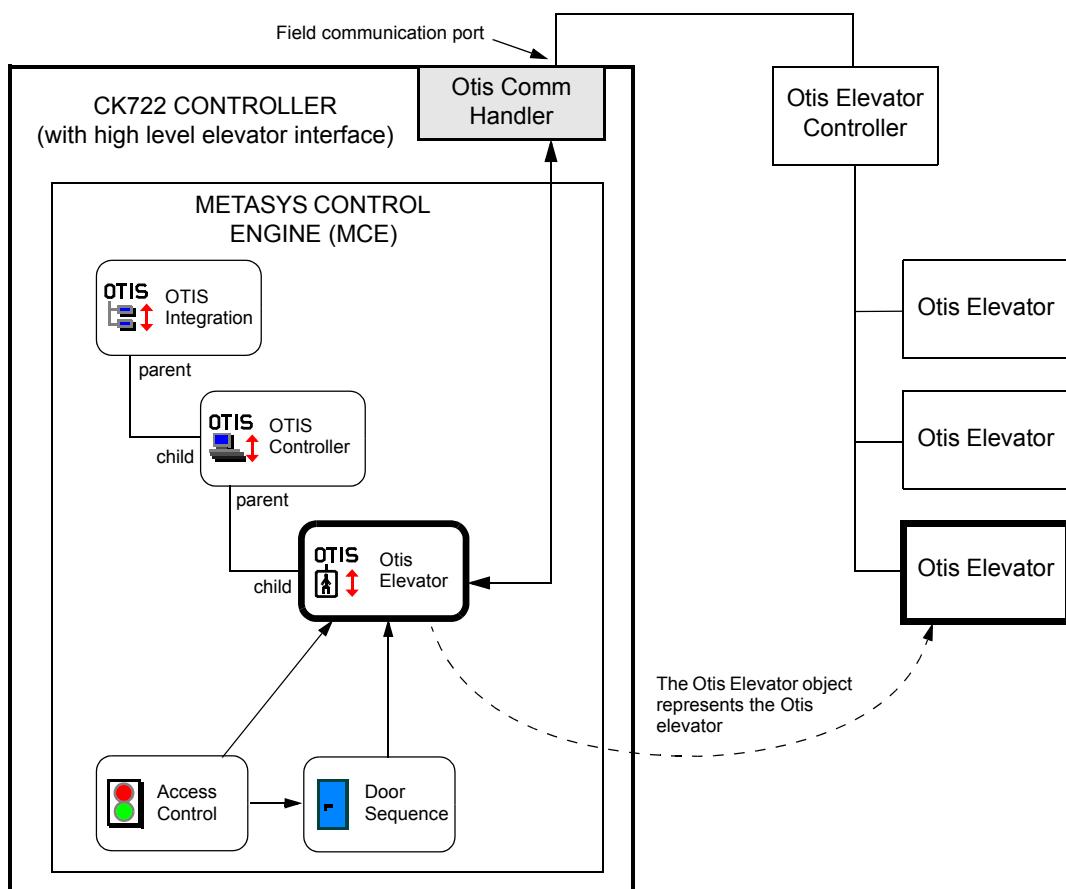


Figure 3-32: Otis Elevator

For detailed information refer to the *Otis Elevator Object* manual.

KONE IP COP Object

The KONE IP COP (Car Operation Panel) object represents KONE IP COP elevator in a KONE IP high level elevator integration. The KONE IP COP object must be a child of the KONE IP Controller object.

In a high level elevator integration, the access control system interfaces with the elevator control system through a communications protocol. Granting access to floors is achieved by sending telegrams to the elevator controller; reporting the pressed floor buttons is achieved by receiving telegrams from the elevator controller.

The arrows in the diagram show the message flow between the different components of the elevator interface.

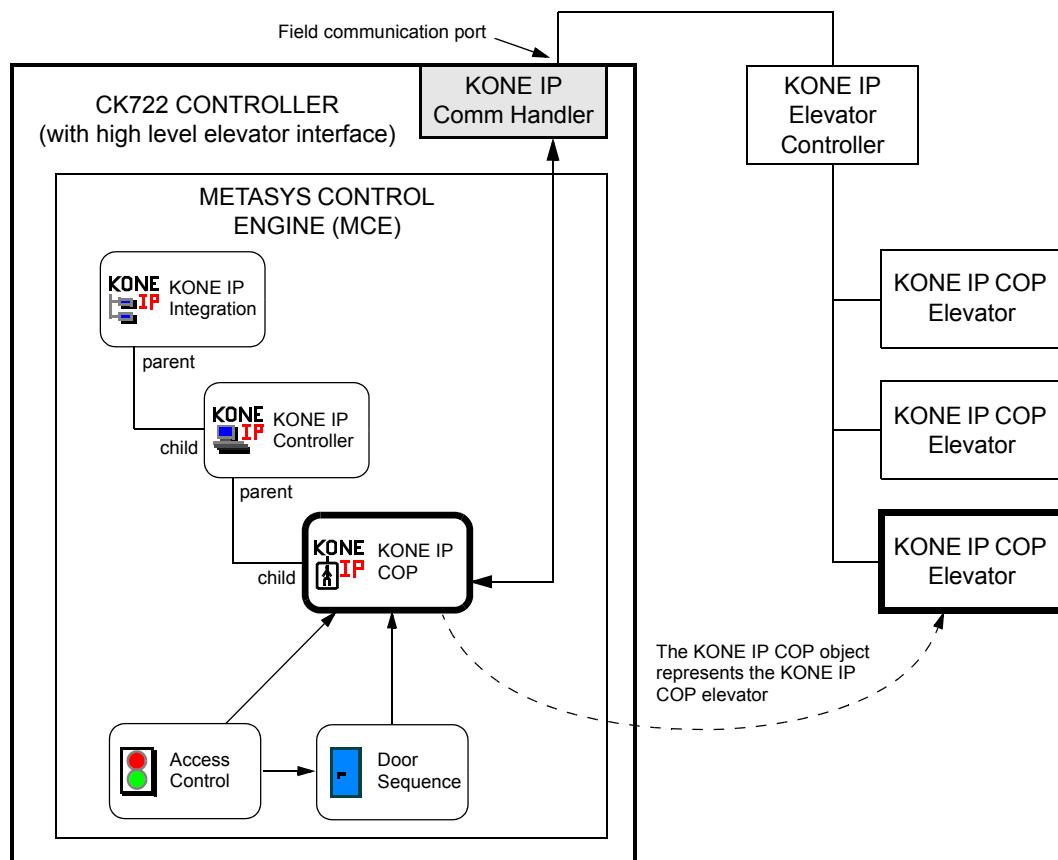


Figure 3-33: KONE IP COP Object

For detailed information refer to the *KONE IP COP Object* manual.

KONE IP DOP Object

The KONE IP DOP (Destination Operation Panel) object represents KONE IP DOP elevator in a KONE IP high level elevator integration. The KONE IP DOP object must be a child of the KONE IP Controller object.

In a high level elevator integration, the access control system interfaces with the elevator control system through a communications protocol. Granting access to floors is achieved by sending telegrams to the elevator controller; reporting the pressed floor buttons is achieved by receiving telegrams from the elevator controller.

The arrows in the diagram show the message flow between the different components of the elevator interface.

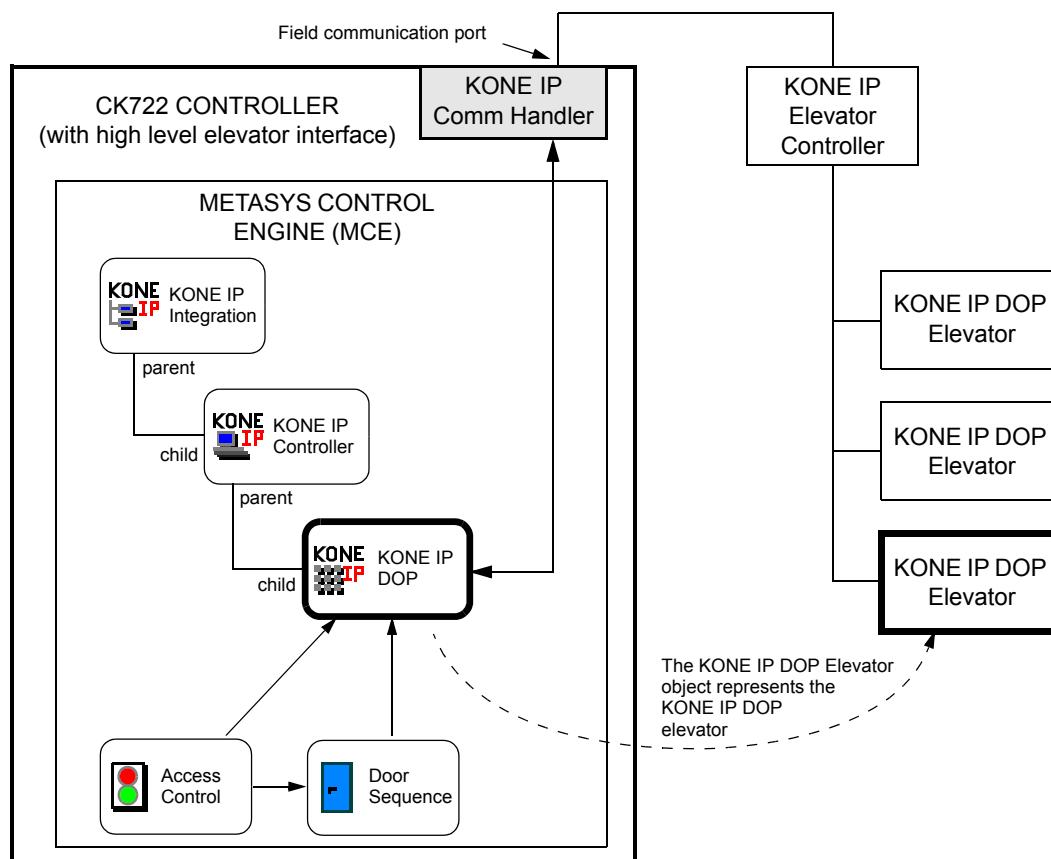


Figure 3-34: KONE IP DOP Object

For detailed information refer to the *KONE IP DOP Object* manual.

Elevator Object

The Elevator object manages the elevator-specific access control functions of a single elevator cab. It determines which entity is allowed access to which floors in an elevator cab, and logs which floor buttons in an elevator cab are pressed by an entity. The reader input is delivered to the Elevator object through an Access Control object. The Elevator object may monitor a Door Sequence object to detect the override mode. When in override, all floors are accessible as if they were in public access.

In a low level elevator integration, the access control system interfaces with the elevator control system through a multitude of binary inputs and outputs. Granting access to floors is achieved by activating outputs, reporting the pressed floor buttons is achieved by monitoring inputs.

The arrows in the diagram show the message flow between the different components of the elevator interface. In a low level elevator integration, the Elevator object is a child of the controller's device object.

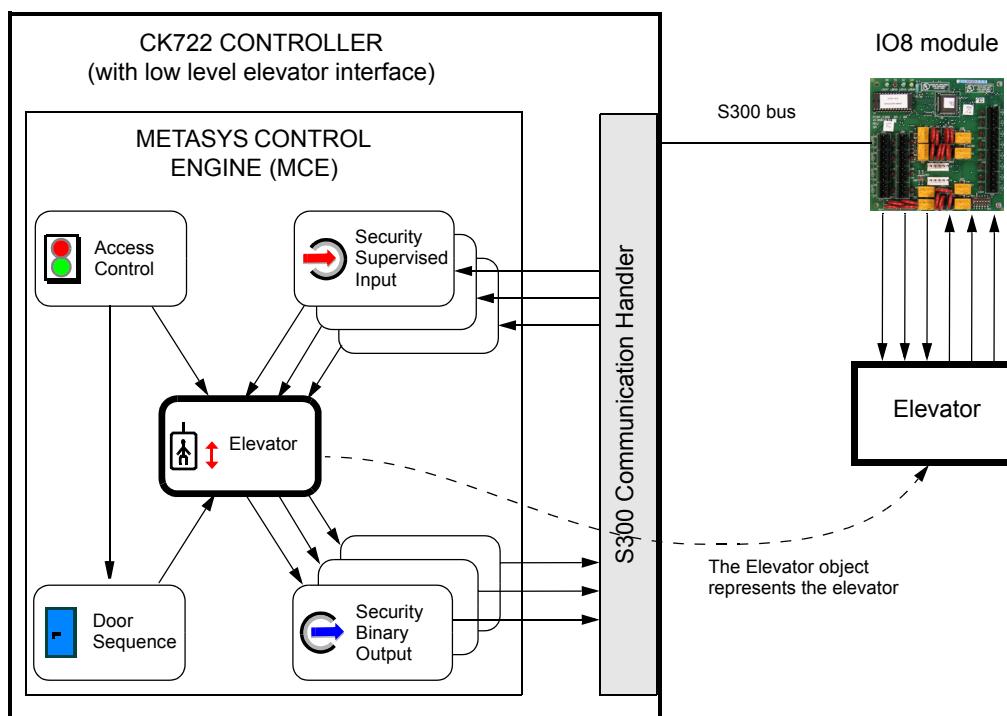


Figure 3-35: Elevator Object

For detailed information refer to the *Elevator Object* manual.

Site Object

The Site object allows you to configure the site's time zone.



Figure 3-36: Site Object

Folder Engine Object

The folder object allows you to organize other objects in the SCT configuration tree.



Figure 3-37: Folder Object

Schedule Object

The Schedule object works behind the scenes of the Scheduling feature. The Schedule object updates attribute values of objects according to the time of day. These times can be specified for days of the week and exception days.

Exception days (Exception Schedule) are days when you do not want the weekly schedule to operate, such as holidays. They can be defined as specific dates, ranges of dates, week and day, or by reference to a Calendar object. You can define a different set of activities to occur on the exception days.

By setting the *Time Zone* attribute to a P2000 Time Zone, you can have the *Weekly Schedule* and *Exception Schedule* attributes filled in automatically by the P2000 server.

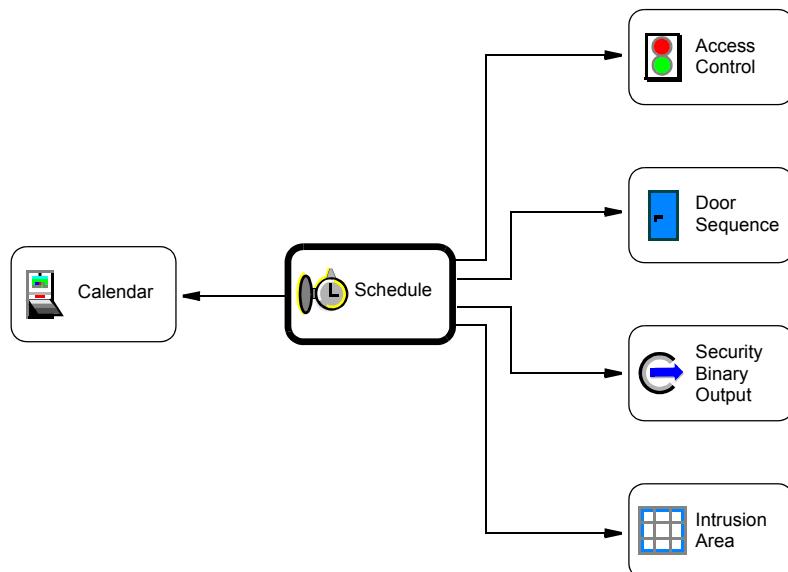


Figure 3-38: Schedule Object

For detailed information refer to the *Schedule Object* manual.

Calendar Object

The Calendar object is used by the Scheduling feature to maintain a list of dates designated as exceptions to the normal schedule. It allows you to accommodate for a special day or days, like a holiday, in which the P2000 system should run differently from the usual operation.

NOTE

In the CK722 controller, three Calendar objects are automatically created to represent the three different holiday types as defined by the P2000 server. Those Calendar objects are for use by P2000 only, and cannot be modified from the SCT. Typically, there is no need to add more Calendar objects for the operation of the P2000 system.

You can define Exception Schedule days (when you do not want the Weekly Schedule to operate) as specific dates or ranges of dates.

Typically, a Schedule object reads the Present Value of the Calendar object to check whether a present day is an exception day. On these exception days a different set of activities from those in the weekly schedule can be defined in the Schedule object.

Multiple Schedule objects can reference a single Calendar object, so that only the Calendar object needs to be changed to affect all schedules.

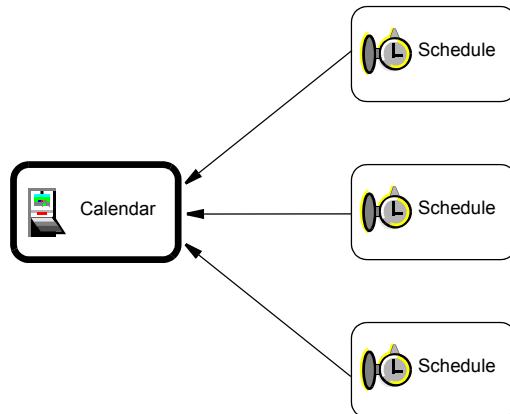


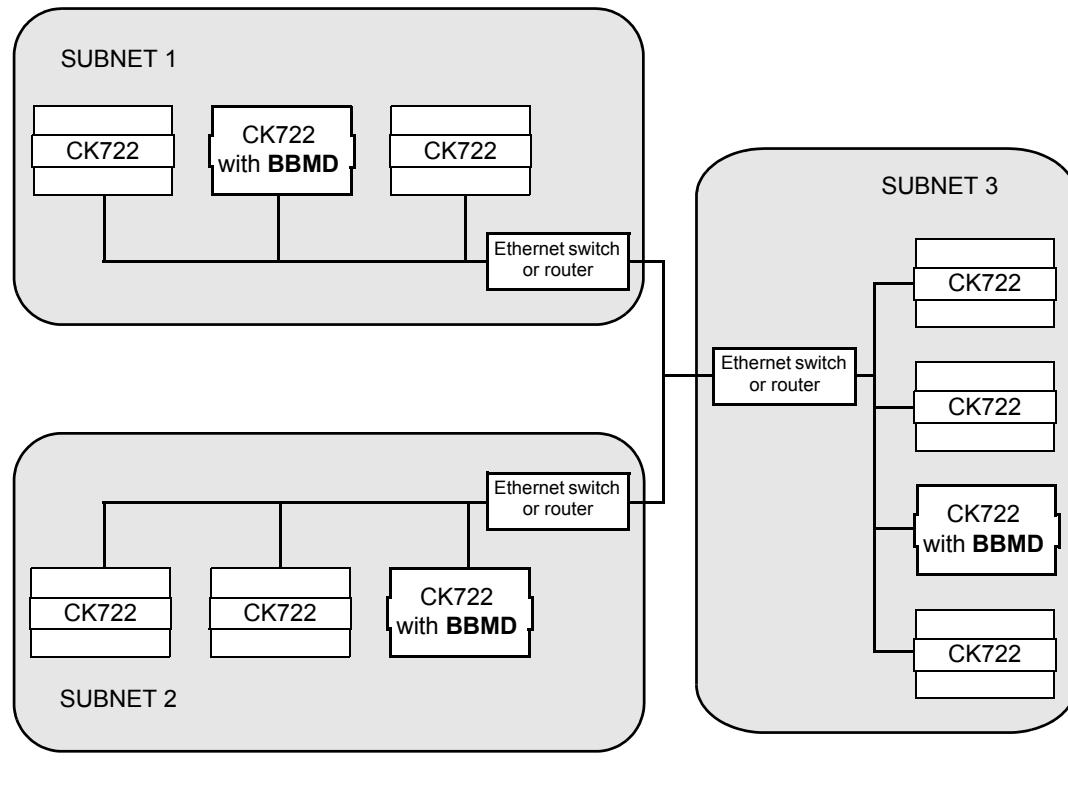
Figure 3-39: Calendar Object

For detailed information refer to the *Calendar Object* manual.

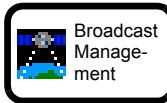
Broadcast Management Object

The Broadcast Management object is used to establish BACnet Broadcast Management Devices (BBMDs). BBMDs, in turn, provide a mechanism to transmit broadcast messages from one IP subnet to another IP subnet.

BBMDs are required for all peer-to-peer applications that span IP subnets. The P2000 host itself may be on a separate IP subnet without the need for a BBMD.



BBMDs are represented by Broadcast Management objects.



Note: Each Broadcast Management object broadcasts *within* its own subnet, and sends broadcasts to other Broadcast Management objects.

Figure 3-40: Broadcast Management Object

For detailed information refer to the *Broadcast Management Object* manual.

Security Notification Class Object

The Security Notification Class object defines event-handling options and sends the Event Notification messages initiated by the objects to other devices on the network.

The following diagram shows the major blocks the Security Notification Class object interacts with.

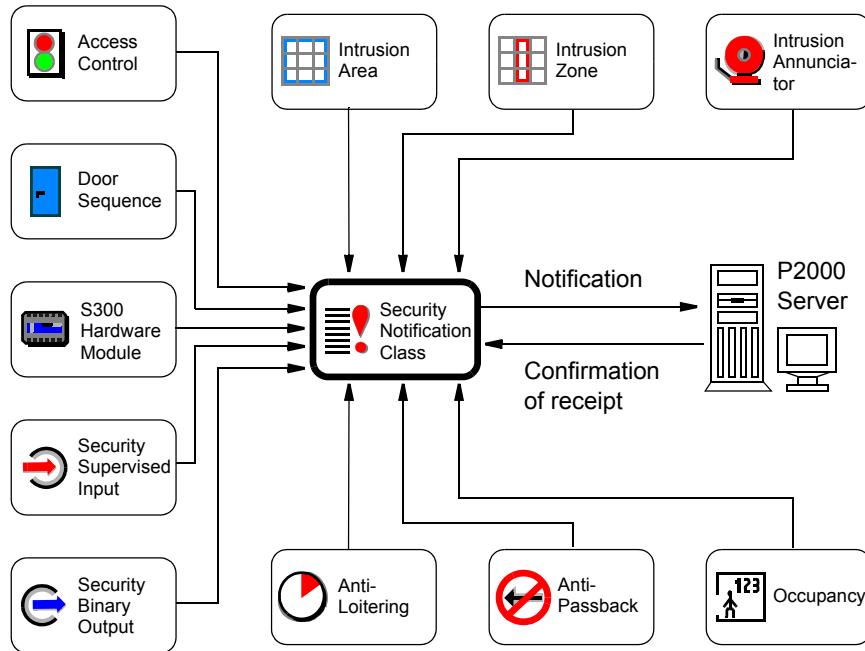


Figure 3-41: Security Notification Class Object

For detailed information refer to the *Security Notification Class Object* manual.

RS485 Bus

The CK722 uses the RS485A and RS485B ports to communicate with S300 I/O modules on the serial bus.

The following hardware modules (also called field devices or terminals) can be used with CK722:

- S300-DIN-RDR2S - Reader
- S300-DIN-RDR2SA - Reader
- S300-KDM - Keypad/Display module
- S300-RDR2 - Reader terminal
- See “Reader Module Comparison” on page 2-40 for a quick reference table listing features of reader modules. - Supervised Input/Output terminal
- S300-IO8 - Unsupervised Input/Output terminal
- See “Reader Module Comparison” on page 2-40 for a quick reference table listing features of reader modules. - Unsupervised Input terminal
- S300-SI8 - Supervised Input terminal

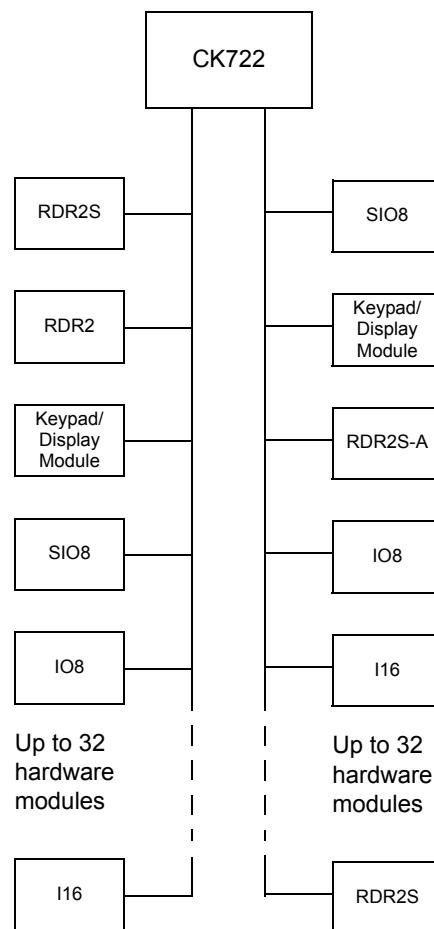


Figure 4: RS485 Buses

Ground Wiring on S300 Bus

Some of the Johnson Controls devices that can be connected to an S300 bus have opto-isolated RS485 interfaces, while others do not.

Examples of devices with opto-isolated RS485 interfaces include CK721, CK722, S300-DIN-RDR2S, and S300-DIN-RDR2S-A.

Examples of devices without opto-isolated RS485 interfaces include S300-KDM, CK705, S300-SI8, S300-SIO8, S300-IO8, and S300-I16.

As long as *all* the devices on an S300 bus have opto-isolated RS485 interfaces, connecting the Ground wire (also called COM or REF) is not needed.

As soon as even one device with a non opto-isolated RS485 interface is connected to the S300 bus, it is necessary to connect the ground wire to all devices on this S300 bus, including the devices which do have opto-isolated RS485 interfaces.

CK722 APPLICATIONS

This chapter provides detailed information on various features and functions of the P2000 SMS, useful when developing security applications that will be applied to the CK722 controller. See also “Chapter 7: Creating Job-Specific Templates” for information on how some of these applications can be configured using the P2000 SCT.

For other P2000 SMS information, especially that which pertains to the P2000 host software, refer to the *P2000AE Software User Manual*.

ENTITY

An entity is a **Person**, **Asset** or **System Account** defined in the P2000 SMS. Once an entity record is entered into the system, according to the system’s configuration settings, the P2000 SMS can provide or deny the entity access to defined areas of the facility based on an entity’s security privileges, track the entity’s location, provide the entity access to the P2000 host software, and in the case of the System Account entity type, interface with another system (e.g. MIS interface). Entity types serve to categorize entities at a high level.

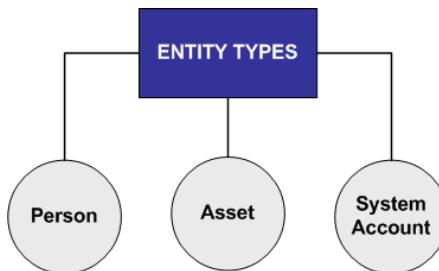


Figure 4-1: Entity Types

NOTE

Entity types cannot be changed.

When adding an entity to the P2000 database, you must select an entity type before any other data can be entered.

Assets

In older versions of P2000, an entity was called a cardholder. However, the system was limited to tracking only persons, not assets. The new asset feature allows you to track the location of physical assets in the facility, such as computer equipment. If the asset is detected in an unauthorized area of the building, an alarm can be generated. See “Radio Frequency Identification (RFID) Tags” on page 4-11 for more information.

Entity Category

Each entity type can have multiple, user-defined entity categories. This allows the user to differentiate between different entities within an entity type. For example, the Person entity type may have entity categories such as Regular Employee, Temporary Employee, Visitor or Guard. The Asset entity type may have categories like Desktop, Notebook, or Monitor. The System Account entity type may have categories like U.S. Account, Europe Account, and Asia Account. See Figure 4-2.

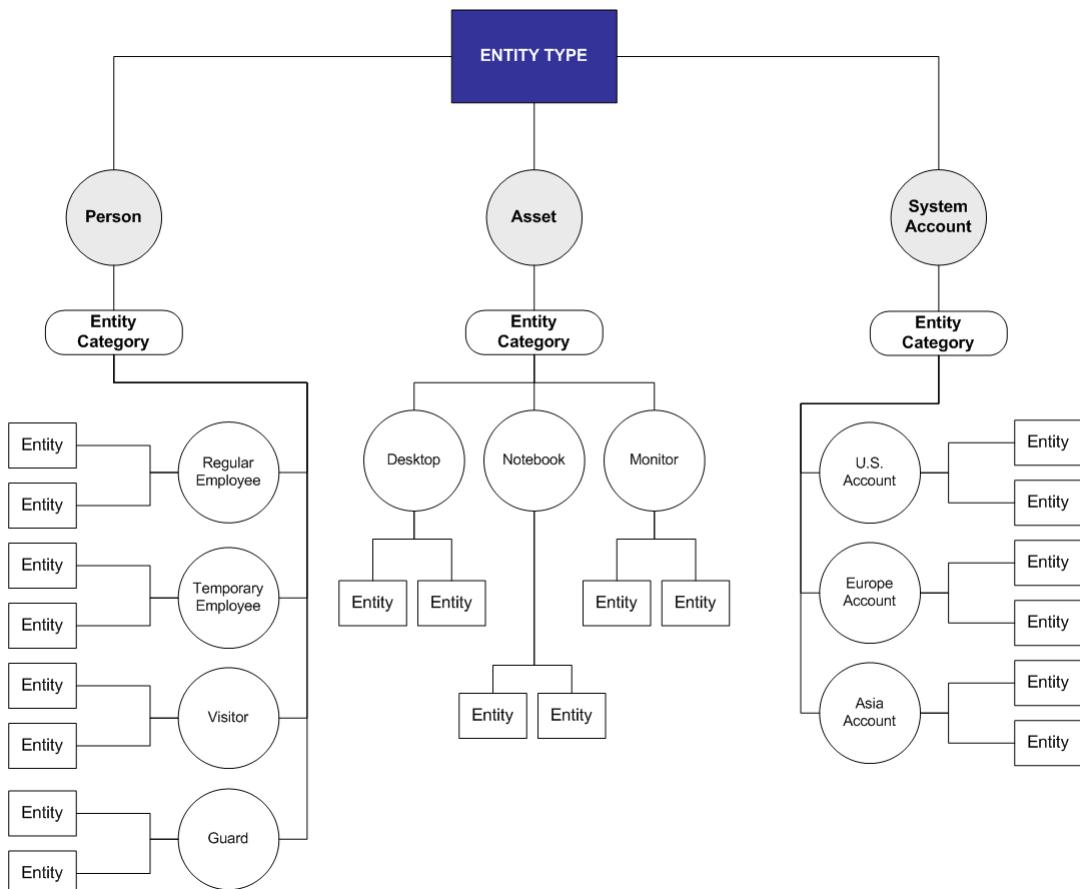


Figure 4-2: Entity Category Diagram

The entity categories in Figure 4-2 are provided as examples.

Entity Group

An Entity Group consists of a user-defined group to which entities are assigned. An Entity Group can be used for sponsoring and/or escorting other entities. With the escort feature, you can assign an entity group to an entity as an escort, meaning that every entity within the selected group becomes an escort of the entity assigned to be escorted. See “Entity Escort” on page 4-5 for information on escorting entities with an entity group.

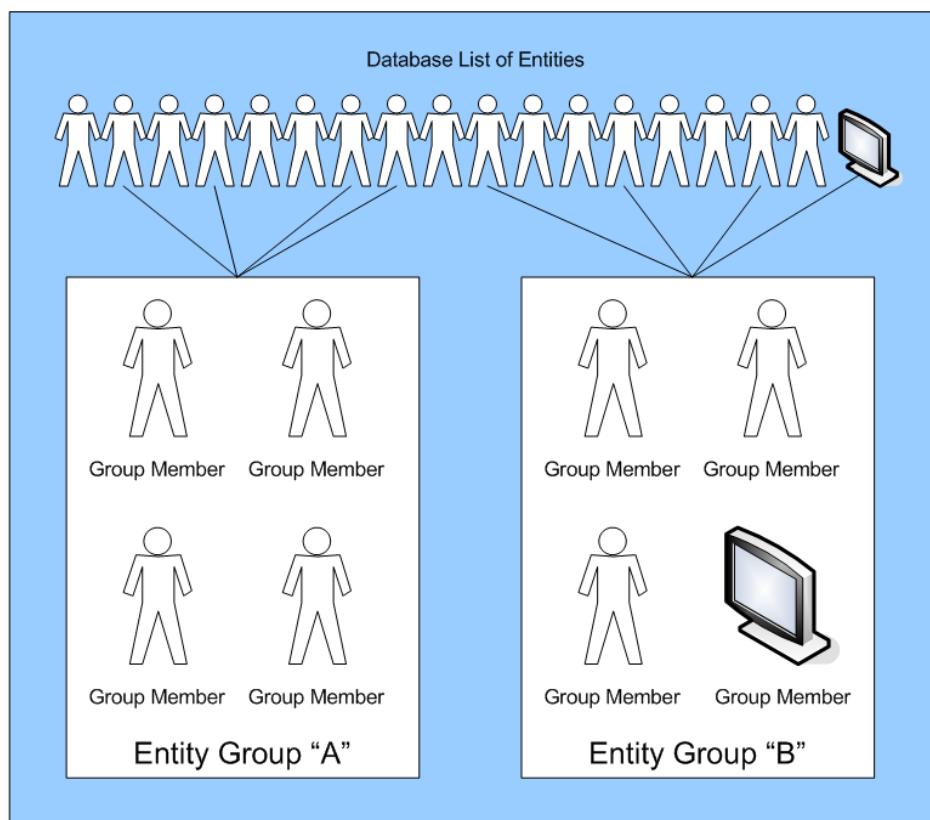


Figure 4-3: Entity Group

Assets can also be a member of an entity group, as illustrated in Figure 4-3.

Entity Sponsor/Owner

The entity management application enables you to assign one or more sponsors or owners to an entity. A sponsor is someone who sponsors an entity, thereby accepting some level of responsibility for the entity. A typical example of a sponsor is an employee who invites a visitor to the facility to conduct business. In this example, the employee is the sponsor of the visitor. You may also assign regular employees as sponsors of temporary employees.

Since entities can also be assets, you can assign an entity (person) as the owner of an entity (asset). For example, if an employee is assigned a particular piece of

electronic equipment, this equipment is added to the P2000 SMS as an asset, and the employee is assigned as the owner of that asset.

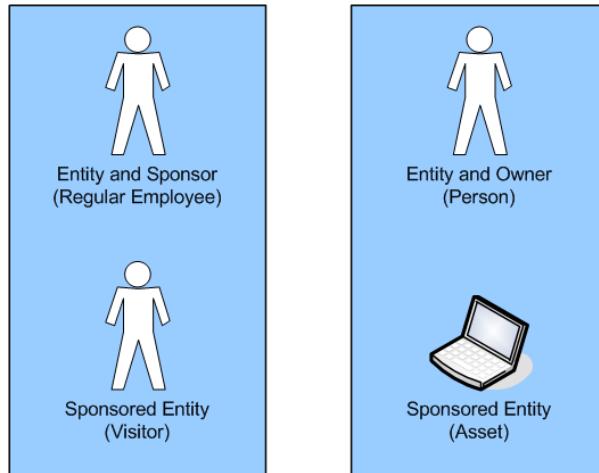


Figure 4-4: Entity Sponsor

A sponsored entity may also be assigned an entity group as the sponsor. If this case, everyone within the entity group is the sponsor/owner of the entity (person or asset). See “Entity Group” on page 4-3.

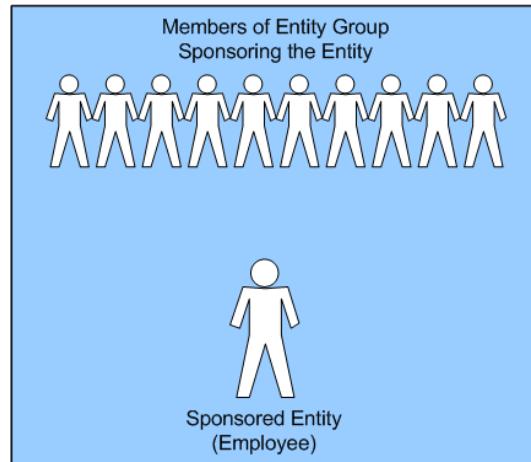


Figure 4-5: Entity Group Sponsor

Entity Escort

You may assign one or more escorts to an entity, if the entity requires some level of supervision inside the facility. If properly configured, the P2000 SMS can deny access to an escorted entity until the system identifies that the escort(s) is present. An escort must present an identifier, such as an access badge, at the controlled location for the system to verify whether he/she is present.

An escort can be one of the following:

- **A single entity unassociated with an entity group.** An escorting entity does not have to be assigned to an entity group. A escorted entity can have one or more escorts assigned to him/her, and each entity does not have to be assigned to an entity group.
- **A single member of an entity group.** Entities who are members of this group are able to escort a non-group member (the escorted person or asset) through the building. In order for the escorted entity to be granted access, one member of the single escort group must be present and identify him/herself within a defined period of time.
- **All members of the entity group.** Entities that are members of this group are required to enter a controlled area together. All entities must present and identify themselves within a defined period of time to be granted access.

See “Entity Group” on page 4-3.

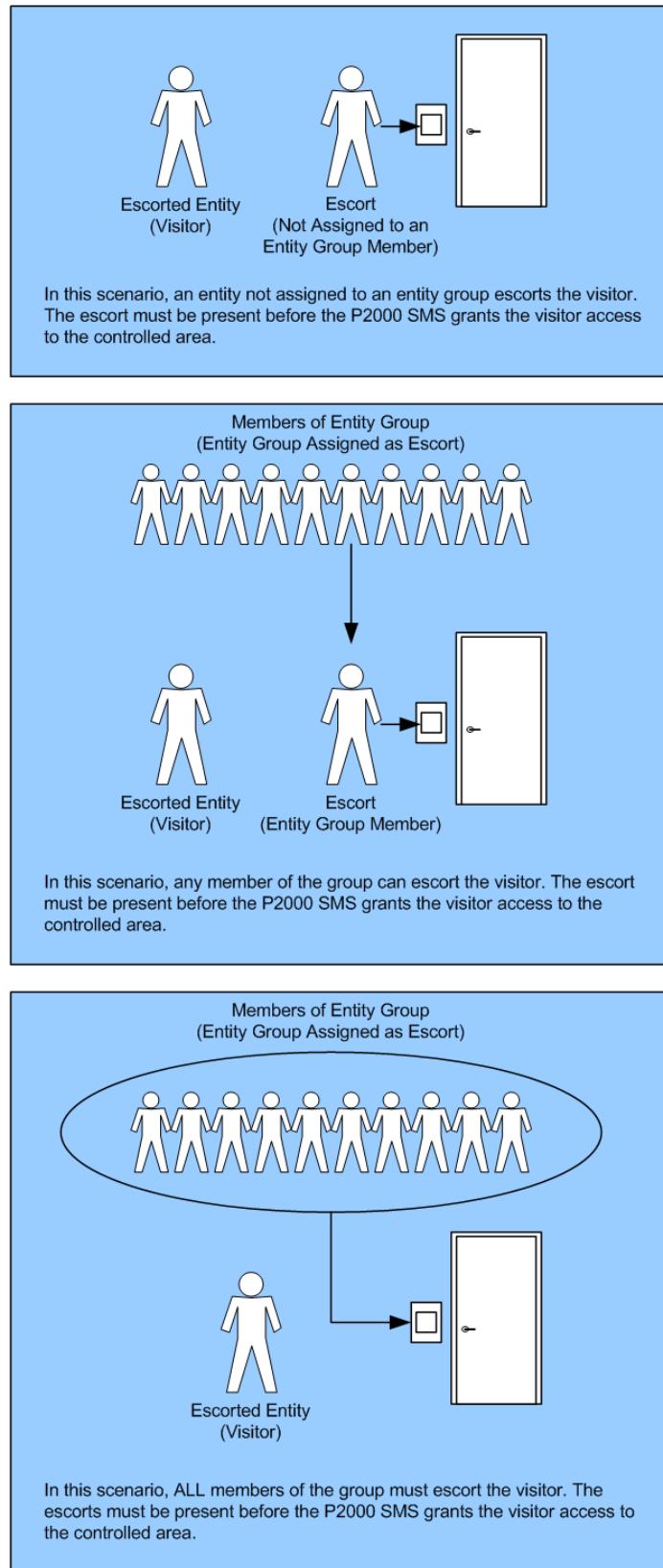


Figure 4-6: Entity Group Escort

NOTE

An entity can also escort an asset to help prevent unauthorized removal of equipment from the facility.

Organization

The P2000 SMS allows you to associate an entity with a user-defined entry for one of the following categories:

- Company
- Division
- Department
- Team

These categories cannot be changed. You may only add, edit, or delete category entries.

Validation

The P2000 SMS provides a validation date range for each entity. When an entity (person) is valid, he/she can perform actions in accordance with his/her assigned privileges (e.g. access to the facility, access as a P2000 user, etc.). Once the validation period expires, those privileges are revoked.

This tool is especially useful for temporary employees or visitors. Someone visiting a facility for a day could have a validation period that starts and ends on that day. When the validation period ends, the P2000 SMS no longer grants the visitor access to the facility.

Status

This feature allows you view to the current status of an entity based on identifier activity. For example, if an entity has presented his access badge identifier at a reader, the system status will display the date and time the identifier was presented and which reader was used. If the system grants access to the entity, the transaction is **Valid**. If the system denies access to the entity, the transaction is **Invalid**. The P2000 software displays both valid and invalid transactions.

If the entity has presented an identifier at a muster reader (i.e. during an emergency), the information on the mustering, such as where the entity mustered, can be accessed from the P2000 software.

Area information associated with entity activity is also available. This includes information consisting of the date and time the entity was detected in the area and which reader was used.

Journal

The Journal supplements entity information by storing notes associated with each entity. For example, you may want to keep track of a person's parking violations, or keep a record of persons that attended specific company training, or track persons with suspicious behavior. You can also add notes for assets, such as the expiration date for leased equipment and who is currently responsible for the equipment.

User Accounts

All entities assigned as a person or system account can become a P2000 user, and each P2000 user account can be associated with:

- Multiple partitions
- Multiple enterprise sites
- Multiple user roles

User-Defined Fields

P2000 allows you to configure user-defined fields (UDFs) for use with entity records. UDFs help by allowing you to add a field that is not provided with the P2000 software. For example, if you wish to track cars used by entities, you could add multiple UDFs to record car data such as Make, Model, License, and Color.

The data types available with UDFs are:

- Date
- Numbers
- Strings
- Boolean

IDENTIFIER

Identifiers, also known as "credentials" in access control, enable the P2000 SMS to identify and track an entity in the system. For person entity types, an identifier can be used to identify the entity and prompt the P2000 SMS to grant or deny access to the facility. For asset entity types, an identifier can be used to track the location of an asset in the facility. If the asset is identified in an unauthorized area of the building, the system can generate an alarm.

The P2000 SMS supports the following identifiers:

- Identification badges
- Access badges

- Personal Identification Numbers (PINs)
- Radio Frequency Identification (RFID) Tags

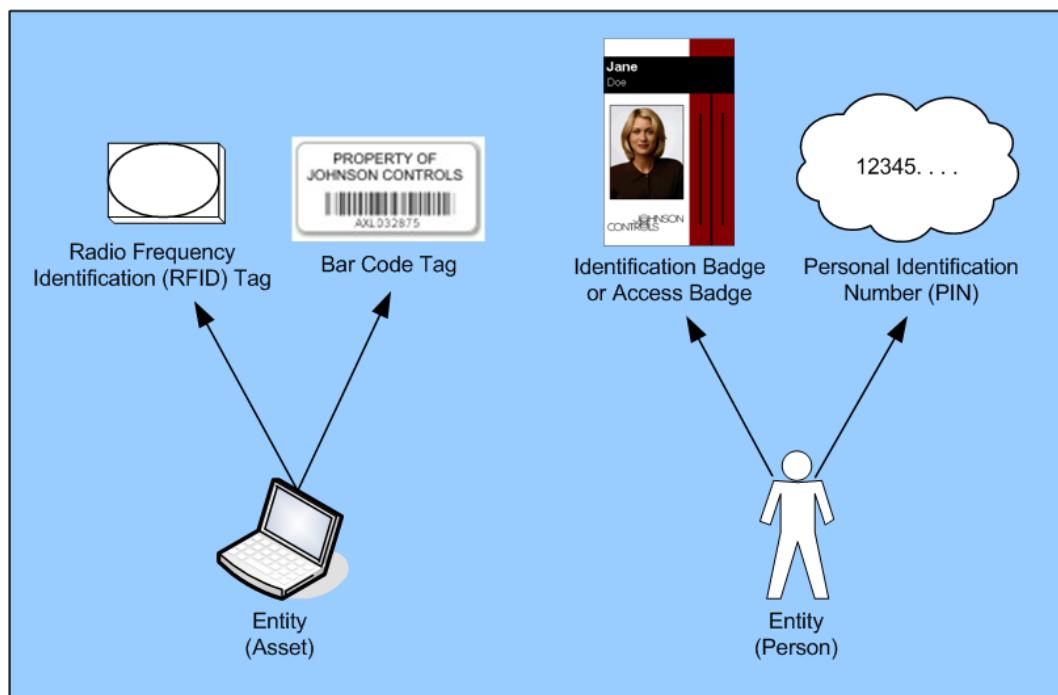


Figure 4-7: Identifiers

An entity may have multiple identifiers. For example, a person might have both badge and PIN identifiers. In some security applications, an entity may have to present an access badge to a reader, and immediately enter a PIN on the reader's keypad, for the system to grant him/her access to the controlled area.

Identification Badge

Similar to access badges, identification badges are typically the size of a credit card and are used to visually identify an entity. However, unlike access badges, they do not have access control capabilities – they cannot be used to access a controlled area of a facility.

Both identification badges and access badges consist of various components that are printed on the badge for identification purposes. This can include the following:

- Portrait image
- Fingerprint image
- Signature image
- Barcode

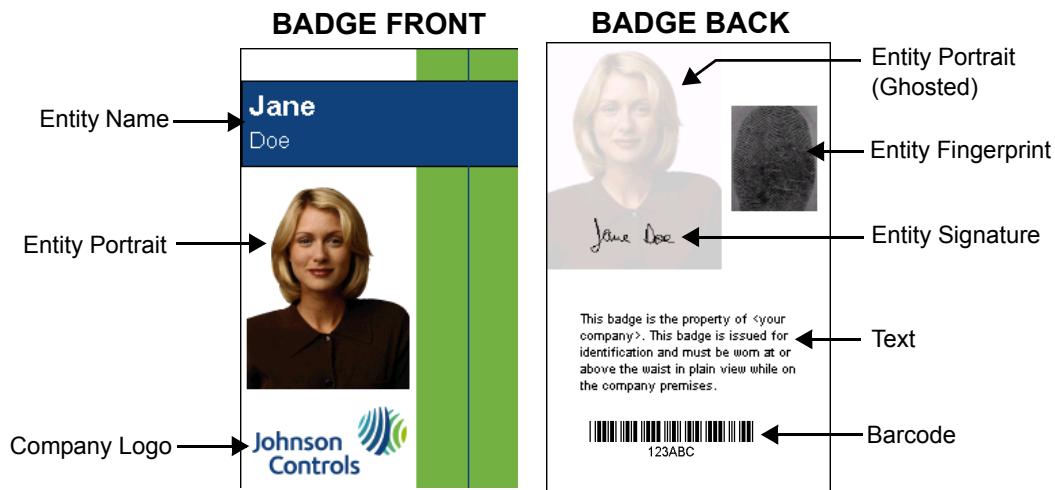


Figure 4-8: Sample Badge Identifier

Identification badges can be created with a P2000 badging station connected to appropriate capture equipment and a badge printer. See the *P2000AE Integrated Video Imaging Manual* (EPI Builder® or ID Server Series) for information on creating badges with the P2000 SMS.

Access Badge

Access badge identifiers serve both to visually identify an entity and to prompt the P2000 SMS to grant the entity access to a controlled area of a facility. Access badges are programmed to activate a system reader, typically used at portals (e.g. doors, gates, etc.), and are assigned to occupants of a facility.

Each badge is unique. When an occupant presents an access badge to a reader, the reader transmits the badge's information to a controller, such as the CK722. The controlling device reads the badge's information and grants or denies access based on the access rights and privileges defined in the P2000 SMS.

Common types of access cards, which are used to create access badge identifiers, are:

- Proximity cards
- Smart (electronic) cards
- Magnetic stripe cards
- Wiegand cards

For more information on access cards, refer to the *Code of Practice (Part I)*.

Personal Identification Number (PIN)

PIN identifiers are programmable, numeric codes assigned to entities for authentication purposes. PINs can be entered on keypads for use in the P2000 SMS. The P2000 SMS can grant or deny the entity access to a controlled area based on the validity of the entered PIN.

A PIN may also be used in conjunction with another identifier, such as an access badge identifier. The P2000 SMS can be configured to require the entity to present a valid access badge and enter a valid PIN before granting him/her access to a controlled area.

For information on PIN identifiers, see *Appendix B: Using Keypad Readers*.

Radio Frequency Identification (RFID) Tags

An RFID tag is typically a small adhesive label that attaches to an asset and can be scanned by a long-range RF reader to determine the location of the asset, thus providing a level of asset protection.

In the P2000 SMS, RFID tags function like access badges. A reader can scan the RFID tag only when the tag comes within scanning range of the reader. At that point, the reader can determine the location of the asset. If the asset is not escorted by the proper escort, or if the asset is located in an unauthorized area of the facility, an alarm can be generated.

However, the P2000 SMS provides limited RFID asset protection. In a standard RFID protection system, the RFID tag transmits its unique RFID to the reader network every n seconds (e.g. every 5-10 seconds). In this type of asset protection system, the system always knows the location of the asset. The P2000 SMS does not provide this level of asset protection.

ACCESS CONTROL

Access Control is a security method that controls the traffic of individuals between an unsecured area and a secured area. An Access Control system, such as the P2000 SMS, grants or denies someone access based on his/her security settings. These settings determine, for example, the facility/area where the occupant has access privileges, the portals within the protected area that will unlock/open, and the days and times during which the occupant can access the area.

The protected areas of a facility are called “access points.” These are controlled by a combination of access control devices. Electronic Access Control (EAC) is an access control method that uses computer technology to control and monitor these points and areas.

Access control devices require the occupant to use an identifier (credential) before the system grants him/her access. Identifiers are something presented or held by an occupant as proof of identity.

In the case of access control, identifiers are:

- Something you *have* (e.g. access badge identifiers presented to a reader device)
- Something you *are* (e.g. biometric devices that scan physical characteristics)
- Something you *know* (e.g. PIN identifier entered on a keypad)

Basic Access Control Components

Although some security management systems are very elaborate and may incorporate a variety of other systems (integration), a typical SMS consists of the components described below.

Host Computer (P2000 Server)

The Host Computer (P2000 Server) provides the following:

- A centralized database and easier database maintenance
- Full, single-point control of access points and peripherals
- Single-point monitoring of all alarm conditions
- Easy disaster recovery (with proper database maintenance)
- An easier, more effective means of integration with other computer-based systems

Redundant System

Security applications requiring minimal or no down time can benefit from a redundant configuration. A redundant system protects its data or functionality by providing a duplicate element such as a second host, component, or database (RAID) that takes over for the failed element.

There are two main types of redundancy systems:

- **Disaster Recovery**
This redundancy option is achieved by having two separate servers with identical databases. In the event of a failure in the primary server, the standby server is able to immediately take over. Other redundant systems require the servers to be in close proximity. Remote Redundancy refers to the ability to have the two servers in physically separate locations.
- **High Availability**
This redundancy option uses a server cluster, which is a group of independent servers running a cluster service, to preserve client access to resources during failures and planned outages. If one of the servers, or one of the services in the cluster, is unavailable due to failure or maintenance, resources and all workstations will move to the other available cluster node.

Supervisory Controllers (CK722)

Supervisory controllers allow access points and the host computer to communicate. In most systems, the controllers can handle access control functions with very little interaction required at the host computer. Some controllers may also incorporate the card reader interfaces. This reduces the amount of hardware needed.

Field Device

The field device, such as the RDR2S-A, is a board or module that serves as the interface between the field controller and the reader itself. In some systems, these may also incorporate relays for control of peripheral devices and/or alarm-monitoring points to which contacts, switches, motion detectors, or other input devices can be connected.

Reader

A reader is an access point device at which access badge identifiers are presented, the access badge's information is read, and the information is transmitted to a controlling device (controller or host computer, depending on system configuration) via a reader interface. Some readers may also have an alphanumeric keypad at which a Personal ID Number (PIN) is entered as a requirement for access.

Door Hardware

Door hardware devices can be divided into the following categories:

- Door contacts
- Electric locks/releases
- Request to Exit (REX) or Egress devices

Door Contacts

Door contacts are switches generally activated (opened or closed) by means of a magnet. Other types are activated by physically applying pressure to the switch by means of a spring-loaded roller ball. Switches can be surface-mounted (on the face of the door and door frame) or flush mounted, embedded in the door and frame (usually above the door).

Switches can be normally open (NO), normally closed (NC), or both.

- A NO switch is *open* until a magnetic field or pressure is applied.
- A NC switch is *closed* until a magnetic field or pressure is applied.

Some switches, such as a “biased” switch, can even detect if the distance between the switch and the magnet changes to an unacceptable distance. This prevents someone from taping a magnet to the switch and “blinding” the system from seeing a change of state (open-to-closed or closed-to-open, depending on the type of switch used).

Electric Locks

There are three main categories of electric locks: Magnetic Locks, Electric Strikes, and Electric Locks.

Magnetic Locks

These locks work by applying electrical power to an electromagnet mounted on the door frame. This in turn attracts a metal plate mounted on the door. This type of lock is fail safe (fail unlocked). A fail safe door unlocks upon lock or power failure. These require lock activation in both directions through the door.

Electric Strikes

These devices mount in the door frame, replacing a conventional lock strike plate. They work by means of an electric coil that engages or disengages the locking mechanism. Electric Strikes are available in two modes of operation. These modes are fail safe (fail unlocked) or fail secure (fail locked). Fail safe devices require power to maintain them in a locked condition. Fail secure door strikes remain locked upon lock or power failure and require power to unlock the device.

Electric strikes allow for “free exit.” With “free exit,” the lock does not need to be activated when someone exits through the door under normal conditions. A simple turn of the doorknob is sufficient.

Electric Locks

These locks replace the conventional door knob/door handle (mounted on the door). As with the electric strike, the lock can be fail safe or fail secure. It may be free exit.

Two characteristics distinguish this type of device from the others:

- Requires a power transfer cable (cable brings power from the door frame through the door, and to the lock; it is less secure and less aesthetically pleasing). Another means could be a power transfer hinge (wiring is run from the door frame through the hinge to the lock; it is much more secure and more aesthetically pleasing).
- Can be unlocked with a key just like any conventional lock.

Request to Exit (REX) or Egress Devices

Request to Exit (REX) or Egress devices, often called “REX buttons,” are used to activate a strike/lock to unlock a door. These devices vary, depending on the site, level of security required, codes and regulations, and other considerations. The most common types are described below.

Motion Detectors

When a motion detector is activated, its internal relay changes state. This change triggers a request to exit command to the door controlling hardware.

Panic or Crash Bars

These are typically used to allow egress through emergency doors. They may be simple mechanical devices outfitted with a switch that signals a request to exit to the door controlling hardware. They can also be sophisticated devices that trigger a change of state in the lock by interacting with the human body's static electricity, or they can be free exit.

Push Button Switch

This is a button which includes a switch. When activated, it can cut power directly to the lock, signal a request to exit, or both. They come in various sizes and colors; some are designed to meet national code requirements. Although not an official standard, green has become the standard for access control egress push button switches.

Key Override Switches

These are mechanisms requiring a conventional key to toggle a switch. This in turn changes the state of the lock to which it is connected (by cutting power, signaling a request to exit, or both).

Access Control Features and Applications

This section describes various access control applications that can be configured with the P2000 Security Management System (SMS).

Portal Entry (Single Reader/Keypad)

This application commonly consists of a single input device (reader/keypad) installed at a door controlled by the SMS. In this application, entities must present their identifier so that the SMS can identify the entity. Once the entity is identified, the SMS grants the person access by unlocking the door (energizing or de-energizing the door hardware, such as a magnetic lock or electric strike device).

Portal Entry

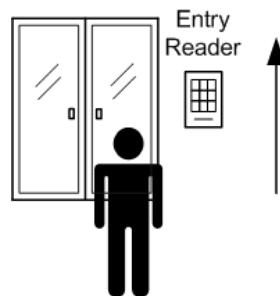


Figure 4-9: Portal with Entry Reader

Portal Entry and Exit (Card-In-Card-Out)

In some applications, a portal can have an entry and exit reader/keypad. With this hardware configuration, the P2000 SMS can be configured to require the entity to present an identifier to enter and exit the controlled area.

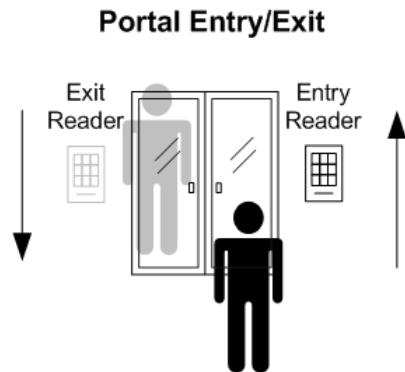


Figure 4-10: Portal with Entry and Exit Readers

Video Imaging

Video Imaging allows you to design and print access badge or identification badge identifiers complete with graphic images, such as portraits and signatures.

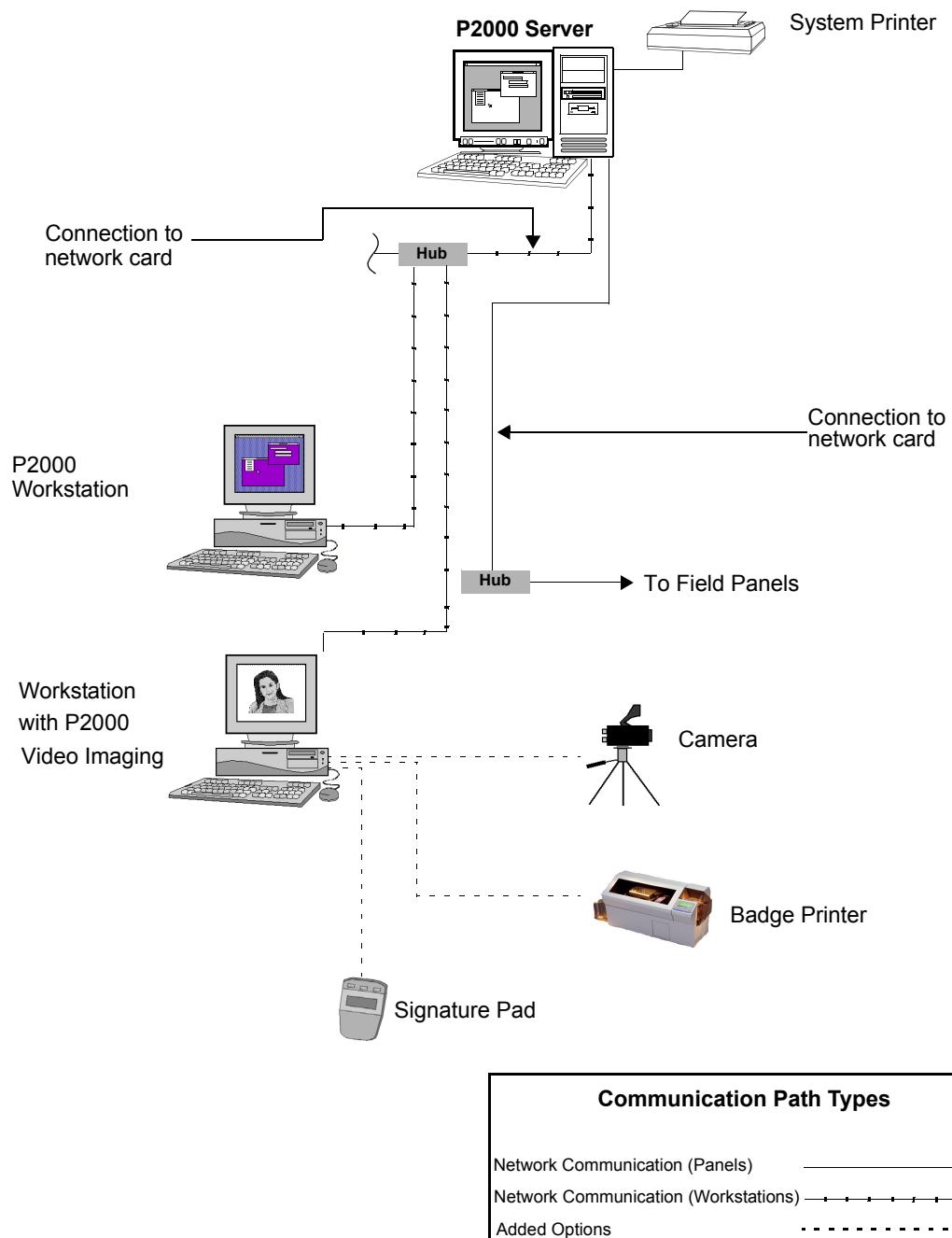


Figure 4-11: Typical P2000 Video Imaging Configuration

See “Identifier” on page 4-8 for more information.

See also the *P2000AE Integrated Video Imaging Manual*.

Fail Safe (Fail Unlocked) and Fail Secure (Fail Locked)

During an emergency, or in the event power is lost to an access control system, the portals will either unlock (fail safe/unlocked) or remain locked (fail secure/locked). The type of security required at the facility usually determines whether portals are fail safe (fail unlocked) or fail secure (fail locked). Most facilities will have fail *safe* doors so occupants can escape the premises during a power outage. Other high security facilities, such as prisons, may have fail *secure* doors so occupants, such as prisoners, will not be able to leave during a power outage.

- A fail *safe* door unlocks upon lock or power failure.
- A fail *secure* door remains locked upon lock or power failure.

Visitor Management

The Visitor Management feature introduces an easier way for entities to make visitor identifier requests, allowing the identifier, such as an access badge, to be ready when visitors arrive. Prior to a visitor's arrival, an authorized entity (e.g. guard) enters the appropriate visitor data into the system, assigns a visitor sponsor and/or escort, enters the date and time period of the scheduled visit, and assigns access privileges using Access Profiles.

Upon arrival, the visitor is signed-in with a simple mouse click and subsequently the visitor identifier is provided. This application eliminates the need for paperwork and keeps track of who requested the identifier and who sponsored the visitor.

See “Entity Sponsor/Owner” on page 4-3 and “Entity Escort” on page 4-5. See also the *P2000AE Software User Manual* and the *P2000AE Web Access Manual*.

Alarm Monitoring

Alarm monitoring is at the heart of the P2000 SMS. According to the P2000 SMS's configuration, alarms are displayed in the Alarm Monitor queue as they occur. Operators assigned to monitor alarms respond according to individual company policy, and the alarm instruction and response text configured for the various alarm types. The Alarm Response text can be pre-configured for operator selection and/or set to enter manually for a more appropriate response.

Scheduling

The scheduling feature allows you to create a time/date schedule for certain functions, such as unlocking a door for a defined amount of time during defined days of the week. For example, if the door will only be unlocked during normal business hours, you could create a schedule to override the lobby's door's default settings and unlock the lobby door (unlocked and open) at 8:00 and lock it (locked and closed mode) at 17:00, Monday through Friday.

In the P2000 SMS, each schedule consists of a *Weekly Schedule* and an *Exception Schedule*. Exception days (Exception Schedule) are days when you do not want the weekly schedule to operate, such as holidays. See “Chapter 8: Scheduling” for details.

Anti-loitering

The Anti-loitering feature allows you to monitor the time an entity spends in an anti-loitering area. If an entity exceeds the area's anti-loitering time, an anti-loitering notification is generated.

NOTE

An anti-loitering area must have an entry and exit reader so that the P2000 SMS can determine when an entity leaves an anti-loitering area.

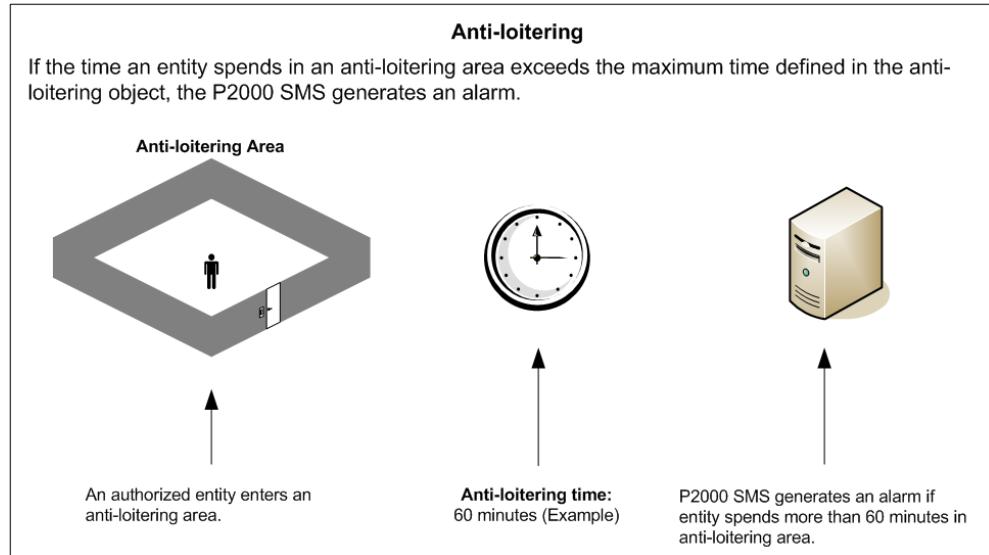


Figure 4-12: Anti-loitering Feature

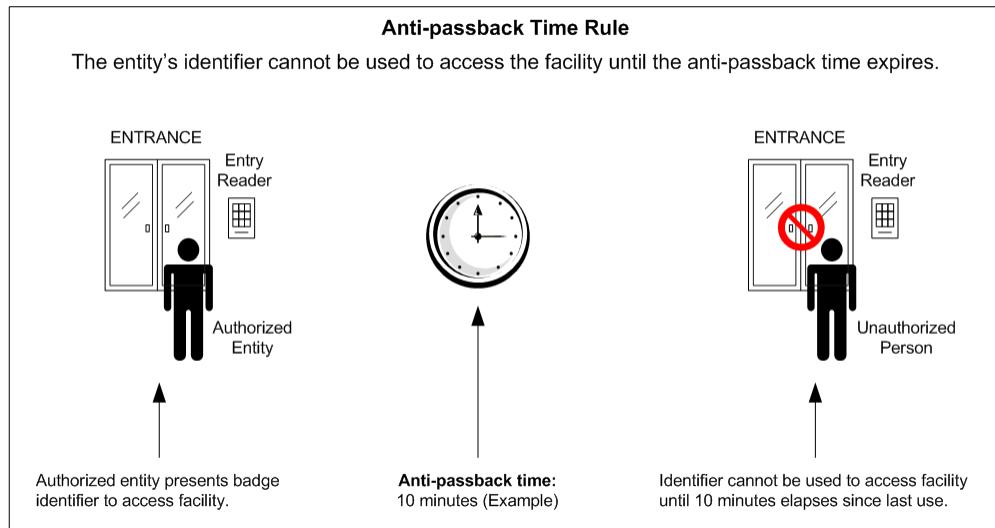
Anti-Passback

The Anti-Passback feature helps prevent unauthorized persons from using the identifier of an authorized entity to gain access to a controlled area. This is accomplished by defining the Time rule or the Entry/Exit rule.

See also the *Anti-Passback Object Manual*.

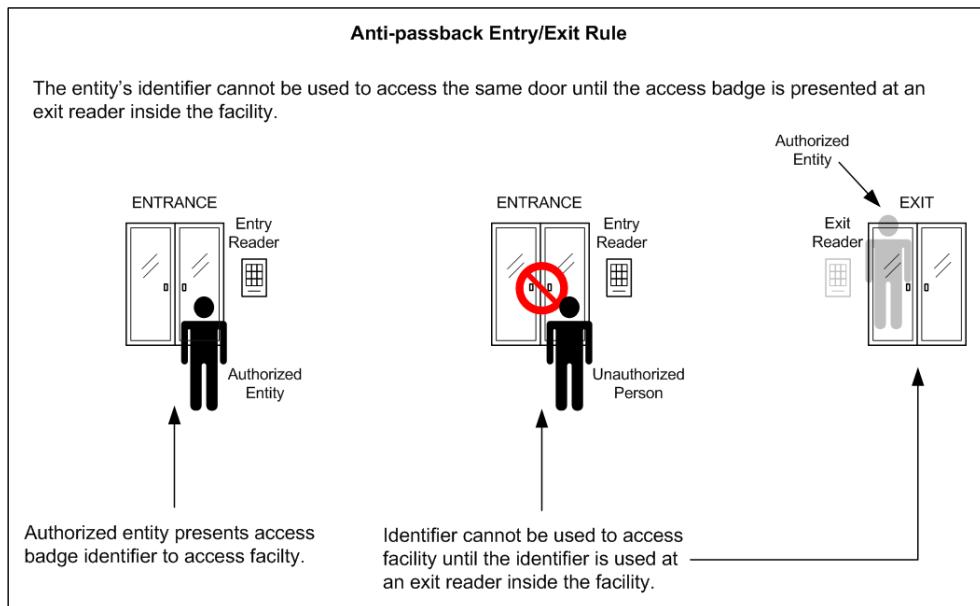
Time Rule

Once an authorized entity presents a valid identifier to access the facility, he cannot access the facility again until the anti-passback time expires. For example, John (an authorized entity) presents his access badge identifier to access the main entrance of a controlled facility. He then slips his identifier under the door a few minutes later for an unauthorized individual to enter the facility. The Time rule will disallow John's identifier to be used again until the defined amount of time expires, thereby preventing the unauthorized person from entering the facility.

*Figure 4-13: Anti-Passback Time Rule*

Entry/Exit Rule

Once an authorized entity presents a valid identifier to access the facility, she cannot access the facility again until she uses an exit reader to exit the facility. For this rule to be used, the facility must have entry and exit readers installed. For example, Jane (an authorized entity) presents her access badge identifier at an entry reader to access the main entrance. She then hands the identifier to an unauthorized person to access the facility. The Entry/Exit rule will disallow Jane's identifier to be used at the entry reader until it is first used at the main entrance's exit reader.

*Figure 4-14: Anti-Passback Entry/Exit Rule*

Occupancy

The Occupancy feature enables you to monitor the number of entities in an occupancy space. An occupancy space can have a maximum and minimum occupancy number defined. If the number of occupants exceeds the maximum or does not meet the minimum, the system generates a notification and can be configured to activate an output point, such as a “LOT FULL” sign in a parking lot.

The P2000 SMS uses an Occupancy object to track the number of entities in an occupancy space. This object allows you to set the maximum and minimum occupancy number.

The Occupancy object functions in two different modes:

- **Anonymous Mode** – In this mode, the P2000 SMS monitors the number of occupants, but does not track which entities entered or exited the occupancy space. If the entity, Jane Doe, enters an occupancy space, the P2000 SMS increases the number of occupants in the room by one, but does not report that Jane Doe is an occupant.
- **Entity Track Mode** – In this mode, the P2000 SMS monitors the number of occupants, while also tracking which entities entered or exited the occupancy space. If the entity, Jane Doe, enters an occupancy space, the P2000 SMS increases the number of occupants in the room by one, and reports that Jane Doe is an occupant. When Jane Doe exits the room, the P2000 SMS decreases the number of occupants in the room by one, and reports that Jane Doe is no longer an occupant.

NOTE

An occupancy space must have an entry and exit reader so that the P2000 SMS can determine when a person enters and exits an occupancy space.

Mustering

The Mustering feature provides the capability of tracking entity movement in the event of an emergency. During the emergency, all entities within a risk area are expected to evacuate and are required to present their identifier at a reader outside the risk area, thereby providing real time printed reports and/or online display information as to who may still be in a hazard area. The report and online display can be used to direct search and rescue operations. The list of entities still in the risk area is derived from the last known access data, and then refined by tracking identifier activity as personnel move out of the risk area.

Man-traps

Man-traps are areas of transition where visitors or occupants are granted or denied entry based on the verification of their access privileges. When moving between two access points (e.g. two electronically controlled doors), an individual will pass through a transitional area (e.g. a hallway between the two doors). In a common man-trap configuration, once passage is granted through the first door, the second

will remain locked until the first one closes and access is granted through the next door. This type of configuration is illustrated in Figure 4-15.

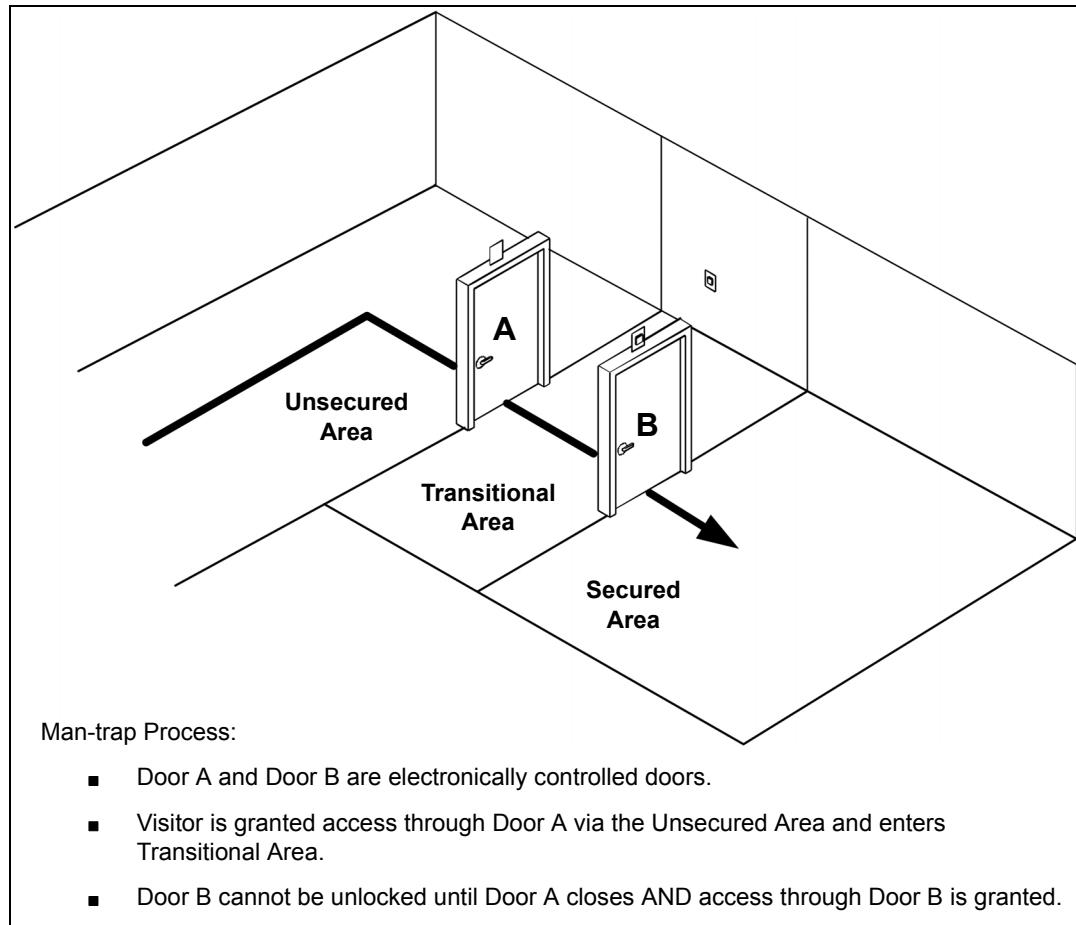


Figure 4-15: Man-traps

Input/Output (I/O)

In access control applications, I/O modules receive a signal from an access control device (input), and send a signal to another device to perform a function as a result (output).

Common access control **input** devices include:

- Request to exit (REX) devices
- Door position contacts
- Readers/Keypads

Common access control **output** devices include:

- Electric locking mechanism (door strike, magnetic lock)
- Shunt alarm

System Events

System (and card-activated) events, also called “global events,” create system-wide events initiated from the SMS host. These events can be triggered from a number of sources including access badge identifiers, controllers, field devices, inputs, outputs, SMS operators, and so on.

System Events allow for both simple and quite complex applications. For example:

- Parking lot counter.
Events can be used to count the number of cars that enter the parking lot entrance and decrease the count as the cars exit the lot. When the count reaches a predetermined number, a “LOT FULL” sign can be turned on.
- Turning lights or AC on/off in an office, a room, a building, etc. when the first entity or certain entity arrives or last entity leaves the facility.
- CCTV Camera/Monitor Assignments.
If the customer has the SMS with CCTV integration, events can be used to have a camera switch to a particular monitor on the CCTV console and a message on Server or Workstation. This event can be triggered by an invalid identifier being presented at a reader. The security guard can be alerted to see who is trying to gain access to an area they are not authorized to enter.
- Turning on an audible alarm when certain unauthorized identifier(s) is presented at a reader.

Controller Events

Controller events, also called “local events,” operate independently from the system. If the system network goes down for any reason, the controller events will continue to operate, even while the controller is offline.

Controller events are “self contained;” that is, they can be performed without the communication with the host. This has the following benefits:

- Response time is shorter.
- Network load decreases because less controller-server communication occurs.
- Events can be performed even when the host is down.

Guard Tour

The main purpose of a Guard Tour feature is to monitor and record that an area has been physically visited. It provides real time monitoring of guard activities, reporting if a guard arrives early or late at designated tour stations. The alarms and reports allow operators to manage the tours and respond to incidents.

Guard Tour stations can be either readers or input points. An identifier can be an access badge presented at a reader, a brass key presented at a key switch, or one of a variety of third party devices including barcode wands, as long as they can interface to input points.

Depending on the needs of the facility, one or more of the below examples may apply.

Guard patrol control

Patrol control is used to ensure that guards are visiting their appointed tour locations in the sequence and within the timing of the tour sequence defined by the owner.

Guard force safety

If the guard does not visit the appointed stations as required by the tour sequence, an alert is issued to an operator that a guard may have been detained or assaulted.

Process inspection management

By using the guard tour feature as a means to ensure that inspectors visit inspection stations at predetermined locations along a production line.

Visitor traffic control

Visitors can be deterred from wandering or loitering in sensitive areas of the facility by setting up a “tour station” at which a visitor check in.

Database Partitioning

Database partitioning is a powerful software feature that restricts operator access to certain user-defined segments of the access control database. Access is based upon operator password.

There are many applications for this option. For example, if you manage a building with several tenants, this option can be used to segregate the databases so one tenant cannot see the other tenant’s data or records. Another example would be to establish partitions for different departments of the same company.

INTRUSION DETECTION

Used for both commercial and residential applications, intrusion detection systems are designed to sense an intrusion into a protected building (detection) and report it to responsible parties (annunciation). This is accomplished with a combination of detection, control, and reporting devices such as input devices (sensors), output devices (bells, sirens), a control panel, a keypad, and a central host. A properly designed, installed, and configured intrusion detection system should:

- Detect an unlawful intrusion
- Identify the location of the intrusion
- Inform local security forces that an intrusion has been detected
- Signal an alarm to a remote location (e.g. a central station), so the proper authorities can be dispatched
- Signal the intruder that he/she has been detected

Intrusion detection systems are not intrusion *protection* systems. They are not designed to protect a facility from intrusion, although the presence of an intrusion system (e.g. a building displaying intrusion protection signs) can help deter a would-be intruder. *Protection* systems use a combination of physical barriers and electronic detection. Intrusion *Detection* Systems offer detection and annunciation only.

NOTE

This section provides general information on intrusion detection systems. For specific information on the P2000 Intrusion Detection System, refer to the P2000AE Software User Manual and “Intrusion Detection Examples” on page 7-56.

Basic Intrusion Components

Detection, control, and reporting devices are the basic components of a intrusion detection system. Sensors detect events. A control panel coordinates how the system responds to sensors and identifies the location of the sensor originating the signal. Local annunciators and remote communications devices allow system activity to be reported.

Perimeter

The perimeter is the boundary surrounding an area to be protected. Perimeters typically consist of a visible, physical structure, such as a fence or a building’s walls.

Detectors

Detectors have sensors that detect an intrusion along an interior and/or exterior boundary. This section describes the many types of detectors available for intrusion detection systems.

Door Sensors

Door sensors enable the intrusion detection system to detect the status of a door (i.e. open or closed). When armed, the system can trigger an alarm when it detects that the door has been opened.

Vibration Detectors

Vibration detectors generate alarms when vibration is detected. These alarms are generated when vibration is outside of a specified range. They are typically used with materials having adequate vibration transmission properties (e.g. glass or metal).

Glass Break Detectors

Glass break detectors generate an alarm condition when triggered by one of the following conditions:

- Vibration
- Sound

Pressure Mats

Pressure mats are mats that generate an alarm condition by detecting an individual's weight when placed upon it.

Infrared Beam Interruption Detectors

Active infrared beam interruption detectors generate an alarm condition when a beam of infrared light between a transmitter and a receiver is interrupted. This detector provides line-of-sight detection between the transmitter and receiver. These units are most effective when concealed or camouflaged.

Movement Detectors

Movement detectors are devices used to detect movement of individuals, objects, or animals. Ultrasonic and microwave doppler detectors are designed to detect movement through the use of ultrasonic and microwave transmission. Passive infrared detectors are designed to sense changes in infrared energy in their field of view.

Microwave Detectors

Microwave detectors generate an alarm condition based on the reflection of radio waves from a moving object within a specified area.

Passive Infrared Detectors

Passive infrared (PIR) detectors generate an alarm condition or signal that a detection has occurred due to changes in the heat signature of objects within its field of view. Various patterns can be established including wide-angle, long-range, and high- or low-sensitivity.

Dual Technology Detectors

These detectors combine two technologies into a single detector (e.g. PIR and microwave detectors) for added security and to decrease the number of false alarms. See "Double Knock and Paired Sensors" on page 7-70.

Ultrasonic Detectors

Ultrasonic detectors generate an alarm condition based on the reflection of ultrasonic waves from a moving object. These detectors are sensitive to changes in the detection region. For example, the addition or removal of items such as boxes, cabinets, etc. may affect the device's detection properties.

Manually-Operated Devices

Manually-operated devices permit the user to trigger an alarm in the event of an emergency. These devices are typically located where an individual can trigger the device either overtly or covertly. An overtly operated device (e.g. a pull station in a hallway) can be triggered when the user is not in danger of physical harm from another individual (e.g. a medical emergency). A covertly operated device (e.g. a push button under a desk) can be triggered when the user is in direct danger of physical harm from another individual (e.g. during a bank robbery).

Keypad/Display Module

The keypad/display module is used for arming and disarming of the intrusion detection system.

Controller

The controller is the central processor of the intrusion detection system. It receives information from the detectors. When it receives a signal indicating an abnormal condition, it activates the reporting device.

Reporting / Annunciation

Reporting may be defined as the notification of an individual that an abnormal condition has been detected. An event, such as an attempted break-in, is typically annunciated locally (e.g. siren) and/or communicated to a remote location. An annunciated alarm alerts the occupants when an intruder has been detected, helps the responding authorities locate the facility, and helps scare away the intruder.

Paired Sensors or Double Knock Detection

To help prevent the occurrence of false alarms, intrusion detection systems can be configured to go into alarm when two sensors are triggered (paired sensors) or when a single sensor is triggered twice (double knock).

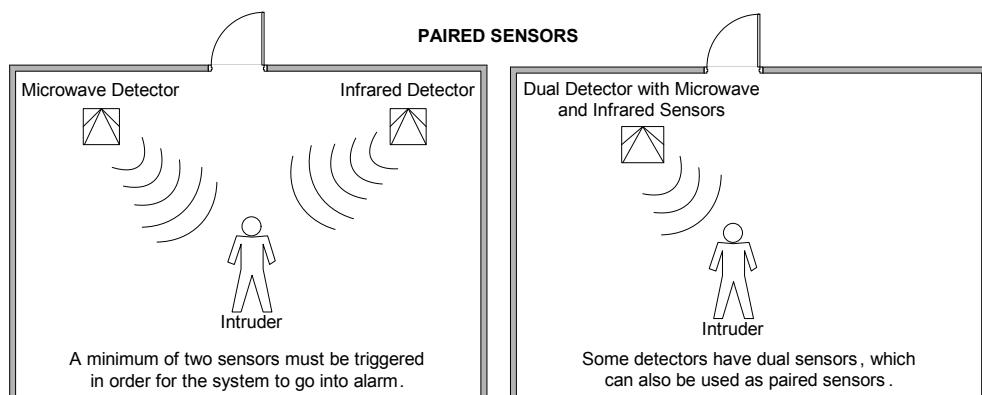


Figure 4-16: Paired Sensors

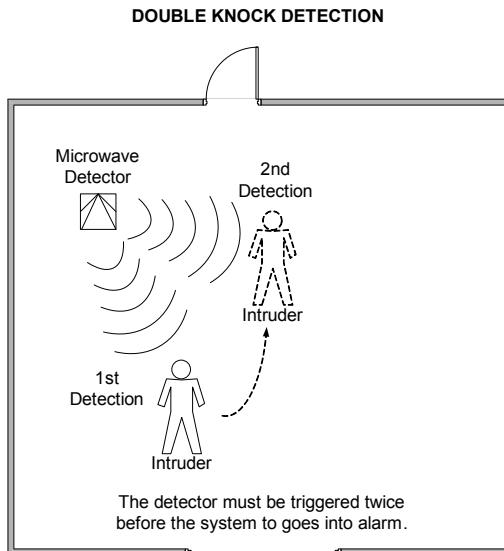


Figure 4-17: Double Knock

Zones

A zone consists of one or more sensors (inputs) connected to a controller. When a sensor detects an intrusion, the controller reports the zone (i.e. the location) of the intrusion based on the location of the detecting sensor.

The main benefits of zones are:

- Easier management of devices (groups of them)
- Easier localization of alarms (which zone)

Zones can be grouped together to form an area. See “Areas” on page 4-28.

See also Figure 4-18 for an example.

Areas

An area consists of a group of zones that are logically and conveniently combined to simplify the task of arming and disarming zones in a building. For example, when an area is armed, all of the zones in that area are armed. When the area is disarmed, all of the zones in that area are disarmed.

Areas can be assigned to large areas of a building, such as its perimeter, main entrance, or the entire floor.

See Figure 4-18 for an example.

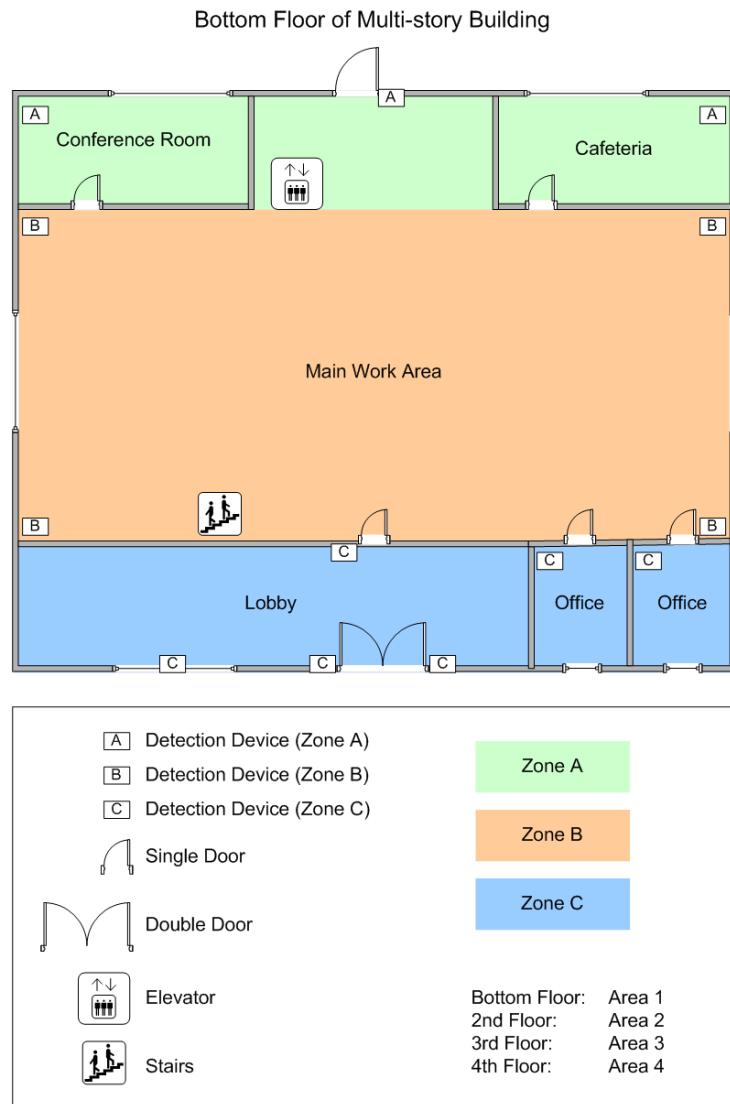


Figure 4-18: Zones and Areas

- The bottom floor of the multi-story building is divided into three zones (A, B and C).
- Each floor is divided into areas.
- Zone A has three sensors, Zone B has four sensors, and Zone C has six sensors.
- If a sensor detects an intrusion, the intrusion system reports the zone where that sensor is located.
- The intrusion system can be used to arm or disarm an area and bypass a zone.

Communicating Systems

In a communicating system, signals are usually indicated at the alarm site and are sent to a remote location. These systems are typically designed to report alarm signals to central station operators for subsequent verification and request for dispatch.

Signaling the Central Station

Signals from an alarm system are transmitted to the central station via telephone lines, radio communications equipment, or network communications methods (e.g. TCP/IP, wide area network - WAN, local area network - LAN, etc.).

About the Central Station

A central station is a secure location dedicated to the receipt of alarm signals. Central station operators monitor incoming signals from customer sites and dispatch the appropriate authorities, if necessary.

Proprietary Monitoring Facility

A proprietary monitoring facility is similar to a central station except the annunciator is located in a constantly manned guard room maintained by the property owner for his/her own internal security operations. The guards monitor the system and respond to all alarm signals and/or alert local law enforcement agencies.

Intrusion Signals

Alarm Signal

Alarm signals are generated when a intrusion detection sensor detects an intrusion.

Secure Signal

Secure signals are generated when a device or the system has returned to normal condition (has been reset by the customer after an alarm, or when an automatic “time-out” feature has reset the system after operating for a specific period of time). Secure signals do not necessarily indicate that an authorized individual has returned the device or system to normal operation. This signal could indicate that the intruder closed a door or window.

Abort or Cancel Signal

An “Abort” or “Cancel” signal cancels a response. If the alarm user disarms the system while it communicates with the central station, the signal will change to an “Abort” or “Cancel” signal.

Hardware

The hardware for intrusion detection systems typically consist of:

- One or more controllers
- One or more keypads
- One or more detectors and/or one or more sensors
- One or more warning and/or signaling devices
- The required power supply equipment
- A backup (standby) battery power supply

Supervised vs. Non-supervised Systems

A **Supervised** system is usually used in business/commercial or high security applications and requires employees to arm the system by a certain pre-designated time. It can also determine if employees disarm the system outside of pre-designated times. Individual codes can further be assigned so that each arming and disarming signal is logged to the individual employee or user.

A Supervised system is monitored at a remote central station or on-site.

A **Non-Supervised** system is usually used for home security systems. It does not require arming or disarming times, and there is no authority notification.

Arming, Disarming and Bypassing the Intrusion Detection System

An intrusion detection system can have the following three states: armed, disarmed, or bypassed.

- **Armed** – In this state, the system will report an intrusion if detected by a zone sensor. If the sensors detect an intrusion, the system can report it to a central station and trigger an annunciator (siren or bell). Other outputs can also be triggered, such as turning on lights in the facility. Zones and/or areas are armed when the protected zone/area is unoccupied and secured.
- **Disarmed** – In this state, the system will not report an intrusion if detected by a zone sensor. Zones and/or areas are disarmed when the protected zone/area is occupied by an authorized individual.
- **Bypassed** – In this state, the system will not report an intrusion if detected by a zone sensor. Zones are typically bypassed when the intrusion system is undergoing maintenance.

CK722 COMMISSIONING

This chapter outlines the process of commissioning the CK722 network controller for use in the P2000 Security Management System (SMS). Detailed instructions for installing and configuring the CK722 are divided into separate P2000 and CK722 documents, all of which will be referenced in this chapter.

INSTALLATION

The first step in commissioning the CK722 is installing the controller.

Installation consists of the following:

- Unpacking the equipment
- Mounting the equipment
- Wiring and powering the equipment

Besides the CK722 controller, other equipment can include field devices (e.g. RDR2S-A, RDR2S, SIO8) and peripheral devices (e.g. keypads, readers, input device and output devices).

Refer to the following documents for detailed hardware installation information:

- *CK722 Network Controller Hardware Installation Manual*
- *S300-DIN-RDR2S-A Hardware Installation Manual*
- *S300-DIN-RDR2S Hardware Installation Manual*
- *S300 Series S300-KDM Installation and Operation Manual*

STARTING A P2000 SCT PROJECT

This section describes how to properly start a P2000 SCT project, which is essential in streamlining the P2000 SCT configuration of a P2000 4.x project.

NOTE

This section does not take into account the installation of an Enterprise Solution or a Redundant P2000 4.x configuration.

The steps are organized into the following subsections, which include instructions that should be performed “once” for each item listed (e.g. project, site, CK722 controller, etc.):

- Once per Project (see page 5-2)
- Once per Site (see page 5-3)
- Once per CK722 (see page 5-5)
- Once per Hardware Module (see page 5-11)
- Once per Door (see page 5-12)
- Cursory Test (see page 5-14)

NOTE

The instructions in the following sections are designed to provide an overview of the steps needed to successfully start a P2000 SCT project. During this process, if you are an inexperienced P2000 user, you may need to reference other sections or documents for more detailed information or instructions. These sections and documents will be identified accordingly.

Once per Project

Before actually installing or configuring the P2000 SCT, do the following as part of your project preparation:

1. Create a naming convention.

Before starting any configuration, have a comprehensive naming convention in place. This affects all items that will be named (e.g. hardware and software items).

See “Create and Adhere to a Naming Convention” on page 5-15 for details about the naming convention for controller applications.

2. Create a list of partitions.

Figure out the partitioning scheme before starting any configuration. Refer to the *P2000AE Software User Manual* for more information on partitions.

3. Create a security flag scheme.

Security flags need to be consistent enterprise-wide. If you intend to use security flags, verify that their assignment and labels are in place before starting any configuration. Refer to the *P2000AE Software User Manual* for more information on security flags.

Once per Site

Perform the following actions for each site in a project:

1. **Networking:** Plan the Information Technology (IT) infrastructure. This includes performing the following actions:
 - Establish whether the controllers need to have peer-to-peer communication.
 - Decide whether nodes will use Dynamic Host Configuration Protocol (DHCP) or static Internet Protocol (IP) addresses.

Note

DHCP is a network protocol that automatically assigns IP addresses to devices on a DHCP-enabled network. CK722 controllers can be connected to this type of network to receive their dynamic IP addresses.

- Document the topology of the IT infrastructure.
2. **P2000:** Install and register the P2000 Version 4.1 software on all P2000 servers and workstations. Refer to the *P2000AE Server/Workstation Software Installation Manual*.
Then launch P2000 at the P2000 server. Refer to the *P2000AE Software User Manual*.
3. **P2000:** Copy any custom card formats (if necessary).
If your project requires card formats that are not included with the standard P2000 installation, copy the *.txt files associated with the custom card formats to the following directory on the P2000 server:
Local Disk:\Program Files\Johnson Controls\P2000\Identifier Formats
The card formats provided with the P2000 installation are included in the previous directory. See also “Appendix D: Identifier Formats”.
4. **P2000:** Import all identifier formats. Refer to the *P2000AE Software User Manual* for detailed instructions.
5. **P2000:** Define the P2000 Site Parameters, accessible from the System Configuration window. Create all required facility codes on the **Facility Code** tab. Refer to the *P2000AE Software User Manual*.
6. **P2000:** Add and configure all Security Flags (if required) from the System Configuration window. Refer to the *P2000AE Software User Manual*.
7. **P2000:** Add and configure all P2000 Time Zones. Refer to the *P2000AE Software User Manual*.
When adding a time zone, click **Yes** when asked whether you want to add it to all Legacy panels that do not have it.

8. **P2000 SCT:** If you wish to reuse existing custom P2000 SCT templates from other projects, copy their associated *.zip file to the following directory:
Local Disk:\Documents and Settings\Application Data\Johnson Controls\MetasysIII\DatabaseFiles
 For more information on templates, see “Chapter 6: JCI Standard Templates” and “Chapter 7: Creating Job-Specific Templates”.
9. **P2000 SCT:** If the P2000 SCT is already running, close it. Then launch the application.
10. **P2000 SCT:** When launching the P2000 SCT initially, select **Yes** when prompted to import all standard templates.

NOTE*This step is skipped when upgrading an existing project.*

To import custom templates at a later time, refer to the *P2000AE System Configuration Tool (SCT) Manual* for detailed instructions.

11. **P2000 SCT:** When prompted to create a Site object, select **Yes**. Create the Site object. Refer to the *P2000AE System Configuration Tool (SCT) Manual* for detailed instructions.

Note*This step is skipped when upgrading an existing project.*

12. **P2000 SCT:** When prompted to create a supervisory device, select **No**. These devices will be added in the “Once per CK722” section starting on page 5-5.
13. **P2000 SCT:** If you will reuse existing custom P2000 SCT templates from other projects, import them into the P2000 SCT database. Refer to the *P2000AE System Configuration Tool (SCT) Manual* for instructions on importing a template.
14. **P2000 SCT:** Copy and adapt Job-Specific Templates, as needed.

NOTE

Typically the Set 1 First Identifier Format attribute of any Access Control Object needs to be set to the desired card format at the site.

For information on copying a template, see “Copying Existing Templates as Job-Specific Templates” on page 7-2. For information on adapting a Job-Specific Template, see “Adapting the New Template According to the Job” on page 7-3.

Once per CK722

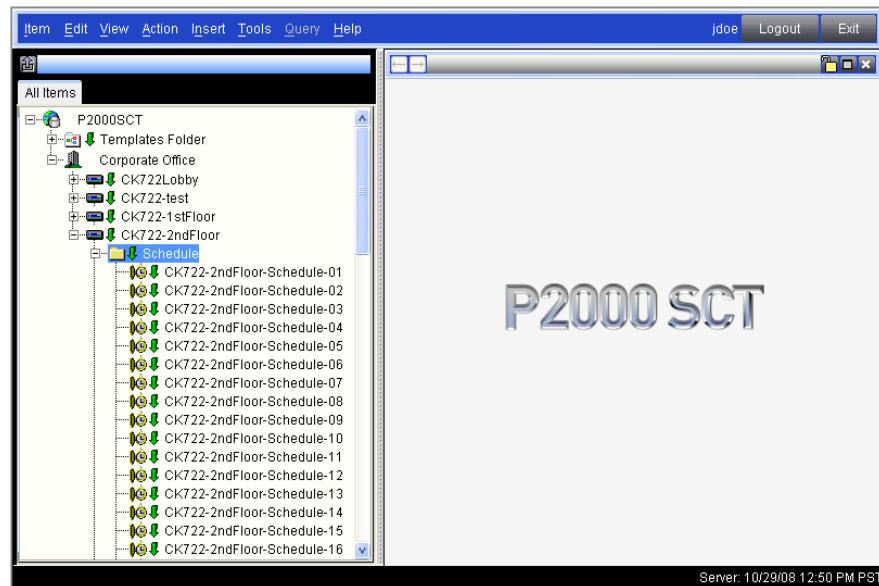
Perform the following actions for each CK722 controller in a site:

1. **P2000 SCT:** Add and configure the CK722 Device object.
 - The CK722 Device object's name should be the name that the IT department uses for this node.
 - Enter the P2000 Server's IP address into the **Database Server IP Address** attribute.
 - If assigning a static IP address to the controller, on the **Network** tab, deselect the **DHCP Enabled** check box, and enter the **IP Address** and **IP Mask** attributes.
 - Refer to the *P2000AE System Configuration Tool (SCT) Manual* for detailed information on adding and configuring a CK722 Device object.
2. **P2000 SCT:** Link Schedule objects to P2000 Time Zones. See page 5-5 for detailed instructions.
3. **CK722:** Verify the CK722 is correctly installed and configured. See "CK722 Installation and Configuration Verification" on page 5-7 for detailed instructions.
4. **P2000 SCT:** Download the object database to the CK722 for the first time. See "Downloading the P2000 SCT Object Database to a CK722 Controller for the First Time" on page 5-8 for detailed instructions.
5. **P2000:** Verify that the CK722 is online. See "Verifying Online Status of the CK722" on page 5-10 for detailed instructions.

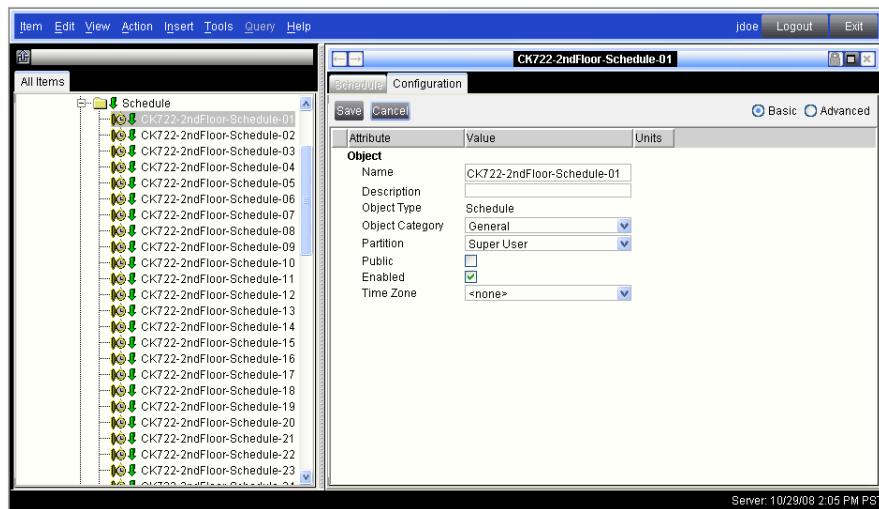
Linking Schedule Objects to P2000 Time Zones

► **To link Schedule objects to P2000 Time Zones:**

1. In the P2000 SCT Navigation Tree, select and expand the CK722 Device object. Then expand the **Schedule** folder.



2. Double-click on the first Schedule object (the object ending in **...-Schedule-01**).
3. Select the **Configuration** tab and click **Edit**.



4. In the **Time Zone** drop-down list, select the P2000 time zone you wish to link to the selected Schedule object.
5. Modify the **Name** attribute by replacing the **...-Schedule-XX** portion with something that identifies the P2000 Time Zone. Although not required, this step makes the Schedule object's names more organized.
6. Click **Save**.
7. Repeat the previous steps for each P2000 Time Zone that will be used by this CK722 controller.

CK722 Installation and Configuration Verification

► To verify the CK722 is installed and configured correctly:

1. Update the CK722 with the latest firmware and operating system. Refer to the *CK722 Network Utility Tool (NUT) Manual* for instructions. See also “Network Utility Tool (NUT)” on page 5-29.
2. Use the NUT to assign the CK722 its ultimate name. Although you can change the name later, assigning the name as early as possible is good practice.
3. If assigning the CK722 a static IP address (i.e. the CK722 is not connected to a DHCP-enabled network), follow the instructions in “Assigning a Static IP Address” starting on page 5-7.
4. From the P2000 server, ping the CK722 controller by its **IP address**.
5. From the P2000 server, ping the CK722 controller by its **name**.

Assigning a Static IP Address

This section describes how to assign a static IP address to a CK722 controller when the device is not connected to a DHCP-enabled network.

► To assign a static IP address:

1. Follow the instructions in “Using the CK722 Command Line Interface” starting on page 5-31 to connect a PC to the CK722 controller using an RS232 null modem cable and to access the CK722’s command line interface.
2. At the command prompt, enter the following command line according to the IP address you wish to assign and press <Enter>, substituting the text in brackets with the new static IP address, subnet mask, and gateway (optional).

Command structure:

```
netconfig -ip <addr> -mask <netmask> -gwy <gateway>
```

Example:

```
netconfig -ip 152.222.7.2 -mask 255.255.255.0 -gwy 152.222.7.4
```

NOTE

As an alternative, you may enter netconfig at the prompt and press <Enter> to enter the IP address, subnet mask, and gateway with separate command prompts, instead of entering them as a single command line.

3. At the prompt, enter reboot and press <Enter> to reboot the CK722 controller. Press “Y” on your keyboard to confirm.
4. Ensure that the CK722 has completely rebooted and indicates its current firmware version and name.

Downloading the P2000 SCT Object Database to a CK722 Controller for the First Time

The P2000 SCT's Download option downloads (copies) the object database from the P2000 SCT and overwrites the database of the selected CK722 controller. Perform a database download for the CK722 controller.

After you successfully download the object database to the controller, it will be renamed according to the name defined in the P2000 SCT.

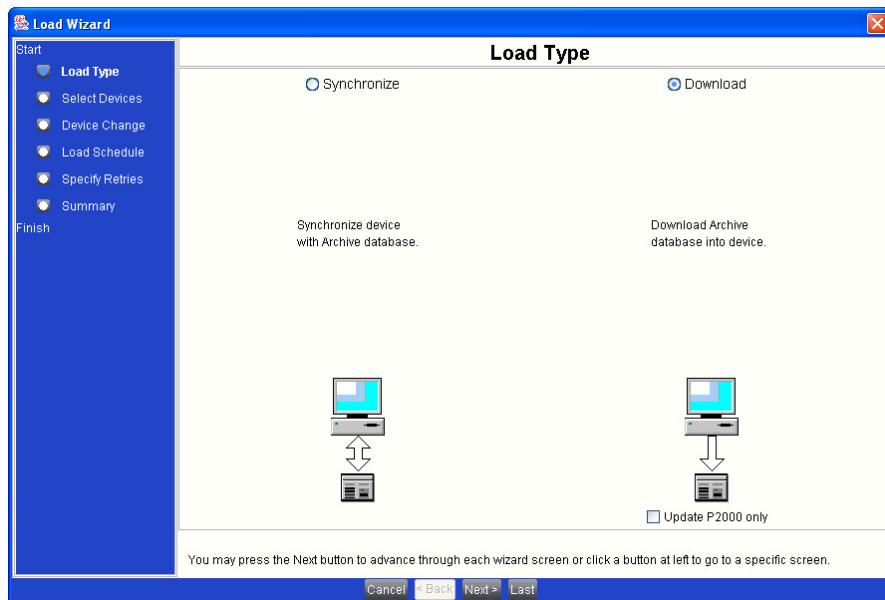


CAUTION Use only the **Download** feature when downloading the archive database to a CK722 controller for the first time. Do not use the Download feature on an already operational controller, as it will cause the controller to reset and will result in the loss of operational data. After performing a full initial download, use the **Synchronize** loading feature for all updates thereafter. The Synchronize feature downloads only the changes made to the SCT, instead of downloading the entire database. Also, the Synchronize feature does not interrupt the operation of the controller.

For additional information on downloading the P2000 SCT object database, or using the Synchronize feature, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

► **To download the P2000 SCT object database to the CK722 controller for the first time:**

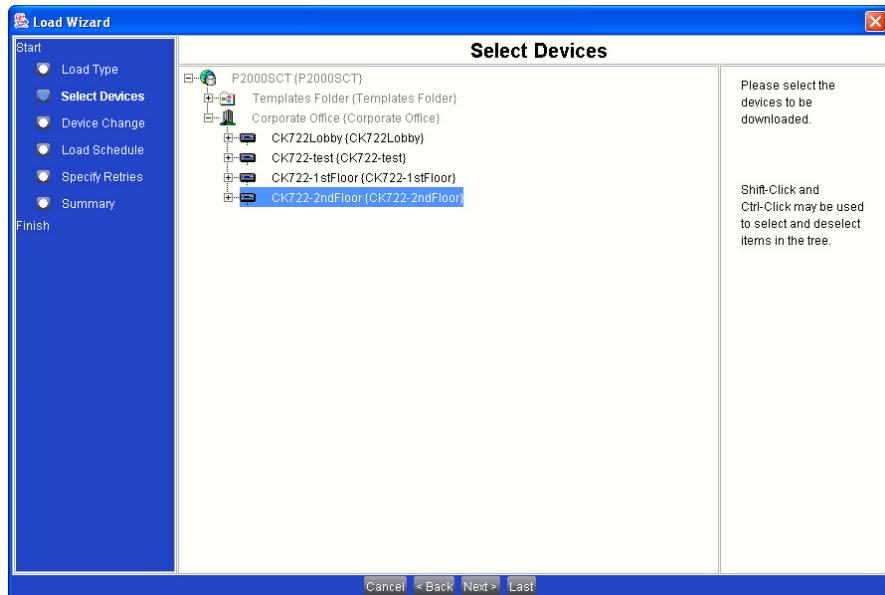
1. From the P2000 SCT, open the archive that will be used to download the P2000 SCT object database to the controller.
2. Select **Tools>Load Archive**. The Load Wizard appears.
3. On the Load Type screen, select the **Download** radio button and click **Next**.



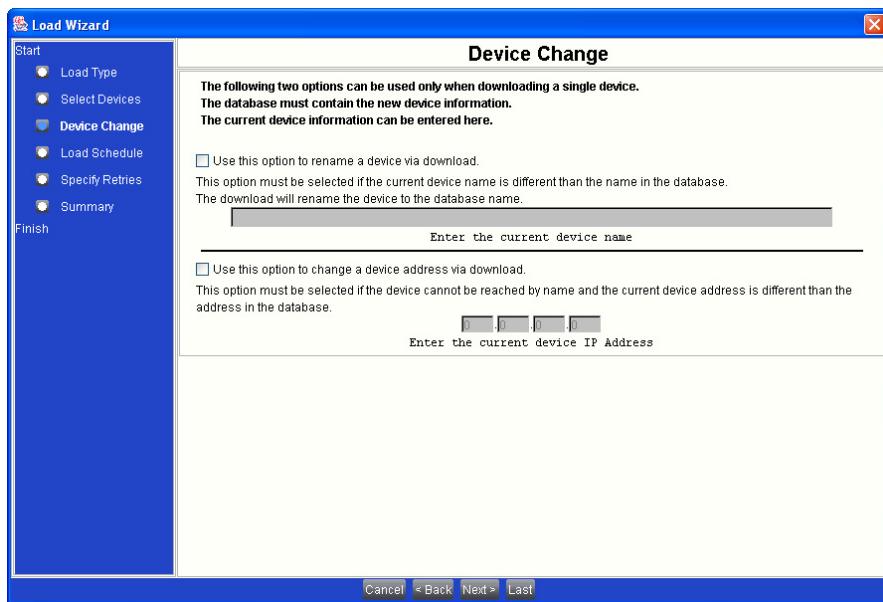
The download warning message appears.



4. Click **OK**.
5. On the Select Devices screen, select the CK722 controller that will receive the object database and click **Next**.



6. The Device Change screen enables you to download to a CK722 controller that only responds to a ping by its IP address but not by its name, or only by its name but not by its IP address.
If necessary, select the appropriate check box and provide the correct address or name information.



7. Click **Last**.
8. View the final information for the load and click **Finish**.
The ActionQ appears. See the *P2000AE System Configuration Tool (SCT) Manual* for information on monitoring the load using the ActionQ.

Verifying Online Status of the CK722

After downloading the P2000 SCT object database (see page 5-8), from the P2000, verify whether the CK722 controller is online.

► **To verify whether the CK722 is online:**

1. From the P2000 Main menu, select **System>System Status**.
2. On the System Status window, select **Panels** in the drop-down list in the upper-left corner of the window.
3. Select the **Network Panels** and **BACnet Panels** check boxes.
4. Verify that the CK722 is indicated as **Up** . Click **Refresh**, if necessary.
Refer to the *P2000AE Software User Manual* for more information on the System Status window.

Once per Hardware Module

When adding hardware modules, use one of following options.

Option 1: Creating a Hardware Module Using a Basic Hardware Module Template

Because some hardware modules, such as the RDR8S, can hold a variety of different door packages, it is more efficient to configure it first as an empty field device, and subsequently add its doors using x-Templates.

We recommend this option when using the Panel Tamper, Power Fail, and Battery Low input points for the hardware module(s) that will be added. Otherwise, consider Option 2 (see page 5-12).

► **To create a hardware module using a basic hardware module template:**

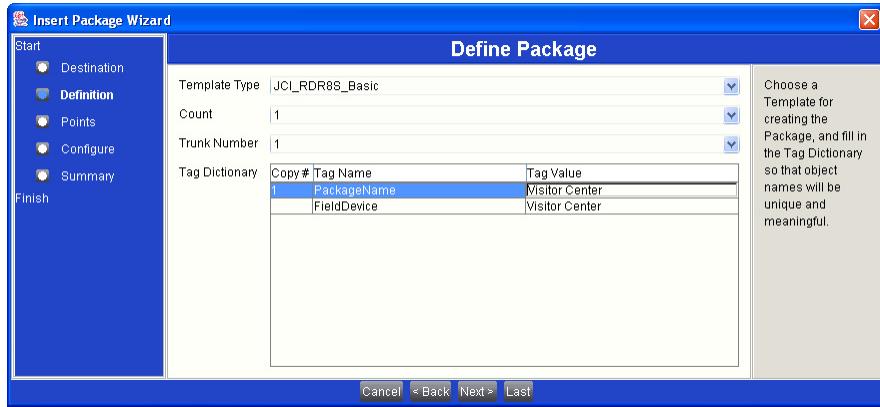
1. From the P2000 SCT menu bar, select **Insert>Package**. The Destination screen of the Insert Package Wizard appears.
2. Select the CK722 to which the hardware module(s) will be added and click **Next**. The Define Package screen appears.



3. In the **Template Type** drop-down list, select the desired Basic Hardware Module Template (e.g. JCI_RDR8S_Basic).
4. In the **Count** field, enter the number of hardware modules of this type that will be created on the same trunk.
5. In the **Trunk Number** field, select the trunk on which the hardware module(s) will reside.
6. In the **Tag Dictionary** table, enter a name for the **FieldDevice** tag value, which is the name that each hardware module will have.
7. Enter a name for the **PackageName** tag value for each hardware module being created.

The P2000 SCT does not let you assign the same name as the one assigned to the **FieldDevice** tag. As a workaround, add a space to the end of the

PackageName tag; however, do not add a space to the end of the **FieldDevice** tag.



8. Click **Next**.
9. On the Define Points screen, click **Next**.
10. On the Configure screen, Enter the **Hardware Module Number** attribute for each hardware module.
11. Click **Next**.
12. Click **Finish**.

For more information on inserting a package, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

Option 2: Creating a Hardware Module Manually

We recommend this option when the Panel Tamper, Power Fail, and Battery Low input points are *not* used for the hardware module(s) that will be added. Otherwise, consider Option 1 (see page 5-11).

For detailed instructions on inserting a hardware module manually, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

Once per Door

Perform one of the following sets of the instructions to create a package for each door, depending on whether the package's Template Type is an x-Template or a Hardware Module Template.

Creating a Package for a Door (x-Templates)

This section assumes that you are using a template that has been previously tested at least once. If not, create only one package at first to reduce the amount of potential rework.

► **To create a package for a door using an x-Template:**

1. From the P2000 SCT menu bar, select **Insert>Package**. The Destination screen of the Insert Package Wizard appears.
2. Select the CK722 to which the package will be added and click **Next**. The Define Package screen appears.
3. Select the **Template Type** and **Trunk Number**.
4. In the **Count** field, enter the number of doors of this type you want to create.
5. Enter a name for the **PackageName** tag value for each door being created.
6. Click **Next**.
7. Select the **S300 Hardware Module** and the **Connector** for each prompted field point.
8. Click **Next**.
9. Click **Last** and then click **Finish**.

For more information on inserting a package from an x-Template, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

Creating a Package for a Door (Hardware Module Templates)

Because the RDR2S and RDR2S-A only hold a limited variety of different door packages, it is more efficient to use the packages that already contain the field device (via a Hardware Module Template). In case free connectors remain on the field device, an additional door can be added via an x-Template.

This section assumes that you are using a template that has been previously tested at least once. If not, create only one package at first to reduce the amount of potential rework.

► **To create a package for a door using a Hardware Module Template:**

1. From the P2000 SCT menu bar, select **Insert>Package**. The Destination screen of the Insert Package Wizard appears.
2. Select the CK722 to which the package will be added and click **Next**. The Define Package screen appears.
3. Select the **Template Type** and **Trunk Number**.
4. In the **Count** field, enter the number of doors of this type you want to create.
5. Enter a name for the **PackageName** tag value for each door being created.
6. Enter a name for the **FieldDevice** tag value for each field device being created.
7. Click **Next**.
8. Click **Next**.
9. Enter all **Hardware Module Numbers** for each field device being created.
10. Click **Next** and then click **Finish**.

For more information on inserting a package from a Hardware Module Template, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

Cursory Test

You are now ready to perform a cursory test of the system.

► **To perform a cursory test of the system:**

1. **P2000 SCT:** Synchronize the P2000 SCT object database with the CK722. Refer to the *P2000AE System Configuration Tool (SCT) Manual* for more information.
2. **P2000:** Calibrate the input.
 - Refer to the *P2000AE Software User Manual* for instructions on calibrating the input.
 - Only calibrate the input once, unless the actual resistance or the connected field device changes.
 - Familiarize yourself with the rules of input calibration stated in the device's product documentation.
 - Verify correct calibration by monitoring the Real Time List.
3. **P2000:** Verify the system status for the following items is Up :
 - CK722 Controller (see “Verifying Online Status of the CK722” on page 5-10)
 - S300 Hardware Modules
 - Inputs
 - Outputs
 - Access Control Objects
 - Door Terminals
- For information on using the System Status window in P2000, refer to the *P2000 AE Software User Manual*.
4. Present a badge to every reader and verify that the badge information is correctly read.
5. **P2000:** Using the P2000 software, unlock each new door. Refer to the *P2000 AE Software User Manual*.
6. **P2000:** Verify that door status is correctly reported. Refer to the *P2000 AE Software User Manual*.
7. Verify that all control strategies work as soon as practically feasible.
Examples: Anti-Passback, Occupancy, Anti-Loitering, Executive Privilege, Overrides, etc.

BEST PRACTICES

This section provides a collection of best practices in populating the P2000 SCT database, as well as a brief description of how to accomplish common tasks.

Create and Adhere to a Naming Convention

No matter what elements of the P2000 SCT database need to be added, be it supervisory controllers, hardware modules, doors, elevators, intrusion zones, intrusion areas, stand-alone inputs and outputs, or any other object, it is imperative to have a comprehensive naming convention in place before starting any configuration.

This naming convention must take into account the entire scope of the project, such that whenever different sites are connected in an enterprise system, no name conflicts or confusion arise.

By using templates, all object names are constructed by appending the “role name” of each object contained in the template to the user-defined name of the application. The resulting name must be unique throughout the project.

All JCI Standard Templates guarantee a minimum of 16 characters for the user-defined portion of object names. In case 16 characters are not sufficient, the “role names” inside the templates may be shortened to 1 character, if necessary, to allow up to 30 characters for the user-defined part of the names.

Link Schedule Objects to Time Zones As Early As Possible

As soon as you add a supervisory controller to the P2000 SCT database, link the required amount of Schedule objects to P2000 Time Zones. A Time Zone is needed in a supervisory controller if it is associated with anyone’s access rights, or if it is used to control features based on the time of day, such as the overriding of doors, activation of outputs, or suppression of inputs.

See “Linking Schedule Objects to P2000 Time Zones” on page 5-5 for more information.

Create Job-Specific Templates

Unless the JCI Standard Templates are a perfect match for the project, we strongly recommend selecting the required templates and adapting them into Job-Specific Templates. Then, you can customize the templates, as needed, such as selecting the card format, determining the handling of Personal Identification Numbers (PINs), configuring the access and shunt times for doors, and adjusting any other attribute.

Using Job-Specific Templates is advantageous, since they will not be overwritten when a newer version of the P2000 SCT is installed.

See “Chapter 7: Creating Job-Specific Templates” for more information.

Test Job-Specific Templates Before Creating More Packages

After you modify a Job-Specific Template, we strongly recommend creating a single package from the template and verifying it operates correctly before creating more packages. If you use a Job-Specific Template as the basis for other Job-Specific Templates, any configuration errors or omissions are carried over to the new templates, which would have to be individually corrected.

See “Chapter 7: Creating Job-Specific Templates” for more information.

Correctly Map All Points in an x-Template

This is essential, as the P2000 SCT does not automatically map all “dedicated points” to the corresponding points on the same door interface of a hardware module.

There are three “dedicated points” on each door interface of an RDR2S, RDR2S-A, or RDR8S hardware module: the Door Contact, the Request to Exit (REX) input, and the Reader itself.

For example, if you map a Reader to connector **Reader 7 Data 0 / Data 1** on an RDR8S hardware module, you will need to map the Door Contact to connector **Reader 7 Door Contact** and the REX input to **Reader 7 REX** on the same hardware module.

Make Adjustments for Doors Without a Door Contact or REX Device

Some doors do not have a door contact or a REX device. Previously, as a common practice, installers would wire the door contact input at the field device as **closed** and leave the REX input device as **open**. The S300 Reader Terminal object allows you to turn off these features through the P2000 SCT software, so that the actual state of the input does not matter, and also allows you to use those inputs for completely different purposes, thus gaining one or two general purpose inputs per door (RDR2S-A and RDR8S only).

In the S300 Reader Terminal object, set the following attributes:

Attribute	Change Value to . . .	Note
<i>Portal Contact Connected</i>	Not Connected	when no door contact is present
<i>Aux Input Connected</i>	Not Connected	when no REX device is present

See also “How to “Steal” Unused RDR2S-A Field Points” on page 5-19.

How to Make Best Use of a Door Opening Device

Some doors may be equipped with a door opening device. The Access Control object supports driving a relay with a configurable delay after the door is unlocked. The driving of that relay can be turned on or off, and can also be made dependent on the person who requests access.

In the S300 Reader Terminal object, set the following attributes:

Attribute	Change Value to . . .
<i>ADA Relay Connected</i>	to Shunt Output
<i>ADA Relay Time</i>	the amount of time the relay should be active
<i>ADA Relay Delay</i>	the amount of time that activating the relay should be delayed

The door opening device can then be wired to the Shunt Output relay of an RDR2S-A or RDR8S hardware module.

See also “How to “Steal” Unused RDR2S-A Field Points” on page 5-19.

How to Make Best Use of a Door Contact of an Inactive Door

If the door contact of an inactive door needs to be individually reported, we recommend using a JCI Standard Template that supports the independent reporting of inactive doors (IAD).

Inactive doors can be reported in two ways, which you can define by setting the *Aux Mode* attribute of the inactive door’s Door Sequence object:

Attribute	Value	Result
<i>Aux Mode</i>	Shunt Only	Causes the inactive door only to report when it is actually opened. This is the default setting in the JCI Standard Templates.
<i>Aux Mode</i>	Shunt and Unlock Only	Causes the inactive door to also report as unlocked every time the active door is shunted.

How to Set Up a Card-In/Card-Out (CICO) Door

A Card-In/Card-Out (CICO) door has an entry reader and an exit reader. If your site uses CICO doors, use a JCI Standard Template especially designed for these types of doors. These templates do not require you to cross-wire the strike output of the master terminal into the REX input of the slave terminal. Furthermore, these templates offer full symmetrical support for anti-tailgating, timed overrides, and repeated access grants from either side of the door, while providing access decision feedback only to the side from which the access request was made. See the template’s description for details (“Chapter 6: JCI Standard Templates”).

How to Make Best Use of a Bond Sensor

The Door Sequence object's *Lock Monitor Attribute* allows the door to monitor a bond sensor or any other input that indicates the actual state of the lock. This state is typically indicated by a Security Supervised Input object, but it can also be the result of a logic equation for more complex applications. The Door Sequence object compares the actual state of the lock to the expected state, and raises an alarm in case of a discrepancy.

How to Use a “Catch-All” Card Format

An Access Control object makes an access decision as soon as an entity presents all required credentials. If an entity presents a card whose format is not recognized by any of the card formats defined for that door, the system ignores this input. For most doors you can add a “Catch-All” card format as the last card format that the Access Control object checks, thus guaranteeing to inform the P2000 about any foreign card presented, provided that the reader itself can read it. Set the following attributes of the Access Control object:

Attribute	Change Value to . . .
<i>Set 4 Enabled Default</i>	True
<i>Set 4 Mode</i>	Validation Only
<i>Set 4 First Identifier Format</i>	30000 – JCI Raw 128 Bit

How to Schedule Practically Anything

Schedule objects allow the changing of any writable attribute inside the controller based on P2000 Time Zones. The common access control functions frequently driven by a schedule are as follows:

- **Access Enabled** – via the *Access Enabled* attribute of an Access Control object
- **PIN Suppression** – via the *PIN Suppressed* attribute of an Access Control object
- **Door Override** – via the *Override* attribute of an Door Sequence object
- **Input Suppression** – via the *Suppress* attribute of a Security Supervised Input object
- **Outputs** – via the *Present Value* attribute of a Security Binary Output object

These functions are implemented by including the listed attributes in the **Scheduled Items** list of a Schedule object. As the *Suppress* attribute of a Security Supervised Input object and the *Present Value* attribute of a Security Binary Output Object are prioritized attributes, we recommend setting the *Priority for Writing* attribute of the Schedule object to **16** if the P2000 will need to control the suppression of an input or the activation of an output.

Scheduling other functions can be useful, such as the enabling/disabling of the following:

- **Identification Set 1** – via the *Set 1 Enabled* attribute of an Access Control object
- **Identification Set 2** – via the *Set 2 Enabled* attribute of an Access Control object
- **Identification Set 3** – via the *Set 3 Enabled* attribute of an Access Control object
- **Identification Set 4** – via the *Set 4 Enabled* attribute of an Access Control object

This allows the required identifiers to vary, based on the time of day.

Ultimately, all writable attributes of a simple data type (boolean, float, integer, enumeration) can be driven by a schedule.

NOTE

When scheduling an attribute defined as “Archiveable” by the letter “A” in the Notes column (refer to the associated object manual), an archive operation will archive the attribute at its current value.

How to “Steal” Unused RDR2S-A Field Points

One of the significant features of the RDR2S-A module is the ability to “steal” unused input and output points for use with other applications or devices. For example, many door applications do not require an output to drive a shunt device. If using an RDR2S module, when a reader is enabled, the shunt output for that reader is automatically assigned and cannot be used as a general purpose output. However, if using an RDR2S-A module, you can “steal” the shunt output (i.e. assign its connector to a Security Binary Output object in the P2000 SCT) and use it to activate another output device or for use in a different application altogether.

Keep in mind, however, that having this ability can cause issues with a site if managed improperly. Since you have the ability to assign an unused input or output, you also have the ability to erroneously assign a *used* input or output, thereby causing an application or device to not work as desired.

“Stealing” an input or output can be accomplished in different ways. To “steal” an output, you can simply select the output connector you wish to use, as described in the previous shunt output example. The following S300 Reader Terminal object attributes also enable you to “steal” an input or output:

Table 5-1: S300 Reader Terminal Object Attributes for “Stealing” Inputs and Outputs

Attribute	“Steals” Input or Output	Explanation
<i>Portal Contact Connected</i>	Input (Door Contact)	Deselect the check box to “steal” the input (i.e. disassociate the input from the door contact). The input used for the door contact associated with this reader can now be used for a different purpose.
<i>Aux Input Connected</i>	Input (REX)	Select Not Connected to “steal” the input (i.e. disassociate the input from the REX device). The input used for the REX associated with this reader can now be used for a different purpose.
<i>ADA Relay Connected</i>	Output (Shunt or Green LED)	For use with assisted access (ADA) applications. Select to Shunt Output or to Green Light Output to use the output for ADA purposes. See “Assisted Access” on page 7-46 for more information.

CREATING A CUSTOM LOGIC APPLICATION

The following objects are the most instrumental in creating custom logic applications:

- Controller Event Objects
- Multiple Command Objects
- Interlock Objects

This section focuses on the different behaviors and strengths of these objects, so that you can use them appropriately.

Synchronous vs. Asynchronous Operation

Understanding the differences between synchronous and asynchronous operation is important. In cases where immediate processing of actions is absolutely critical, synchronous operation is recommended. While asynchronous operations also preserve the order of any scheduled actions, they do allow other actions to be interleaved with those scheduled actions.

The example starting on page 5-21 illustrates the differences between synchronous and asynchronous operation. In this example, three different outputs need to be turned on as a result of an access decision.

Synchronous Operation

The Access Control object always invokes its Controller Events in a synchronous manner; that is, the Access Control object interrupts its access decision process to immediately handle Controller Events.

The Controller Event object also always invokes its action in a synchronous manner; that is, it interrupts its processing to immediately process the desired action (in our example, turning on an output).

Once all Controller Events are finished, the Access Control object invokes the Door Sequence object in a synchronous manner; that is, the Access Control object interrupts its current processing to immediately let the Door Sequence object carry out the access decision. Once the Door Sequence object is finished, the processing returns to the Access Control object to finish the access request.

Asynchronous Operation

The Multiple Command object operates in asynchronous mode by performing its actions in the order defined by the Multiple Command object's action table (these actions are placed at the end of the controller's action queue). Other objects and processes are also added to the end of the controller's action queue, which results in the microprocessor performing the actions from the queue in the order they were entered.

Advantages of Synchronous Operation

Synchronous operation guarantees that the programmed actions are handled immediately. This behavior allows the creation of custom logic, such as disabling access to other readers and placing doors into lockdown mode before access is granted, so that only one door of a defined set of doors is accessible at a time.

Constraints of Synchronous Operation

Synchronous operations can only be performed on objects that reside within the same controller as the calling object. Also, any synchronous operations have the responsibility of not holding up the processing of an object for too long, or even leading to infinite recursions. The majority of applications does not require synchronous operation. For example, there is no harm in allowing other actions to occur between unlocking a door and turning on the green light, as most actions inside the controller are handled within a few milliseconds.

Example

Figure 5-1 on page 5-22 illustrates the differences between synchronous and asynchronous operation by turning on three different outputs: BO1, BO2, and BO3.

In the synchronous operation (using Controller Events objects), the CK722 turns on all three outputs before unlocking the door. This operation stretches out the total time before the Access Control object unlocks the door and reports its access decision to the P2000 host.

In the asynchronous operation using a Multiple Command Object, the CK722 turns on all three outputs after the door is unlocked. In this case the Access Control object unlocks the door and reports its access decision earlier.

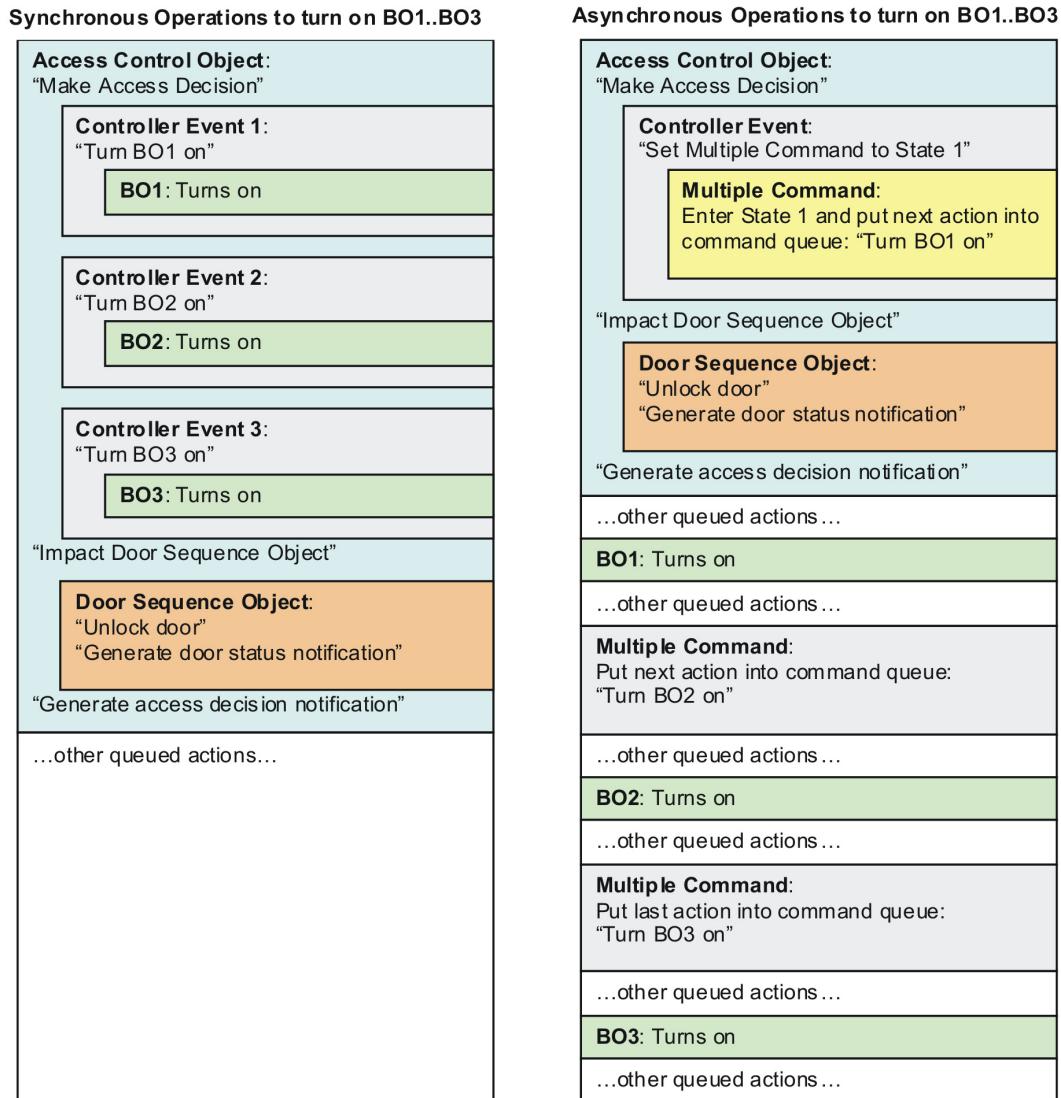


Figure 5-1: Synchronous vs. Asynchronous Operation

Controller Event

The Controller Event object is a simple yet powerful object used to create custom logic that can eliminate any race conditions – it is primarily designed to invoke a programmable operation in conjunction with an access request, or with the unlocking or locking of a door.

It also allows a targeted attribute to be toggled; the effect of the programmed action depends on the current state of the targeted attribute.

The Controller Event object invokes its action as a **synchronous** operation.

For more information, refer to the *Controller Event Object Manual*.

Multiple Command

The Multiple Command object is a versatile object used to create custom logic that can impact other objects by invoking their commands or writing their attributes.

It supports issuing prioritized writing of attributes, as well as issuing commands with programmable delays. It also allows issuing commands or writing attributes to objects on other controllers via the peer-to-peer communication feature.

With up to 32 states, the Multiple Command object is ideal to model state machines. A state machine, or finite state machine, is a model of behavior composed of a finite number of states, transitions between those states, and actions.

The Multiple Command object invokes its actions as **asynchronous** operations.

For more information, refer to the *Multiple Command Object Manual*.

Interlock

The Interlock object allows complex logic equations to be computed, and invokes actions based on the result of the logic equation. It allows input to the equation to be brought in from other controllers via the peer-to-peer communication feature.

Identical to the Multiple Command object, the Interlock object supports issuing prioritized writing of attributes, as well as issuing commands with programmable delays. It also allows issuing commands or writing attributes to objects on other controllers via the peer-to-peer communication feature.

The Interlock object invokes its actions as **asynchronous** operations.

The Interlock object is ideal to drive Multiple Command state machines based on occurring events.

For more information, refer to the *Interlock Object Manual*.

ATTRIBUTE PRIORITIZATION

Some objects have prioritized attributes. Attribute prioritization allows several applications to control an attribute while resolving any conflicts in a predictable and well-defined manner.

Each prioritized attribute has 16 independent priority slots to which the attribute's allowed values can be written (see Figure 5-2). A prioritized attribute always returns the value stored in the highest non-empty priority slot. If all 16 priority slots are empty, the attribute returns its default value, which may be stored in another attribute.

- Entering a value into a priority slot of a prioritized attribute is referred to as *issuing a prioritized write*.
- Removing a value from a priority slot of a prioritized attribute is referred to as *releasing a priority*.

Figure 5-2 shows an example of a prioritized attribute being controlled by several different applications. As each application uses a unique priority, all conflicts are settled.

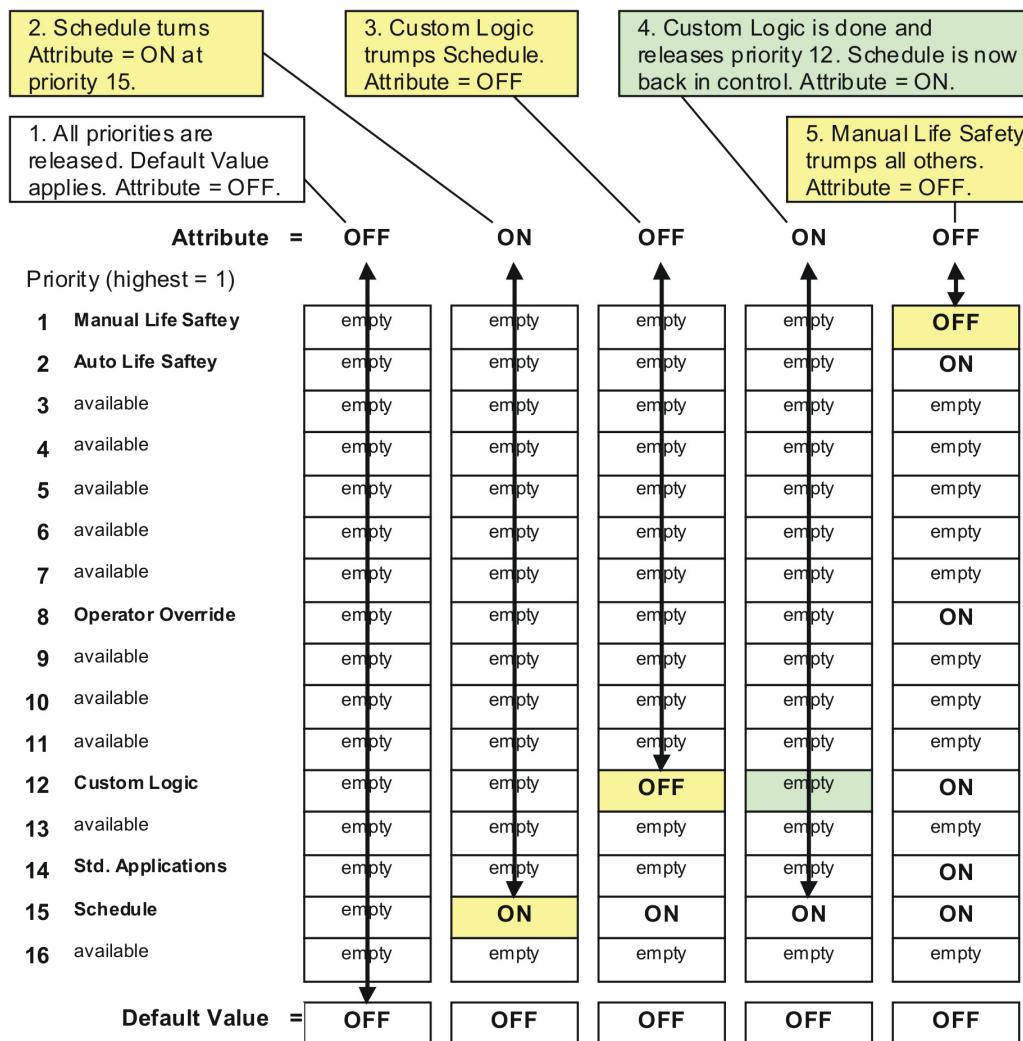


Figure 5-2: Prioritized Attribute Controlled by Several Different Applications

Prioritized attributes are identified by the letter "Z" in the **Notes** column of the object's attribute table in the respective object manual. Either the **Initial Value** or the **Value/Options/Range** column specifies which attribute is used to store the default value.

Examples of Prioritized Attributes

Table 5-2 provides examples of prioritized attributes and the attributes that store the corresponding default values.

Table 5-2: Examples of Prioritized Attributes

Object	Prioritized Attribute	Default Value Attribute	Allowed Values
Interlock	Present_Value	N/A (interlock equation governs)	False / True
Multiple Command	Present_Value	Relinquish_Default	1..32
Intrusion Area	Present_Value	Relinquish_Default	Disarm / Arm
Binary Output	Present_Value	Default_Value	Active / Inactive / Unknown
Door Sequence	Portal_Mode	Portal_Default	Normal / Override / Lockdown
Door Sequence	Suppress_Forced_Door	Suppress_Forced_Door_Default	False / True
Door Sequence	Suppress_Propped_Door	Suppress_Propped_Door_Default	False / True
Supervised Input	Suppress	Suppress_Default	False / True
Anti Loitering	Suppress	N/A (always False)	False / True

Recommended Priorities

To ensure that different applications resolve conflicts in a consistent manner, we recommend that certain features use the priorities defined in Table 5-3. Based on the needs of a particular application, the used priorities can be changed. As the priorities were initially named for applications for building automation, those names can also be sometimes found on the user interface.

Table 5-3: Recommended Priorities

Priority Value	Priority Name for Building Automation Applications	Priority Name for Security Applications	Objects Typically issuing Writes at the Specified Priority
1	Manual Life Safety	-	-
2	Auto Life Safety	-	-
3	Application	-	-
4	Application	-	-
5	Critical Equipment	-	-
6	Minimum On Off	-	-
7	Heavy Equipment Delay	-	-
8	Operator Override	-	-
9	Application	Door System Override	Door Sequence

Table 5-3: Recommended Priorities

Priority Value	Priority Name for Building Automation Applications	Priority Name for Security Applications	Objects Typically issuing Writes at the Specified Priority
10	Application	Door Security Mode	Door Sequence
11	Demand Limiting	Door Lockdown	Door Sequence
12	Application	General Custom Logic	Interlock, Multiple Command
13	Load Rolling	Door Override	Door Sequence
14	Application	Standard Applications	Elevator, Door Sequence
15	Scheduling	Scheduling	Schedule
16	Default	-	-

Writing and Releasing Prioritized Attributes

Many objects issue prioritized writes and release priorities when dealing with other objects. In some cases, users can configure the priorities; in other cases, they are fixed according to how they are implemented.

The objects in Table 5-4 currently issue prioritized writes and releases.

Table 5-4: Objects that Issue Prioritized Writes and Releases

Object	Prioritized Attribute to Write to ...	Priority	Default	Configurable through ...
Interlock	Any attribute in its Action Table	1..16	16	Action Table
Multiple Command	Any attribute in its Action Table	1..16	16	Action Table
Door Sequence	Own Portal_Mode for System Override	3..16	9	System_Override_Priority ¹
Door Sequence	Own Portal_Mode for Security Mode	3..16	10	Security_Mode_Priority ¹
Door Sequence	Own Portal_Mode for Lockdown	3..16	11	Lockdown_Priority ¹
Door Sequence	Own Portal_Mode for Override	3..16	13	Override_Priority ¹
Door Sequence	Any attribute referenced as output	14	14	N/A
Elevator	Any output type attribute in Floor_List	14	14	N/A

1. Attribute configurable in the Advanced section of the object's configuration view on the P2000 SCT.

The object in Table 5-5 currently issues prioritized writes, but has no possibility of releasing priorities.

Table 5-5: Object that Issues Prioritized Writes (No Releasing of Priorities)

Object	Prioritized Attribute to Write to . . .	Priority	Default	Configurable through . . .
Schedule	Any attribute in its Scheduled Items	9..16	15	Priority_For_Writing ¹

1. Attribute configurable in the Advanced section of the object's configuration view on the P2000 SCT.

Since the Schedule object cannot release its priority, any prioritized writes at a lower priority become ineffective; therefore, we recommend that Scheduling always uses the lowest priority of any application.

The recommended priority for Interlock and Multiple Command objects' Action Table is 12 (General Custom Logic), but the actual priority used depends on the requirement of the application.

Releasing Multiple Priorities

The objects in Table 5-6 also provide boolean attributes that facilitate the releasing of certain priorities.

Table 5-6: Releasing Multiple Priorities

For Object	Writing this Attribute to True	Releases Priorities	of Prioritized Attribute
Binary Output	Release_Present_Value	3..16	Present_Value
Door Sequence	Release_Portal_Mode	3..16	Portal_Mode
Door Sequence	Release_Suppress_Forced_Door	3..16	Suppress_Forced_Door
Door Sequence	Release_Suppress_Propped_Door	3..16	Suppress_Propped_Door
Supervised Input	Release_Suppress	3..16	Suppress
Anti-Loitering	Release_Suppress	3..16	Suppress

P2000 Host Priority

The P2000 currently issues all commands without priority, which is the equivalent to Priority 16. This means that all other controller applications can lock out the P2000's operator from modifying any of the prioritized attributes.

In cases where the P2000 operator should be allowed to adjust a prioritized attribute, all controlling applications should use priority 16, which results in a "last-one-wins" situation when controlling the attribute.

In cases where the P2000 operator should only be allowed to adjust a prioritized attribute when no internal application controls the attribute, all internal applications must release the attribute once they no longer want to control the attribute.

HARDWARE MODULE NUMBER GUIDELINES

The Hardware Module Number uniquely identifies a field device on a field bus, such as an S300 trunk. Hence, to avoid any addressing conflicts, verify that each field device on its field bus has a unique Hardware Module Number.

For field devices that use DIP switches to set their address, the Hardware Module Number equals the numeric value of those DIP switches.

NOTE

The KDM does not use DIP switches, as its Hardware Module Number is programmed through the keypad.

The S300 trunk supports up to 32 field devices; therefore, you can use Hardware Module Numbers ranging from 0 through 31 to uniquely address all field devices.

Since different field devices have different numbers of address-related DIP switches, some field devices cannot use the full range of available addresses. Nonetheless, an entire bus can be populated with 32 devices without any conflicts.

NOTE

The RDR2 field device has 3 DIP switches used to set its address, yielding Hardware Module Numbers ranging from 0 through 7. In addition, RDR2 field devices can have the same Hardware Module Number as legacy I/O field devices (S108, S18, I08, I16), since the S300 field bus protocol protects these groups of field devices from any address conflicts.

NETWORK UTILITY TOOL (NUT)

The NUT is a troubleshooting tool that allows you to restore functionality to the CK722 network controller in the event of device failure, or upgrade the controller's operating system, by deleting the data from the unit and re-installing the controller's operating system and firmware.



The Network Utility Tool should only be used on a new CK722 controller, if the CK722 is unresponsive and requires drastic measures to get the unit up and running, or when you need to update the controller's operating system. Do **not** use the Network Utility Tool to update the CK722 firmware if the unit is functioning properly and does not require an operating system update.

To update the firmware for a properly functioning controller, use the P2000 Host software's firmware update feature. See the *P2000AE Software User Manual* for details.

For more information, refer to the *Network Utility Tool (NUT) Manual*.

RECEIVING SNMP TRAPS

CK722 controllers support Simple Network Management Protocol (SNMP) Trap functionality. An SNMP Trap in the P2000 SMS is a notification of a system event or alarm associated with a CK722 controller that can be viewed on any computer that meets the requirements described in this section.

SNMP Traps are fully supported for SNMP Version 1 and 2c.

► **To receive SNMP traps for CK722 controllers:**

1. Verify that the computer to receive the SNMP Traps has a valid IP address by pinging it from another computer.
2. Install adequate third-party SNMP utility software on the computer receiving the SNMP traps. Use the software to configure the computer as an SNMP Trap recipient (refer to the software's documentation for assistance).
3. In the P2000 SCT, configure the CK722 Device object of each CK722 controller that will send SNMP Traps to the recipient computer. On the object's **Network** tab, configure the following attributes:
SNMP Enabled – Select the check box.
SNMP Management Device – Enter the IP address of the computer that will receive SNMP messages.
4. Synchronize the P2000SCT archive database with the CK722 controllers that will be used to transmit SNMP Traps to the recipient computer.

USING THE CK722 COMMAND LINE INTERFACE

The CK722 provides a command line interface via an RS-232 serial connection from the CK722 RS232C A port to the COM port of any computer running terminal emulation software, such as Tera Term Pro. This interface enables you to perform the following commands:

- **netconfig -ip <addr> -mask <netmask> -gwy <gateway>**
Enables you to manually set the controller's IP address, subnet mask, and gateway using a single command line. Use this command (or the **netconfig** command) if assigning a static IP address to the controller. See "Assigning a Static IP Address" on page 5-7 for more information.
- **netconfig**
Enables you to manually set the controller's IP address, subnet mask, and gateway with separate command prompts, instead of entering them as a single command line. Use this command (or the **netconfig -ip <addr> -mask <netmask> -gwy <gateway>** command) if assigning a static IP address to the controller. See "Assigning a Static IP Address" on page 5-7 for more information.
- **chpasswd**
Enables you to change the Metasys password used by technicians to debug the controller.
- **ping <addr | hostname>**
Enables you to ping devices on the network. You may ping by IP address or host name. For more information on this type of command, refer to the Microsoft MS-DOS documentation.
- **tracert <addr | hostname>**
Enables you to determine the route taken by packets across the network. For more information on this type of command, refer to the Microsoft MS-DOS documentation.
- **ver**
Enables you to determine the current firmware version, IP address, subnet mask, and gateway of the CK722 controller.
- **help**
Lists the available commands.
- **?**
Lists the available commands.
- **reboot**
Enables you to restart the CK722 controller.

To use the CK722 command line interface, the following hardware and software are required:

- Windows®-compatible PC
- Terminal Emulation Software (Tera Term Pro Preferred) – To use the free Tera Term Pro software, download and install the latest version from the following site:
<http://hp.vector.co.jp/authors/VA002416/teraterm.html>
- RS-232 Null Modem Cable (DB9F/F) – Black Box EYN257T-0015-FF or equivalent

The null modem cable provides an RS-232 serial connection from the CK722 RS232C A port to the COM port of the computer running the terminal emulation software, such as Tera Term Pro.

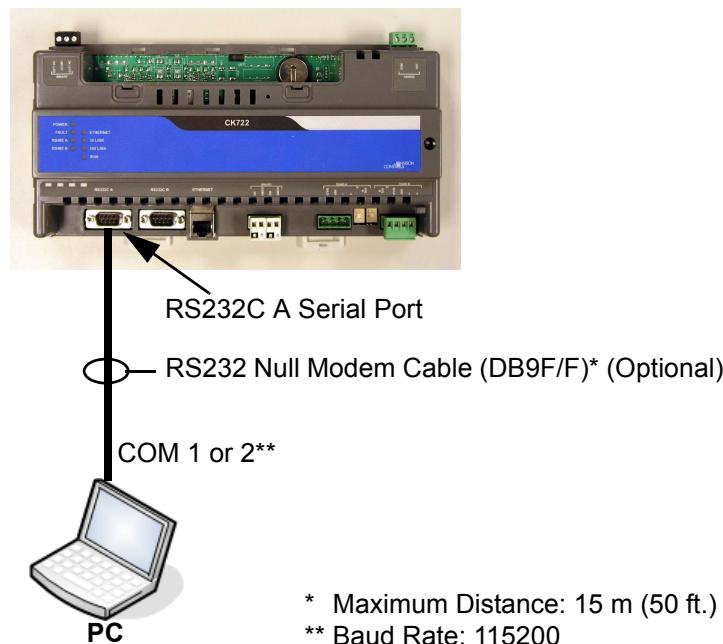
Connector 1 Pinout	Connector 2 Pinout
3 TX	2 RX
2 RX	3 TX
7 RTS	8 CTS
8 CTS	7 RTS
5 SG	5 SG
6 DSR	4 DTR
4 DTR	6 DSR

TX = Transmit
 RX = Receive
 RTS = Request to Send
 CTS = Clear to Send
 SG = Signal Ground
 DSR = Data Set Ready
 DTR = Data Terminal Ready

► To use the CK722 Command Line Interface:

1. Connect the RS232 Null Modem Cable to the PC's **COM 1** or **COM 2** port. Connect the other end of the cable to the CK722's **RS232C A** port. See Figure 5-3.

This connection enables you to use your PC as a CK722 terminal.

CK722 Controller*Figure 5-3: PC to CK722 Controller Connections*

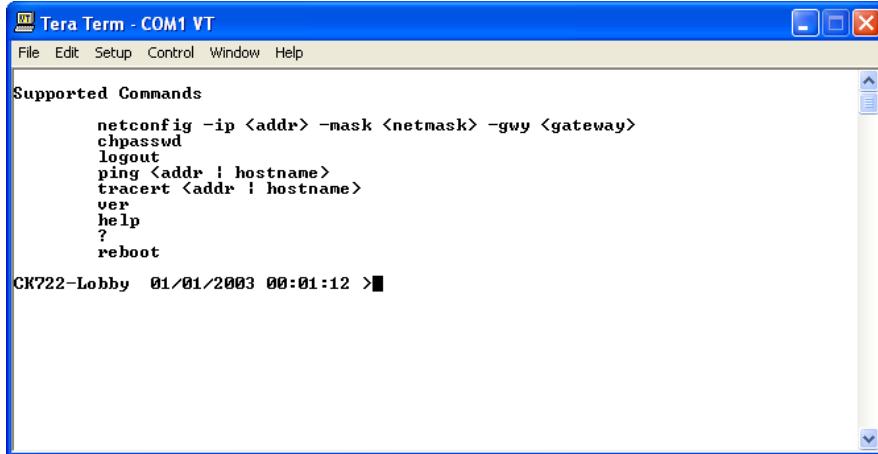
2. Launch Tera Term Pro on the PC.

NOTE

The Tera Term Pro terminal emulation software was used in the development of this section. Use of other terminal emulation programs may differ.

3. From the menu bar, select **Setup>Serial Port**.
4. Select a COM port from the **Port** drop-down list, depending on which COM port you are using.
5. Select **115200** from the **Baud rate** drop-down list.
6. Click **OK**.

7. Press <Enter> on your PC's keyboard. A list of supported commands appears.



The screenshot shows a Windows application window titled "Tera Term - COM1 VT". The menu bar includes File, Edit, Setup, Control, Window, and Help. The main window displays a list of "Supported Commands" in a monospaced font. The commands listed are: netconfig -ip <addr> -mask <netmask> -gwy <gateway>, chpasswd, logout, ping <addr | hostname>, tracert <addr | hostname>, ver, help, ?, and reboot. At the bottom of the window, the text "CK722-Lobby 01/01/2003 00:01:12 >" is visible, indicating the current date and time.

8. At the prompt, enter the desired command and press <Enter>.
9. Follow any additional prompts, as needed.

WRITING THE CK722 DATABASE TO FLASH MEMORY

Legacy controllers in a P2000 Security Management System, such as the CK721, CK720, and CK705, require P2000 operators to manually perform a **Controller Write DB to Flash** command from the P2000 host software, or configure a P2000 event to automatically send the command at defined time periods, during the initial system configuration and periodically thereafter as major changes occur to the system. The CK722 controller, however, automatically writes the database to flash memory:

- Every day between midnight and 1:00 AM
- As a result of major changes to the database that require operators to perform a full download from the P2000 SCT or P2000 host

During the write-to-flash process, which runs in the background, all applications continue working and the controller will not reset or restart.

If the CK722 is restarted, it reverts back to its last archived database, which typically is the database archived at the beginning of the day between midnight and 1:00 AM.

JCI STANDARD TEMPLATES

This chapter provides detailed information on each of the JCI Standard Templates provided with the P2000 SCT, and describes how to use them to build customized applications. You can use the JCI Standard Templates as a starting point for your own access control and/or intrusion detection application.

For instructions on how to create templates and load packages, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

ABOUT P2000 SCT TEMPLATES

Templates are “rubber stamps” of pre-defined applications that can be used to rapidly populate the P2000 SCT hardware configuration database. Templates are used to create *packages*, which contain all of the components for a single application, such as a door.

JCI Standard Templates are delivered with the P2000 SCT installation and are intended to be a starting point to create Job-Specific Templates, which are more closely adapted to the job-specific requirements. These Job-Specific Templates are then used to populate the P2000 SCT hardware configuration database.

Templates are used to define security logic functions using the P2000 SCT graphics tool, and a package is simply an instance of a particular template. Applying templates enables you to streamline the object creation and logic definition process by assigning pre-defined security functions to CK722 controllers.

Template Types

There are two basic types of templates: Hardware Module Templates and x-Templates.

- **Hardware Module Templates** contain one or more hardware modules. This is beneficial when the contained hardware modules are essentially entirely dedicated to the application, such as a 2-door hardware module in a Card-In-Card-Out door application, or a fully loaded Input/Output (I/O) board in an elevator or intrusion application. Hardware Module Templates therefore save the step of creating a hardware module manually.
- **x-Templates** do not contain any hardware modules. This is beneficial when the application shares the hardware module with other applications, such as a

single door on an 8-door hardware module, or a Contact-Only door on a still available input on an existing hardware module. x-Templates therefore allow a wide variety of doors to be added to a single hardware module.

JCI Standard Template Naming Convention

The names of all JCI Standard Templates (e.g. JCI_RDR2SA_Basic) are divided into the following parts, each separated by underscores for easier readability:

- **Job Prefix**
Each JCI Standard Template uses the job prefix **JCI**.
- **Hardware Descriptor**
Identifies the type of S300 hardware module used in the template. Current hardware descriptors are: I16, IO8, SI8, SIO8, RDR2S, RDR2SA, RDR8S, KDM, and x. The x designates x-Templates, which allow you to assign a door to an existing hardware module when loading a package.
- **Application Name**
Identifies the actual name of the application. Examples are:
 - **Full-IO** – Adds the entire set of available inputs and outputs to the P2000 SCT database.
 - **Basic** – Adds only a few device-specific inputs, such as Panel Tamper, Power Fail, and Battery Low.

Examples of door-related application names are **Card-In-Card-Out (CICO)** for Card-In-Card-Out doors, **Card-In** for Card-In doors, and fairly involved application names such as **CICO_IAD_TAMP** for a Card-In-Card-Out door with an independently reporting Inactive Door and Tamper switches on both readers.

NOTE

Template names cannot exceed 32 characters.

Identifier Format for Access Control Objects

All JCI Standard Door Templates use the **10000 - Default** identifier format as their default card format, which is identical to the **30000 - JCI Raw 128 Bit** format. The **10000 - Default** identifier format enables the Access Control object to process the information from any card type presented to the reader (as long as the reader itself can read the card) and make an access decision based on the first 128 bits of data received from the reader.

Card-In-Card-Out Doors

All JCI Standard Templates for Card-In-Card-Out (CICO) doors use the **Local Feedback** feature, which indicates the access decision (e.g. access granted or denied) only on the side of the door where the entity requested access. Feedback not associated with a particular direction, such as override and lockdown, as well as commands from the P2000, are indicated on both sides of the door.

Copying and Modifying Existing Templates

Security applications can differ dramatically between sites, so the templates provided with the P2000 SCT may not meet the needs of a particular site. For this reason, templates can be loaded as is (i.e. exactly as they have been defined) or can be copied and then edited for specific application needs. The latter option allows you to customize your own template without having to start from scratch.

Before creating a template from scratch, check whether a template provided with the P2000 SCT closely matches the security function/application you wish to develop.

See “Chapter 7: Creating Job-Specific Templates” for information on copying and editing templates.

TEMPLATE INFORMATION FRAMEWORK

Each JCI Standard Template provided with the P2000 SCT is described in detail in this chapter. The information for each template is separated into the following sections:

Template Introduction/Description – A brief introduction and basic description of the template.

Object Diagram – An image of the template diagram as it appears in the P2000 SCT.

NOTE

The layout of the diagram in the software and as it appears in this manual may differ slightly.

Graphical Representation – An illustration of how the template can be applied and how the objects relate to the hardware.

Object List and Description – A detailed description of each object in the template.

Non-Default Attributes – Attribute values that are different than the factory default attribute values.

Assumptions – Information associated with the template’s objects that may not be visibly apparent.

NOTE

Certain high-level objects, such as S300 Trunk objects or S300 Hardware Module objects, although included as part of a template, are excluded from the diagram.

NOTE

The previously described framework categories that are not applicable or that would simply duplicate other information in other categories have been removed.

JCI STANDARD TEMPLATES

The JCI Standard Templates can be divided into the following categories:

- Full-IO Hardware Module Templates (see page 6-4)
- Basic Hardware Module Templates (see page 6-21)
- Door Hardware Module Templates (see page 6-24)
- Door x-Templates (see page 6-34)
- Miscellaneous Templates (see page 6-58)
- Legacy Templates (see page 6-63)

Full-IO Hardware Module Templates

Use the following templates under the following conditions:

- When the inputs and outputs residing on the hardware modules are used in **elevator** or **intrusion** applications
- If you use the majority of the inputs and outputs created, since the number of allowed inputs and outputs on a controller is limited

Also, once you finalize the application, delete all unused input and output points.

NOTE

In the Full-IO Hardware Module Templates, many Security Supervised Input objects are set, by default, to suppress the associated input's alarm state, which is useful in most Elevator and Intrusion applications. For example, typical Elevator applications do not require notification of an input's alarm state every time someone presses an elevator button to access a particular floor. In Intrusion applications, the notification of inputs that go into alarm are handled by the associated Intrusion Zone object. In Elevator applications, the notification of inputs that go into alarm are handled by the associated Elevator object.

JCI_I16_Full-IO

This template adds objects for an S300-I16 hardware module with 16 suppressed inputs.

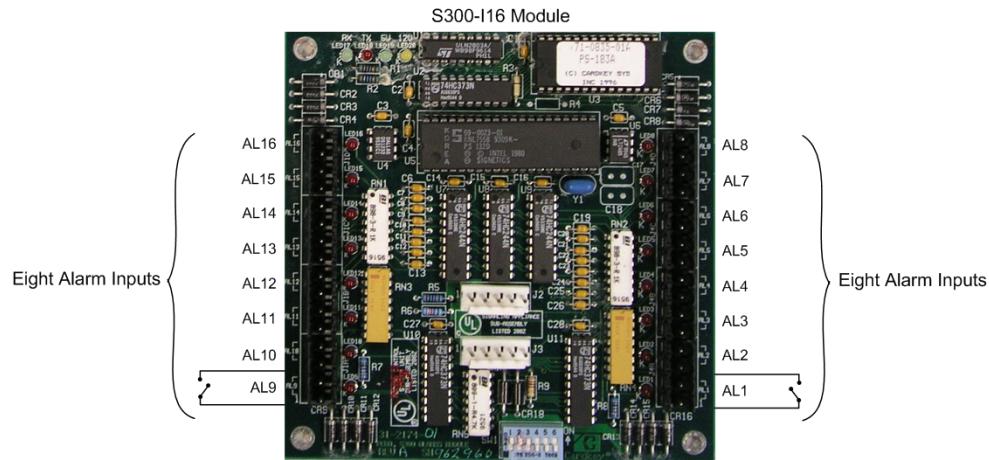
NOTE

*All inputs with the S300-I16 module are **unsupervised**, even though the object name refers to each input as a Security **Supervised Input**.*

Object Diagram

Figure 6-1: Object Diagram for JCI_I16_Full-IO Template

Graphical Representation



The S300-I16 module is represented by an S300 Hardware Module object in the P2000 SCT.

Each alarm input is represented by a Security Supervised Input object in the P2000 SCT.

Figure 6-2: Graphical Representation of JCI_I16_Full-IO Template

Object List and Description

Table 6-1: Object List for JCI_I16_Full-IO Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the S300-I16 module.
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8 {FieldDevice} AL9 {FieldDevice} AL10 {FieldDevice} AL11 {FieldDevice} AL12 {FieldDevice} AL13 {FieldDevice} AL14 {FieldDevice} AL15 {FieldDevice} AL16	General purpose inputs. Connectors: AL1 is the connector for {FieldDevice} AL1, AL2 is the connector for {FieldDevice} AL2, and so on. Each input device is wired to one of the AL inputs on the S300-I16 module.

Non-Default Attributes

Table 6-2: Non-Default Attributes for the JCI_I16_Full-IO Template

Object	Attribute	Non-Default Value
{FieldDevice} AL1 through {FieldDevice} AL16	Suppress Default	Selected

JCI_IO8_Full-IO

This template adds objects for an S300-IO8 hardware module with 8 suppressed inputs and 8 outputs.

NOTE

All inputs with the S300-IO8 module are unsupervised, even though the object name refers to each input as a Security Supervised Input.

Object Diagram

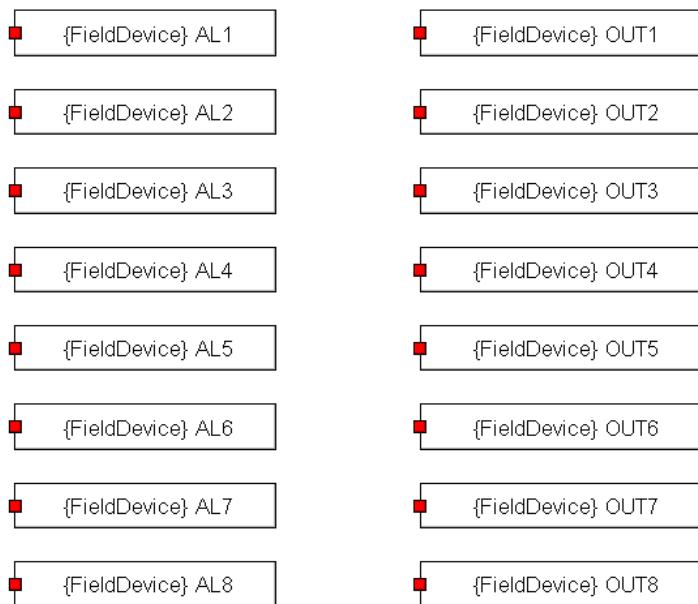
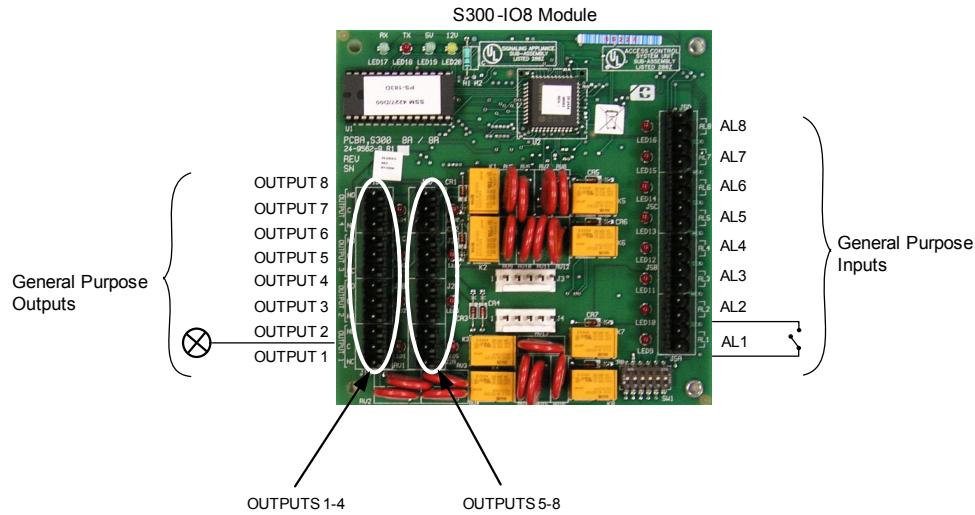


Figure 6-3: Object Diagram for JCI_IO8_Full-IO Template

Graphical Representation



The S300-IO8 module is represented by an S 300 Hardware Module object in the P2000 SCT.

Each general purpose input is represented by a Security Supervised Input object in the P2000 SCT.

Each general purpose output is represented by a Security Binary Output object in the P2000 SCT.

Figure 6-4: Graphical Representation of JCI_IO8_Full-IO Template

Object List and Description

Table 6-3: Object List for JCI_IO8_Full-IO Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the S300-IO8 module.
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8	General purpose inputs. Connectors: AL1 is the connector for {FieldDevice} AL1, AL2 is the connector for {FieldDevice} AL2, and so on. Each input device is wired to one of the AL inputs on the S300-IO8 module.

Table 6-3: Object List for JCI_IO8_Full-IO Template

Type	Name	Description
Security Binary Output	{FieldDevice} OUT1 {FieldDevice} OUT2 {FieldDevice} OUT3 {FieldDevice} OUT4 {FieldDevice} OUT5 {FieldDevice} OUT6 {FieldDevice} OUT7 {FieldDevice} OUT8	General purpose outputs. Connectors: OUTPUT 1 is the connector for {FieldDevice} OUT1, OUTPUT 2 is the connector for {FieldDevice} OUT2, and so on. Each output device is wired to one of the OUTPUT connectors on the IO8 module.

*Non-Default Attributes**Table 6-4: Non-Default Attributes for the JCI_IO8_Full-IO Template*

Object	Attribute	Non-Default Value
{FieldDevice} AL1 through {FieldDevice} AL8	Suppress Default	Selected

JCI_SI8_Full-IO

This template adds objects for an S300-SI8 hardware module with 8 suppressed inputs.

Object Diagram

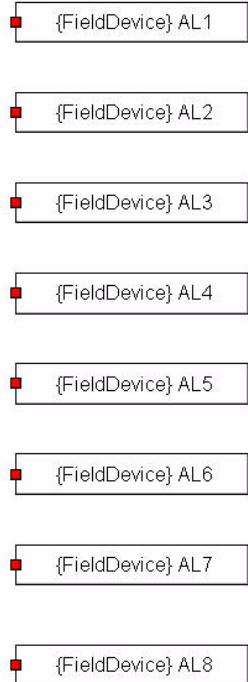
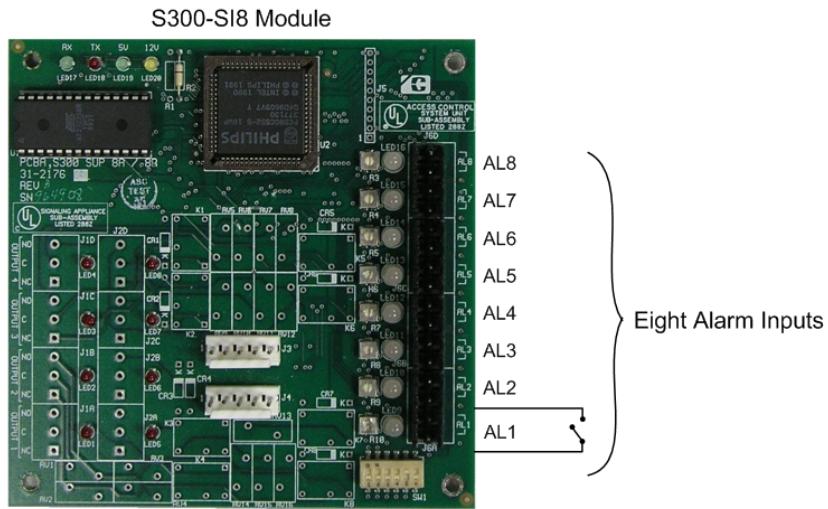


Figure 6-5: Object Diagram for the JCI_SI8_Full-IO Template

Graphical Representation



The S300-SI8 module is represented by an S300 Hardware Module object in the P2000 SCT.

Each alarm input is represented by a Security Supervised Input object in the P2000 SCT.

Figure 6-6: Graphical Representation of the JCI_SI8_Full-IO Template

*Object List and Description**Table 6-5: Object List for the JCI_SI8_Full-IO Template*

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the SI8 module.
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8	General purpose inputs. Connectors: AL1, AL2, AL3, AL4, AL5, AL6, AL7, and AL8 respectively. Each input device is wired to one of the AL inputs on the SI8 module.

*Non-Default Attributes**Table 6-6: Non-Default Attributes for the JCI_SI8_Full-IO Template*

Object	Attribute	Non-Default Value
{FieldDevice} AL1 through {FieldDevice} AL8	Suppress Default	Selected

JCI_SIO8_Full-IO

This template adds objects for an S300-SIO8 hardware module with 8 suppressed inputs and 8 outputs.

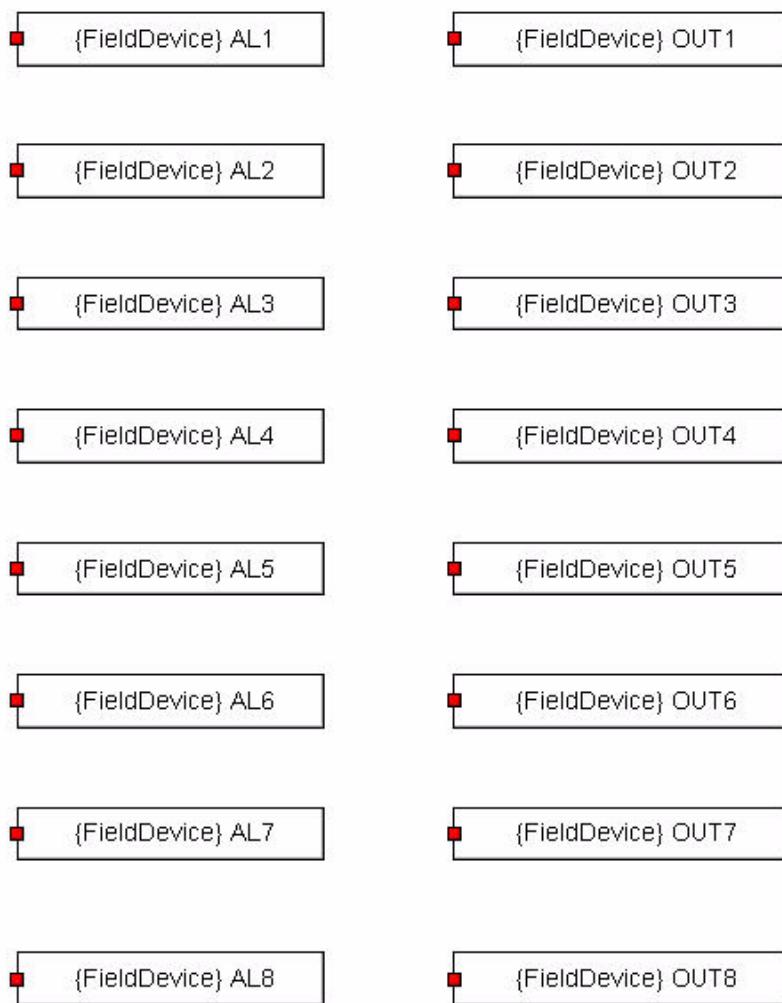
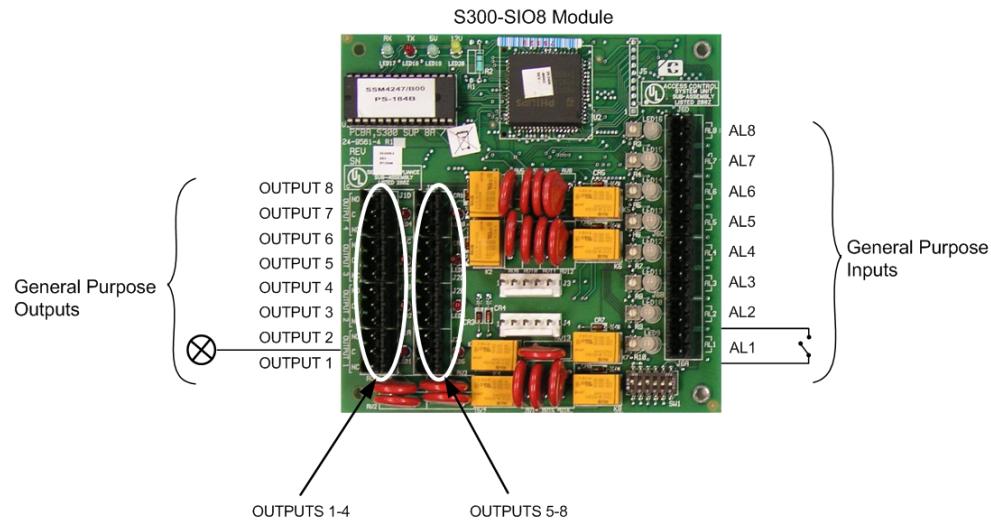
Object Diagram

Figure 6-7: Object Diagram for the JCI_SIO8_Full-IO Template

Graphical Representation

The S300-SIO8 module is represented by an S300 Hardware Module object in the P2000 SCT.

Each general purpose input is represented by a Security Supervised Input object in the P2000 SCT.

Each general purpose output is represented by a Security Binary Output object in the P2000 SCT.

Figure 6-8: Graphical Representation of the JCI_SIO8_Full-IO Template

Object List and Description

Table 6-7: Object List for the JCI_SIO8_Full-IO Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the SIO8 module.
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8	General purpose inputs. Connectors: AL1, AL2, AL3, AL4, AL5, AL6, AL7, and AL8 respectively. Each input device is wired to one of the AL inputs on the SIO8 module.

Table 6-7: Object List for the JCI_SIO8_Full-IO Template

Type	Name	Description
Security Binary Output	{FieldDevice} OUT1 {FieldDevice} OUT2 {FieldDevice} OUT3 {FieldDevice} OUT4 {FieldDevice} OUT5 {FieldDevice} OUT6 {FieldDevice} OUT7 {FieldDevice} OUT8	General purpose outputs. Connectors: OUTPUT 1 is the connector for {FieldDevice} OUT1, OUTPUT 2 is the connector for {FieldDevice} OUT2, and so on. Each output device is wired to one of the OUTPUT connectors on the SIO8 module.

*Non-Default Attributes**Table 6-8: Non-Default Attributes for the JCI_SIO8_Full-IO Template*

Object	Attribute	Non-Default Value
{FieldDevice} AL1 through {FieldDevice} AL8	Suppress Default	Selected

JCI_RDR2SA_Full-IO

This template adds objects for an RDR2S-A hardware module with 3 device-related inputs (Panel Tamper, Power Fail, and Battery Low), 8 suppressed inputs, and 8 outputs.

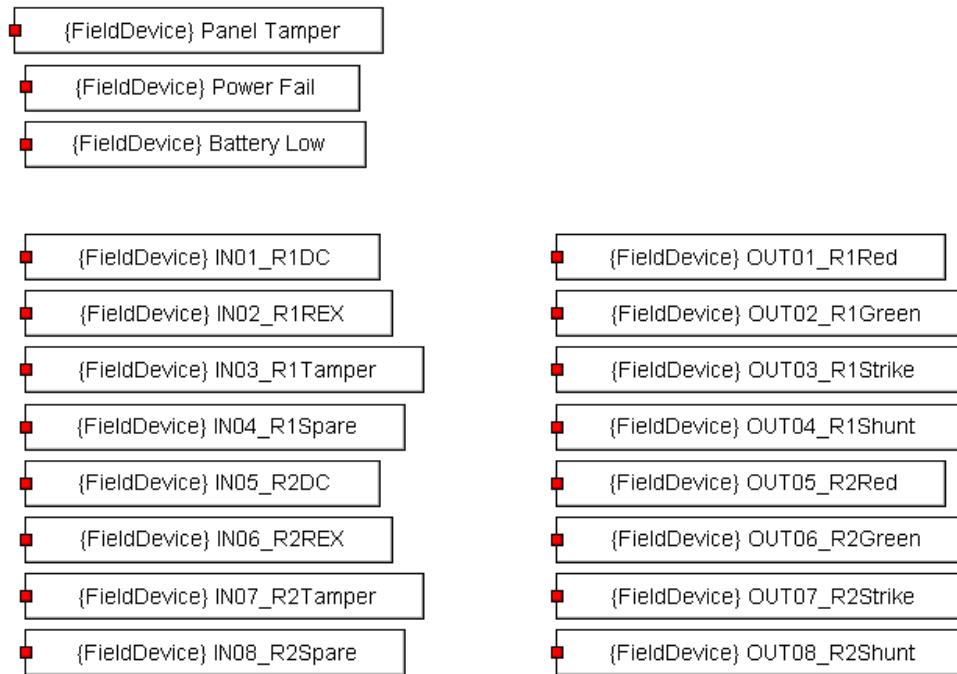
Object Diagram

Figure 6-9: Object Diagram for the JCI_RDR2SA_Full-IO Template

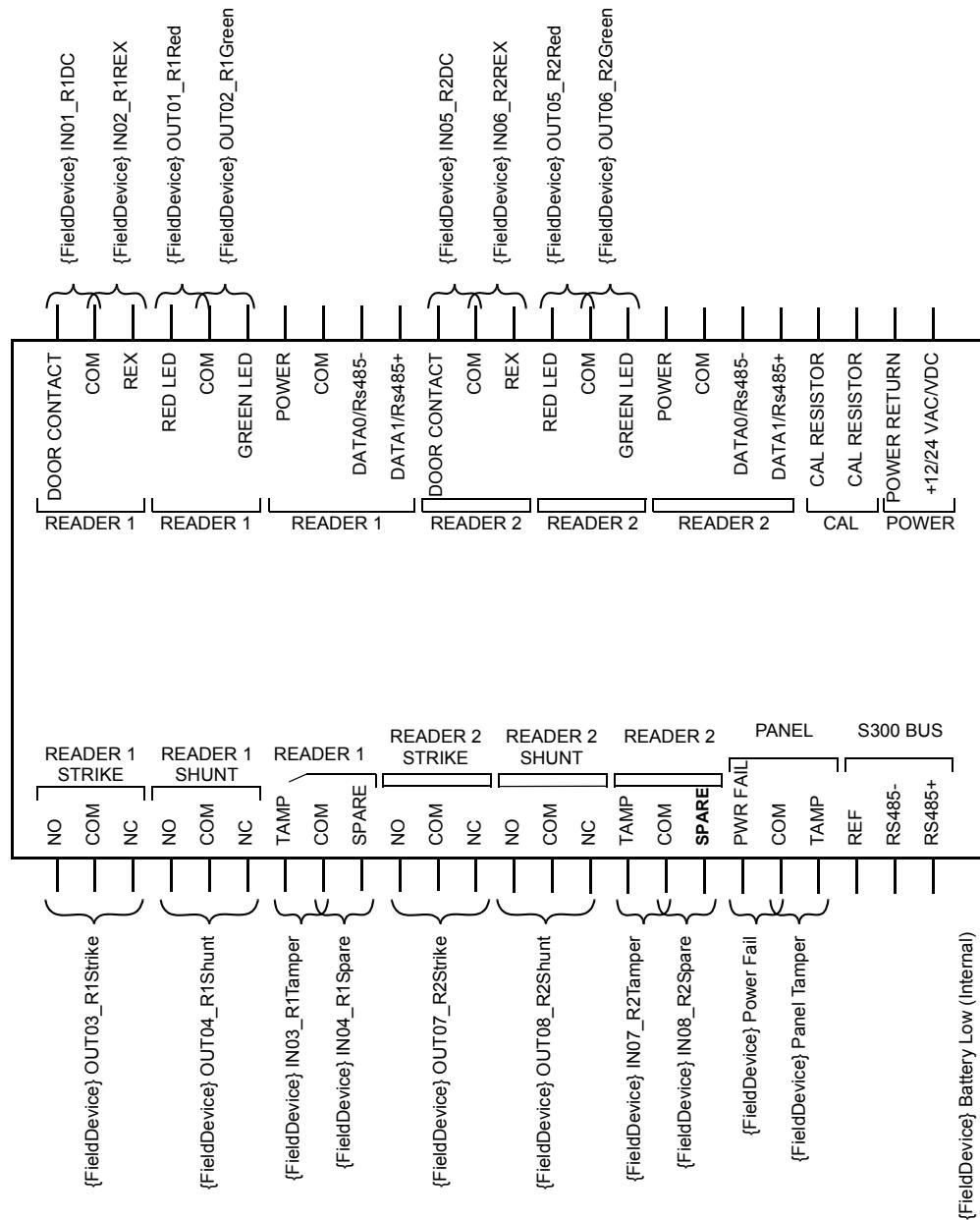
Graphical Representation

Figure 6-10: Graphical Representation of the JCI_RDR2SA_Full-IO Template

Object List and Description

Table 6-9: Object List for the JCI_RDR2SA_Full-IO Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the RDR2S-A module.
Security Supervised Input	{FieldDevice} Panel Tamper {FieldDevice} Power Fail {FieldDevice} Battery Low	Device-related inputs. Connector: Panel Tamper Connector: Power Fail Connector: Panel Battery Low
Security Supervised Input	{FieldDevice} IN01_R1DC {FieldDevice} IN02_R1REX {FieldDevice} IN03_R1Tamper {FieldDevice} IN04_R1Spare {FieldDevice} IN05_R2DC {FieldDevice} IN06_R2REX {FieldDevice} IN07_R2Tamper {FieldDevice} IN08_R2Spare	General purpose inputs. Configuration for inputs can be 2-state or 4-state. Connector: Reader 1 Door Contact Connector: Reader 1 REX Connector: Reader 1 Tamper Connector: Reader 1 Spare Connector: Reader 2 Door Contact Connector: Reader 2 REX Connector: Reader 2 Tamper Connector: Reader 2 Spare
Security Binary Output	{FieldDevice} OUT1_R1Red {FieldDevice} OUT2_R1Green {FieldDevice} OUT3_R1Strike {FieldDevice} OUT4_R1Shunt {FieldDevice} OUT5_R2Red {FieldDevice} OUT6_R2Green {FieldDevice} OUT7_R2Strike {FieldDevice} OUT8_R2Shunt	General purpose outputs. Each output can be timed, set, reset, fast flash, or slow flash. Connector: Reader 1 Red LED Connector: Reader 1 Green LED Connector: Reader 1 Strike Connector: Reader 1 Shunt Connector: Reader 2 Red LED Connector: Reader 2 Green LED Connector: Reader 2 Strike Connector: Reader 2 Shunt

Non-Default Attributes

Table 6-10: Non-Default Attributes for the JCI_RDR2SA_Full-IO Template

Object	Attribute	Non-Default Value
All Security Supervised Input objects	Suppress Default	Selected

JCI_RDR8S_Full-IO

This template adds objects for an RDR8S hardware module with 3 device-related inputs (Panel Tamper, Power Fail, and Battery Low), 32 suppressed inputs, and 32 outputs.

Object Diagram

The object diagram is not included in this section due to the large number of objects in this template.

Graphical Representation

The RDR8S hardware module is currently not available for purchase.

Object List and Description

Table 6-11: Object List for the JCI_RDR8S_Full-IO Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the RDR8S module.
Security Supervised Input	{FieldDevice} Panel Tamper {FieldDevice} Power Fail {FieldDevice} Battery Low	Device-related inputs. Connector: Panel Tamper Connector: Power Fail Connector: Panel Battery Low

Table 6-11: Object List for the JCI_RDR8S_Full-IO Template

Type	Name	Description
Security Supervised Input	{FieldDevice} IN01_R1DC {FieldDevice} IN02_R1REX {FieldDevice} IN03_R1Tamper {FieldDevice} IN04_R1Spare {FieldDevice} IN05_R2DC {FieldDevice} IN06_R2REX {FieldDevice} IN07_R2Tamper {FieldDevice} IN08_R2Spare {FieldDevice} IN09_R3DC {FieldDevice} IN10_R3REX {FieldDevice} IN11_R3Tamper {FieldDevice} IN12_R3Spare {FieldDevice} IN13_R4DC {FieldDevice} IN14_R4REX {FieldDevice} IN15_R4Tamper {FieldDevice} IN16_R4Spare {FieldDevice} IN17_R5DC {FieldDevice} IN18_R5REX {FieldDevice} IN19_R5Tamper {FieldDevice} IN20_R5Spare {FieldDevice} IN21_R6DC {FieldDevice} IN22_R6REX {FieldDevice} IN23_R6Tamper {FieldDevice} IN24_R6Spare {FieldDevice} IN25_R7DC {FieldDevice} IN26_R7REX {FieldDevice} IN27_R7Tamper {FieldDevice} IN28_R7Spare {FieldDevice} IN29_R8DC {FieldDevice} IN30_R8REX {FieldDevice} IN31_R8Tamper {FieldDevice} IN32_R8Spare	General purpose inputs. Configuration for inputs can be 2-state or 4-state. Connector: Reader 1 Door Contact Connector: Reader 1 REX Connector: Reader 1 Tamper Connector: Reader 1 Spare Connector: Reader 2 Door Contact Connector: Reader 2 REX Connector: Reader 2 Tamper Connector: Reader 2 Spare Connector: Reader 3 Door Contact Connector: Reader 3 REX Connector: Reader 3 Tamper Connector: Reader 3 Spare Connector: Reader 4 Door Contact Connector: Reader 4 REX Connector: Reader 4 Tamper Connector: Reader 4 Spare Connector: Reader 5 Door Contact Connector: Reader 5 REX Connector: Reader 5 Tamper Connector: Reader 5 Spare Connector: Reader 6 Door Contact Connector: Reader 6 REX Connector: Reader 6 Tamper Connector: Reader 6 Spare Connector: Reader 7 Door Contact Connector: Reader 7 REX Connector: Reader 7 Tamper Connector: Reader 7 Spare Connector: Reader 8 Door Contact Connector: Reader 8 REX Connector: Reader 8 Tamper Connector: Reader 8 Spare

Table 6-11: Object List for the JCI_RDR8S_Full-IO Template

Type	Name	Description
Security Binary Output	{FieldDevice} OUT1_R1Red {FieldDevice} OUT2_R1Green {FieldDevice} OUT3_R1Strike {FieldDevice} OUT4_R1Shunt {FieldDevice} OUT5_R2Red {FieldDevice} OUT6_R2Green {FieldDevice} OUT7_R2Strike {FieldDevice} OUT8_R2Shunt {FieldDevice} OUT9_R3Red {FieldDevice} OUT10_R3Green {FieldDevice} OUT11_R3Strike {FieldDevice} OUT12_R3Shunt {FieldDevice} OUT13_R4Red {FieldDevice} OUT14_R4Green {FieldDevice} OUT15_R4Strike {FieldDevice} OUT16_R4Shunt {FieldDevice} OUT17_R5Red {FieldDevice} OUT18_R5Green {FieldDevice} OUT19_R5Strike {FieldDevice} OUT20_R5Shunt {FieldDevice} OUT21_R6Red {FieldDevice} OUT22_R6Green {FieldDevice} OUT23_R6Strike {FieldDevice} OUT24_R6Shunt {FieldDevice} OUT25_R7Red {FieldDevice} OUT26_R7Green {FieldDevice} OUT27_R7Strike {FieldDevice} OUT28_R7Shunt {FieldDevice} OUT29_R8Red {FieldDevice} OUT30_R8Green {FieldDevice} OUT31_R8Strike {FieldDevice} OUT32_R8Shunt	General purpose outputs. Each output can be timed, set, reset, fast flash, or slow flash. Connector: Reader 1 Red LED Connector: Reader 1 Green LED Connector: Reader 1 Strike Connector: Reader 1 Shunt Connector: Reader 2 Red LED Connector: Reader 2 Green LED Connector: Reader 2 Strike Connector: Reader 2 Shunt Connector: Reader 3 Red LED Connector: Reader 3 Green LED Connector: Reader 3 Strike Connector: Reader 3 Shunt Connector: Reader 4 Red LED Connector: Reader 4 Green LED Connector: Reader 4 Strike Connector: Reader 4 Shunt Connector: Reader 5 Red LED Connector: Reader 5 Green LED Connector: Reader 5 Strike Connector: Reader 5 Shunt Connector: Reader 6 Red LED Connector: Reader 6 Green LED Connector: Reader 6 Strike Connector: Reader 6 Shunt Connector: Reader 7 Red LED Connector: Reader 7 Green LED Connector: Reader 7 Strike Connector: Reader 7 Shunt Connector: Reader 8 Red LED Connector: Reader 8 Green LED Connector: Reader 8 Strike Connector: Reader 8 Shunt

*Non-Default Attributes**Table 6-12: Non-Default Attributes for the JCI_RDR8S_Full-IO Template*

Object	Attribute	Non-Default Value
All Security Supervised Input objects	Suppress Default	Selected

Basic Hardware Module Templates

You can use the following templates as an alternative to adding the hardware module via the **Insert>Field Device** menu option on the P200 SCT menu bar.

JCI_RDR2SA_Basic

This template adds objects for an RDR2S-A hardware module with three device-related inputs (Panel Tamper, Power Fail, and Battery Low). The remaining 8 inputs, 8 outputs and 2 readers are not configured and can be added through x-Templates.

Object Diagram

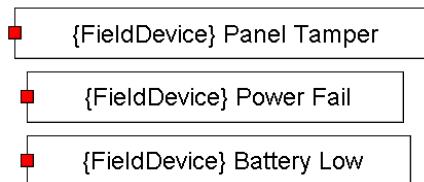


Figure 6-11: Object Diagram for the JCI_RDR2SA_Basic Template

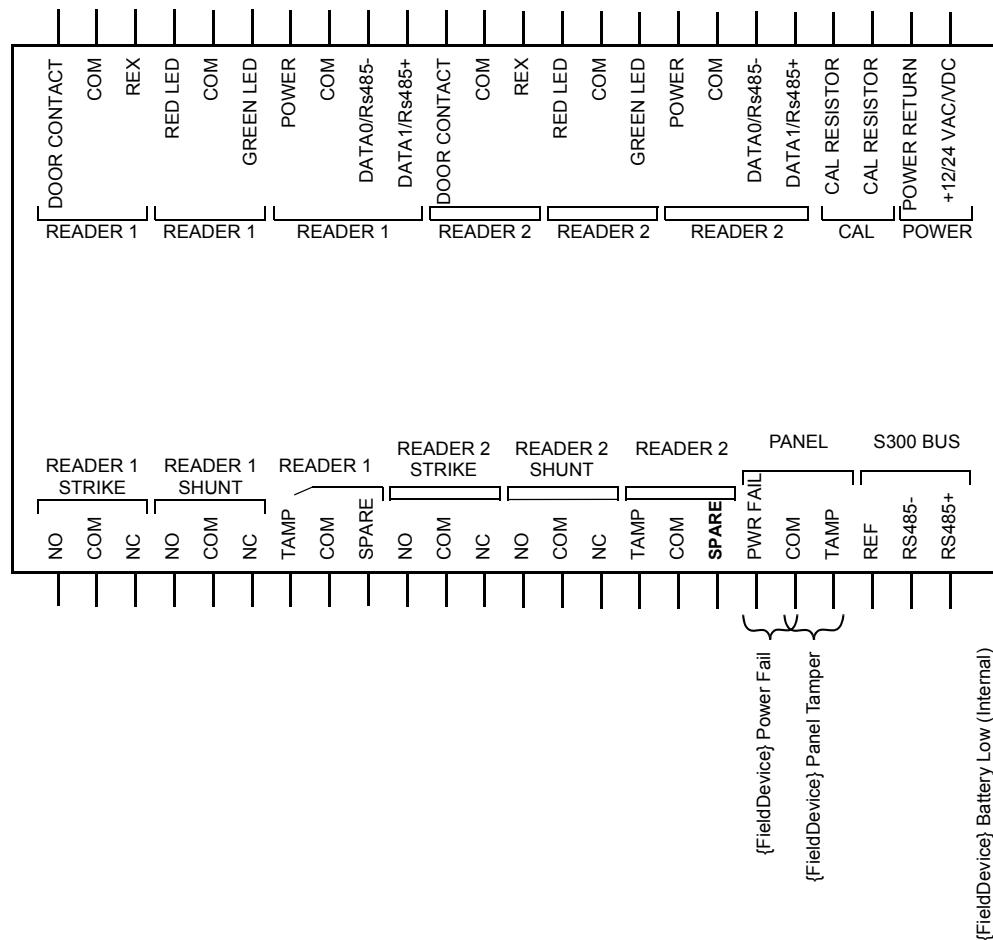
Graphical Representation

Figure 6-12: Graphical Representation of the JCI_RDR2SA_Basic Template Object List and Description

Table 6-13: Object List for the JCI_RDR2SA_Basic Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the RDR2S-A module.
Security Supervised Input	{FieldDevice} Panel Tamper {FieldDevice} Power Fail {FieldDevice} Battery Low	Device-related inputs. Connector: Panel Tamper Connector: Power Fail Connector: Panel Battery Low

Non-Default Attributes

The objects in this template do not have non-default attributes.

JCI_RDR8S_Basic

This template adds objects for an RDR8S hardware module with three device-related inputs (Panel Tamper, Power Fail, and Battery Low). The remaining 32 inputs, 32 outputs, and 8 readers are not configured and can be added through x-Templates.

Object Diagram

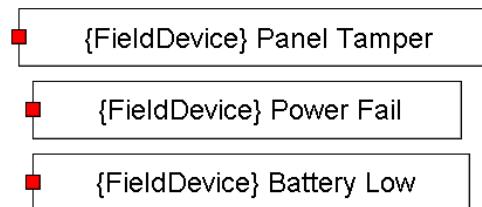


Figure 6-13: Object Diagram for the JCI_RDR8S_Basic Template

Graphical Representation

The RDR8S hardware module is currently not available for purchase.

Object List and Description

Table 6-14: Object List for the JCI_RDR8S_Basic Template

Type	Name	Description
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the RDR8S-A module.
Security Supervised Input	{FieldDevice} Panel Tamper {FieldDevice} Power Fail {FieldDevice} Battery Low	Device-related inputs. Connector: Panel Tamper Connector: Power Fail Connector: Panel Battery Low

Non-Default Attributes

The objects in this template do not have non-default attributes.

Door Hardware Module Templates

The following templates create a two-door hardware module and a door application. Depending on the hardware module and the door type used, unused field points may remain, possibly allowing you to add another door.

NOTE

The RDR2S-A and RDR2S have different modes of operation. Depending on the mode selected, the unit maps its I/O terminals accordingly. Keep this in mind when using templates with these hardware modules.

NOTE

General purpose inputs and outputs are not defined in these templates.

JCI_RDR2S_Card-In

This template adds objects for an RDR2S hardware module with Reader 1 enabled for use with a single, fully configured Card-In door. The remaining input/output points, which are not defined in the P2000 SCT, are designated as general purpose I/Os, meaning you can use these I/Os for other applications, such as a reader with a tamper switch (input), alarm annunciation output (siren, lights), or for a different door entirely.

Object Diagram

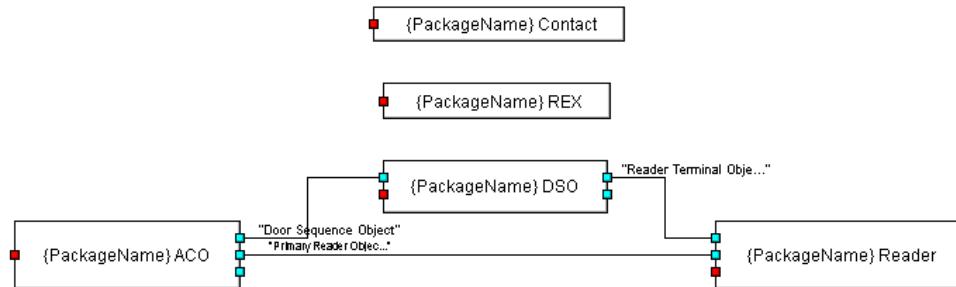


Figure 6-14: Object Diagram for the JCI_RDR2S_Card-In Template

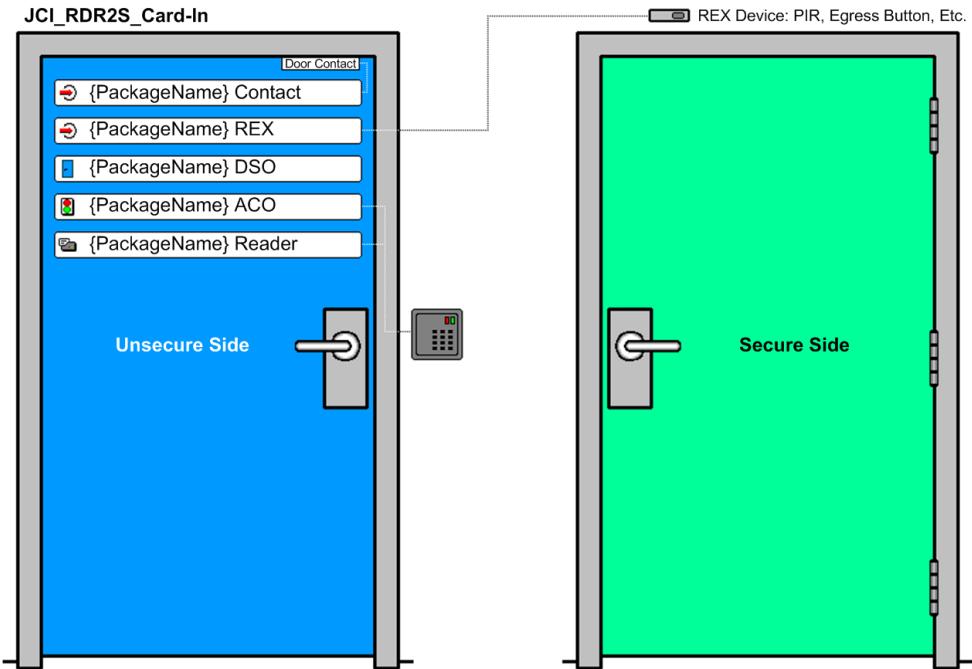
Graphical Representation

Figure 6-15: Graphical Representation of the JCI_RDR2S_Card-In Template

Object List and Description

Table 6-15: Object List for the JCI_RDR2S_Card-In Template

Type	Name	Description
Access Control	{PackageName} ACO	Controls access control logic for the door. Sends door commands to the {PackageName} DSO object. Destination Objects: {PackageName} DSO {PackageName} Reader
Door Sequence	{PackageName} DSO	Logic for controlling the door hardware. When prompted by the {PackageName} ACO object, the {PackageName} DSO object can control the door hardware, such as the strike or magnetic lock. Source Object: {PackageName} ACO Destination Object: {PackageName} Reader
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.

Table 6-15: Object List for the JCI_RDR2S_Card-In Template

Type	Name	Description
S300 Hardware Module	{FieldDevice}	S300 hardware module object that represents the RDR2S module.
Security Supervised Input	{PackageName} Contact	Object representing the door contact, which allows the system to determine the state of the door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device. Connector: IN11 The door contact is wired to the IN11 input on the RDR2S module.
Security Supervised Input	{PackageName} REX	Object representing the request to exit (REX) device. See “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device. Connector: IN12 The REX device is wired to the IN12 input on the RDR2S module.
S300 Reader Terminal	{PackageName} Reader	Object representing the entry reader. Source Objects: {PackageName} ACO {PackageName} DSO Connector: DATA0 / DATA1 (top)

*Non-Default Attributes**Table 6-16: Non-Default Attributes for the JCI_RDR2S_Card-In Template*

Object	Attribute	Non-Default Value
{PackageName} ACO	Primary Reader Object	{PackageName} Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} DSO	Reader Terminal Object	{PackageName} Reader
	Timed Override Mode	Timed Override
{PackageName} Contact	Suppress Default	Selected
{PackageName} REX	Debounce Time	50 ms
	Suppress Default	Selected

Assumptions

Since Reader 1 is defined, the following output points are automatically assigned (and therefore not available as general purpose outputs):

OUT11 – Red Light Output for Reader 1. Output is set every time an access deny occurs.

OUT12 – Green Light Output for Reader 1. Output is set every time an access grant occurs.

OUT13 – Shunt Output for Reader 1. Output is set every time an access grant occurs.

NO / NC (top) – Output controlling the door strike or magnetic lock. NO = Normally Open; NC = Normally Closed.

JCI_RDR2S_Card-In-Card-Out

This template adds objects for the RDR2S hardware module with Reader 1 and Reader 2 enabled for use with a single Card-In-Card-Out door. There are two remaining input points and two output points for use as general purpose I/Os.

Wiring Instructions for Offline Mode and When Using an External REX Device

In previous Card-In-Card-Out installations, it was required to physically connect the Ingress Terminal's REX input to the Egress Terminal's normally open Strike Output. See Figure 6-16.

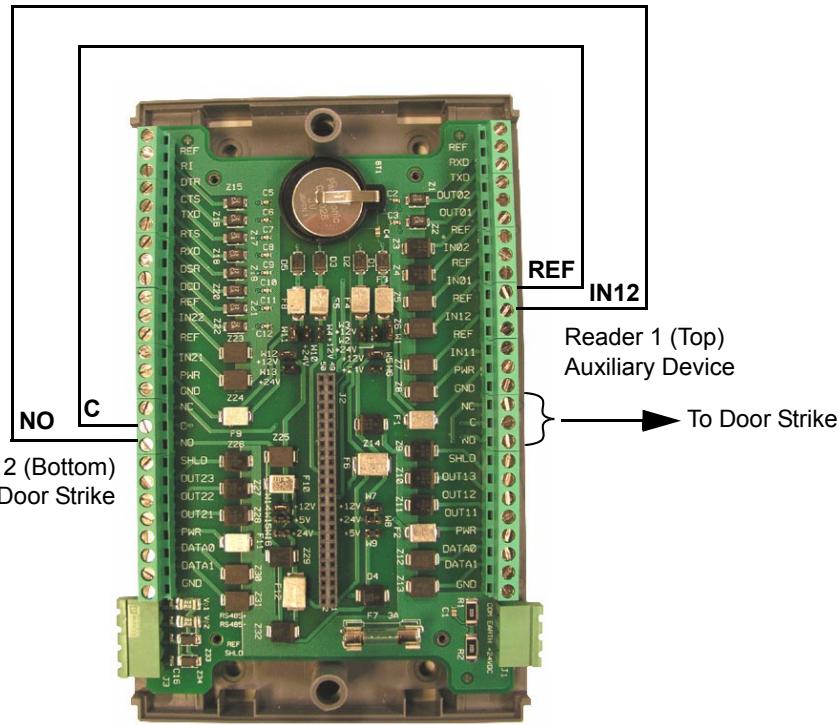


Figure 6-16: RDR2S Cross-Wiring for Two Readers Sharing a Single Door Strike

This cross-wiring is no longer needed, unless the RDR2S is required to operate in **Offline Mode**, which enables the RDR2S to grant an entity access based on **Facility Code Only** when the RDR2S has lost communication with its supervisory device (i.e. the CK722 controller).

To support an external REX device, it must be connected as follows:

- If the cross-wiring is in place, the external REX device must be connected to the **Egress** Terminal's REX input (*Secondary REX*).
- If the cross-wiring is **not** in place, the external REX device needs to be connected to the **Ingress** Terminal's REX input (*Primary REX*).

Object Diagram

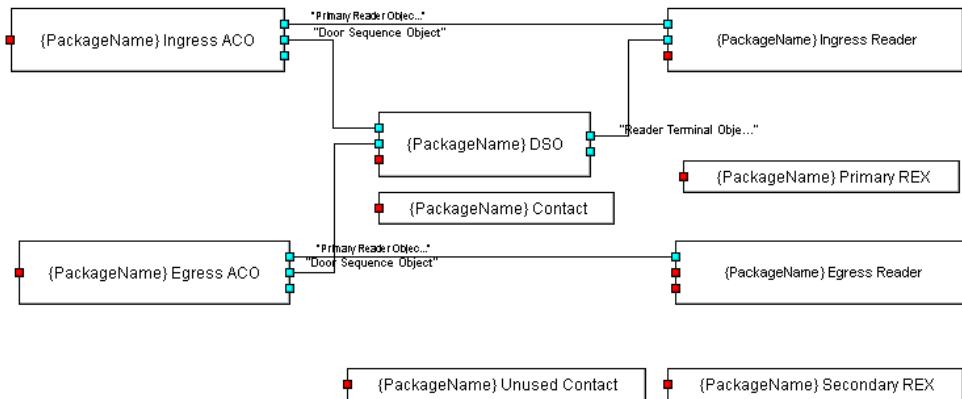


Figure 6-17: Object Diagram for the JCI_RDR2S_Card-In-Card-Out Template

Graphical Representation

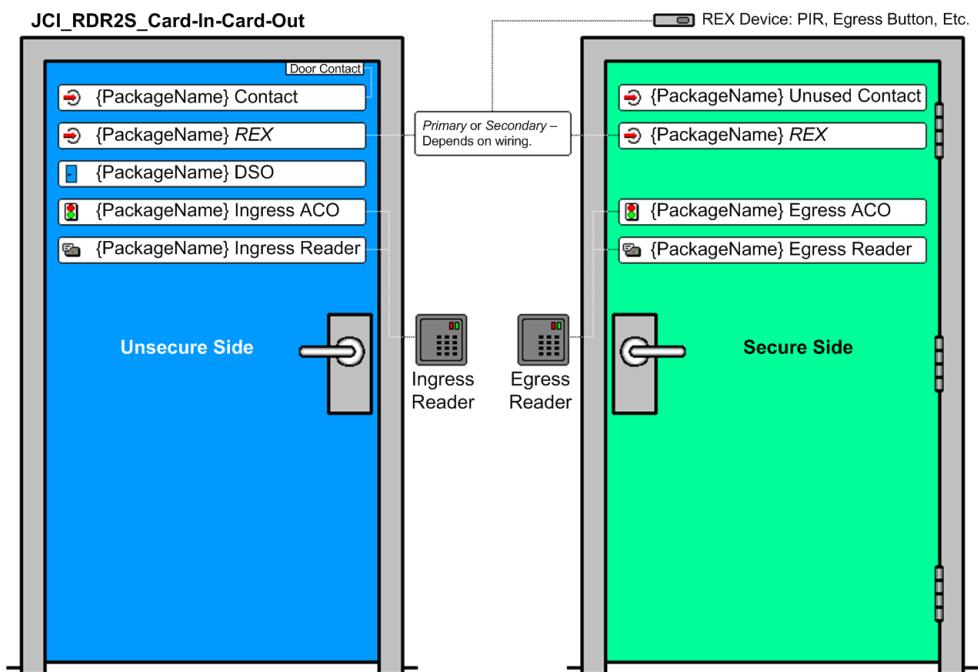


Figure 6-18: Graphical Representation of the JCI_RDR2S_Card-In-Card-Out Template

Object List and Description

Table 6-17: Object List for the JCI_RDR2S_Card-In-Card-Out Template

Type	Name	Description
Access Control	{PackageName} Ingress ACO	Controls access control logic for the <i>unsecure</i> side of the door. Sends door commands to {PackageName} DSO object. Destination Objects: {PackageName} DSO {PackageName} Ingress Reader
Access Control	{PackageName} Egress ACO	Controls access control logic for the <i>secure</i> side of the door. Sends door commands to {PackageName} DSO object. Destination Objects: {PackageName} DSO {PackageName} Egress Reader
Door Sequence	{PackageName} DSO	Logic for controlling the door hardware. When prompted by the {PackageName} Ingress ACO object or {PackageName} Egress ACO, the {PackageName} DSO object can control the door hardware, such as the strike or magnetic lock. In a Card-In-Card-Out application using the RDR2S hardware module, the Door Sequence object connects to the reader that is wired to the door contact. In this template, that reader is {PackageName} Ingress Reader. Source Objects: {PackageName} Ingress ACO {PackageName} Egress ACO Destination Object: {PackageName} Ingress Reader
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the RDR2S module.
Security Supervised Input	{PackageName} Contact	Object representing the door contact, which allows the system to determine the state of the door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device. Connector: IN11 The door contact is wired to the IN11 input on the RDR2S module.

Table 6-17: Object List for the JCI_RDR2S_Card-In-Card-Out Template

Type	Name	Description
Security Supervised Input	{PackageName} Unused Contact	Object representing the unused contact. In a Card-in-Card-Out application, only one door contact is required, which would be wired to connector IN11. Therefore, there is an unused input at IN21.
Security Supervised Input	{PackageName} Primary REX	Object representing the request to exit (REX) device, if used and if cross-wiring is in place (see “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27). See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device. Connector: IN12 The <i>primary</i> REX device would be wired to the IN12 input on the RDR2S module.
Security Supervised Input	{PackageName} Secondary REX	Object representing the request to exit (REX) device, if used and if cross-wiring is not in place (see “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27). See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device. Connector: IN22 The <i>secondary</i> REX device would be wired to the IN22 input on the RDR2S module.
S300 Reader Terminal	{PackageName} Ingress Reader	Object representing the door’s ingress reader. Source Objects: {PackageName} Ingress ACO {PackageName} DSO Connector: DATA0 / DATA1 (top)
S300 Reader Terminal	{PackageName} Egress Reader	Object representing the door’s egress reader. Source Objects: {PackageName} Egress ACO {PackageName} DSO Connector: DATA0 / DATA1 (bottom)

Non-Default Attributes*Table 6-18: Non-Default Attributes for the JCI_RDR2S_Card-In-Card-Out Template*

Object	Attribute	Non-Default Value
{PackageName} Ingress ACO	Direction	Ingress
	Primary Reader Object	{PackageName} Ingress Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} Egress ACO	Direction	Egress
	Primary Reader Object	{PackageName} Egress Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} DSO	Reader Terminal Object	{PackageName} Ingress Reader
	Aux Mode	Shunt and Unlock Only
	Timed Override Mode	Timed Override
{PackageName} Contact	Suppress Default	Selected
{PackageName} Unused Contact	Suppress Default	Selected
{PackageName} Ingress Reader	Access Control Feedback	Selected
{PackageName} Egress Reader	Access Control Feedback	Selected
	Portal Contact Connected	Deselected
	Aux Input Connected	Not Connected
{PackageName} Primary REX	Debounce Time	50 ms
	Suppress Default	Selected
{PackageName} Secondary REX	Debounce Time	50 ms
	Suppress Default	Selected

Assumptions

Since Reader 1 is defined, the following output points are automatically defined (and therefore not available as general purpose outputs):

OUT11 – Red Light Output for Reader 1. Output is set every time an access deny occurs.

OUT12 – Green Light Output for Reader 1. Output is set every time an access grant occurs.

OUT13 – Shunt Output for Reader 1. Output is set every time an access grant occurs.

NO / NC (top) – Output controlling the door strike or magnetic lock for Door 1. NO = Normally Open; NC = Normally Closed.

The following outputs for Reader 2 are also not available as general purpose outputs:

OUT23 – Shunt Output for Reader 2. Output is set every time an access grant occurs.

NO / NC (bottom) – Output controlling the door strike or magnetic lock for Door 2. NO = Normally Open; NC = Normally Closed.

NOTE

OUT21, the red light output for Reader 2, and OUT22, the green light output for Reader 2, are not wired at the RDR2S. The red light and green light terminals for Reader 2 should be wired in parallel with Reader 1, so the readers' LEDs match when access or egress is granted or denied.

JCI_RDR2SA_Card-In

This template adds objects for an RDR2S-A hardware module with Reader 1 enabled for use with a single, fully configured Card-In door. The remaining input/output points, which are not defined in the P2000 SCT, are designated as general purpose I/Os, meaning you can use these I/Os for other applications.

For more information on this template, see the description of the “**JCI_RDR2S_Card-In**” template starting on page 6-24. Both templates are exactly the same, except for the *Hardware Module Type* attribute of the **S300 Hardware Module** object.

JCI_RDR2SA_Card-In-Card-Out

This template adds objects for the RDR2S-A hardware module with Reader 1 and Reader 2 enabled for use with a single Card-In-Card-Out door. There are three remaining input points and two output points for use as general purpose I/Os.

For more information on this template, see the description of the “**JCI_RDR2S_Card-In-Card-Out**” template starting on page 6-27. Both templates are exactly the same, except for the following:

- The *Hardware Module Type* attribute of the **S300 Hardware Module** object.

- The JCI_RDR2SA_Card-In-Card-Out template does not include the **{PackageName} Unused Contact** Security Supervised Input object, since the unused input point can be turned off on an RDR2S-A hardware module.

Door x-Templates

The following templates create a door application with varying levels of complexity. These templates do not define specific hardware modules; the *Hardware Module Type* attribute of all **S300 Hardware Module** objects are set to **Generic**, allowing you to define the module when loading the template as a package.

JCI_x_Contact

This template adds objects for a door, consisting of only a door contact, to an existing hardware module. In this type of application, the door remains physically unlocked and the door contact (**Security Supervised Input** object) and the **Door Sequence** object track and report the opening and closing of the door. A door propped open signals a P2000 alarm. To activate an alarm output device, such as a sounder, see “**JCI_x_Contact-w-Alarm**” on page 6-36.

Object Diagram

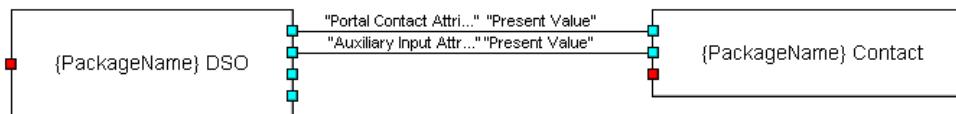


Figure 6-19: Object Diagram for the *JCI_x_Contact* Template

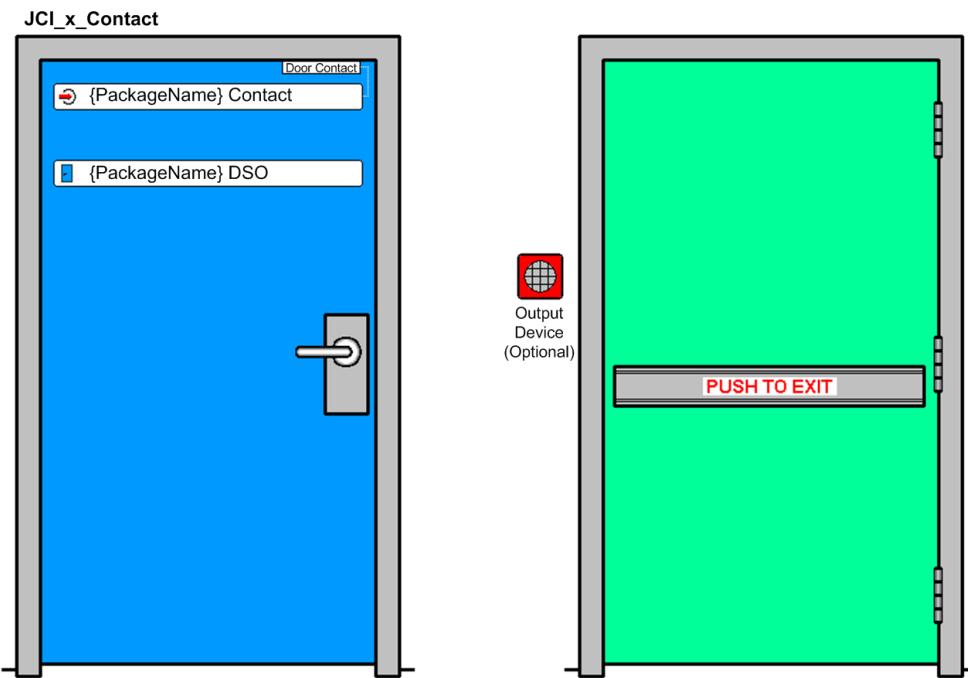
Graphical Representation

Figure 6-20: Graphical Representation of the JCI_x_Contact Template

Object List and Description

Table 6-19: Object List for the JCI_x_Contact Template

Type	Name	Description
Door Sequence	{PackageName} DSO	Logic for controlling the door hardware. The <i>Shunt Time</i> attribute is set for 10 seconds. A door held open longer than 10 seconds generates a P2000 notification. Destination Object: {PackageName} Contact
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.

Table 6-19: Object List for the JCI_x_Contact Template

Type	Name	Description
Security Supervised Input	{PackageName} Contact	<p>Object representing the door contact, which allows the system to determine the state of the door (open, closed, propped). See “Door Contacts” on page 4-13 for details on this type of device.</p> <p>Connector: Input 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

Non-Default Attributes*Table 6-20: Non-Default Attributes for the JCI_x_Contact Template*

Object	Attribute	Non-Default Value
{PackageName} DSO	Aux Mode	Shunt and Unlock Only
	Momentary Aux	Selected
	Access Time	10 seconds
	Alternate Access Time	10 seconds
	Shunt Time	10 seconds
	Timed Override Mode	Timed Override
	Portal Contact Attribute	{PackageName} Contact Attribute: Present Value
	Aux Input Attribute	{PackageName} Contact Attribute: Present Value
{PackageName} Contact	Suppress Default	Selected

JCI_x_Contact-w-Alarm

This template adds objects for a door, consisting of only a door contact and an alarm output, to an existing hardware module. In this type of application, the door remains physically unlocked and the door contact (**Security Supervised Input** object) and the **Door Sequence** object track and report the opening and closing of the door. A door propped open signals a P2000 alarm and activates the alarm output device, such as a sounder, controlled by the **Security Binary Output** and **Interlock** objects.

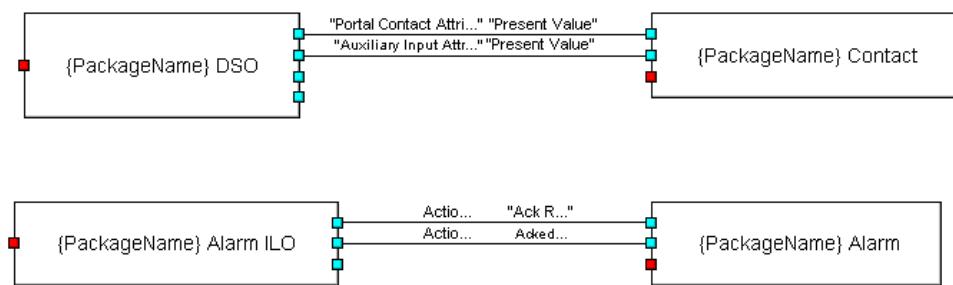
Object Diagram

Figure 6-21: Object Diagram for the JCI_x_Contact-w-Alarm Template

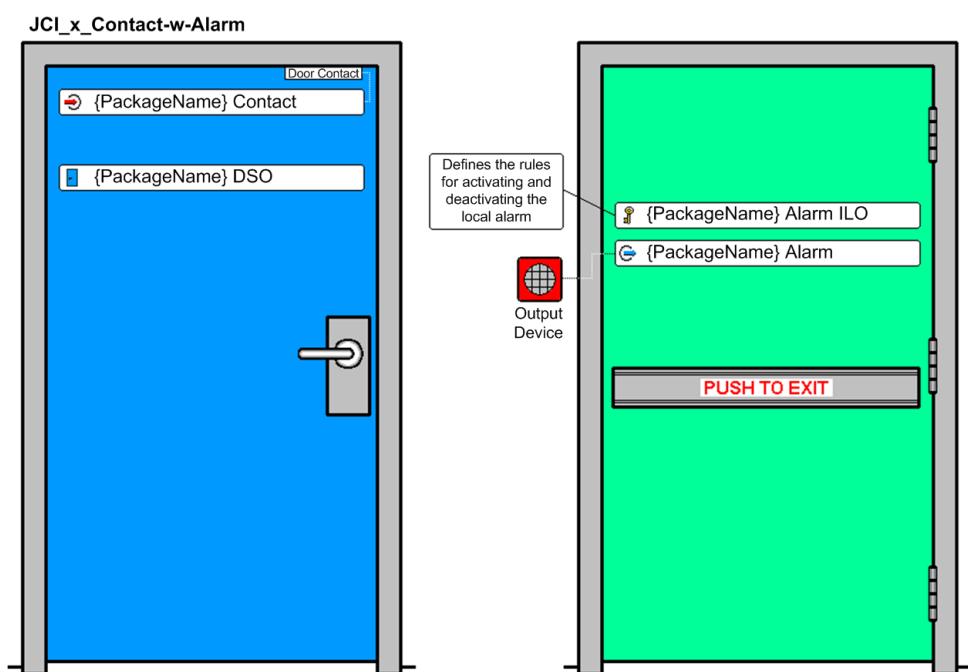
Graphical Representation

Figure 6-22: Graphical Representation of the JCI_x_Contact-w-Alarm Template

Object List and Description

Table 6-21: Object List for the JCI_x_Contact-w-Alarm Template

Type	Name	Description
Door Sequence	{PackageName} DSO	Logic for controlling the door hardware. The <i>Shunt Time</i> attribute is set for 10 seconds. A door held open longer than 10 seconds generates a P2000 notification and activates the alarm output. Destination Object: {PackageName} Contact
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.
Security Supervised Input	{PackageName} Contact	Object representing the door contact, which allows the system to determine the state of the door (open, closed, propped). See “Door Contacts” on page 4-13 for details on this type of device. Connector: Input 1 The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.
Security Binary Output	{PackageName} Alarm	Object representing the alarm output device, which is activated according to the conditions defined in the {PackageName} Alarm ILO object. Connector: Output 1 The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.
Interlock	{PackageName} Alarm ILO	Object with the following interlock definition and action: If the {PackageName} DSO object's <i>Present Value</i> attribute value is Propped Open and the <i>Suppress Propped Door</i> attribute value is deselected (false), the {PackageName} Alarm output will be in the activated state (e.g. the local alarm sounder will be activated). If the conditions do not match those previously described, the {PackageName} Alarm output will be in the released state (e.g. the local alarm sounder will be deactivated).

Non-Default Attributes

Table 6-22: Non-Default Attributes for the JCI_x_Contact-w-Alarm Template

Object	Attribute	Non-Default Value
{PackageName} DSO	Aux Mode	Shunt and Unlock Only
	Momentary Aux	Selected
	Access Time	10 seconds
	Alternate Access Time	10 seconds
	Shunt Time	10 seconds
	Timed Override Mode	Timed Override
	Portal Contact Attribute	{PackageName} Contact Attribute: Present Value
	Aux Input Attribute	{PackageName} Contact Attribute: Present Value
{PackageName} Contact	Suppress Default	Selected
{PackageName} Alarm ILO	Logic Equation	Match All (AND)
	Interlock Definition	{PackageName} DSO Present Value equal to Propped Open and Suppress Propped Door equal to Deselected (False)
	All Commands Priority	Selected – 16 (Default)
	Action Table (True)	Item: {PackageName} Alarm Command: Output Set: Activate Priority: 16 Delay: 0 second
	Action Table (False)	Item: {PackageName} Alarm Command: Output Set: Release Priority: 16 Delay: 0 second

JCI_x_Card-In

This template adds objects for a Card-In door to an existing hardware module. When mapping the field points, the contact, REX, and reader must be mapped to the same terminal (e.g. Terminal 1).

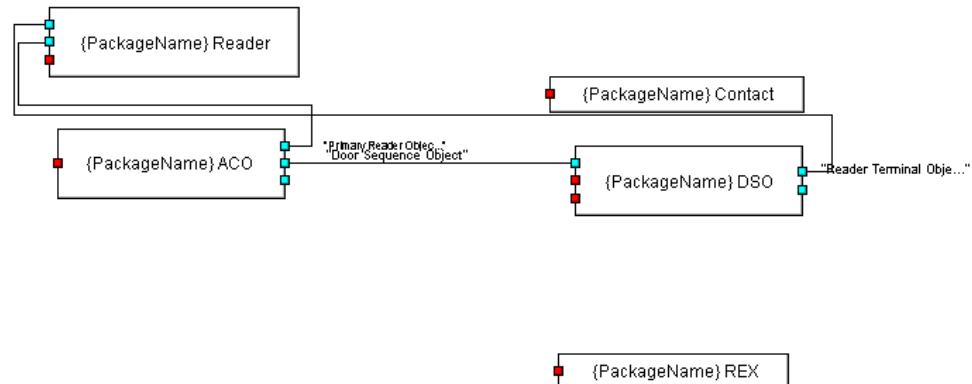
Object Diagram

Figure 6-23: Object Diagram for the JCI_x_Card-In Template

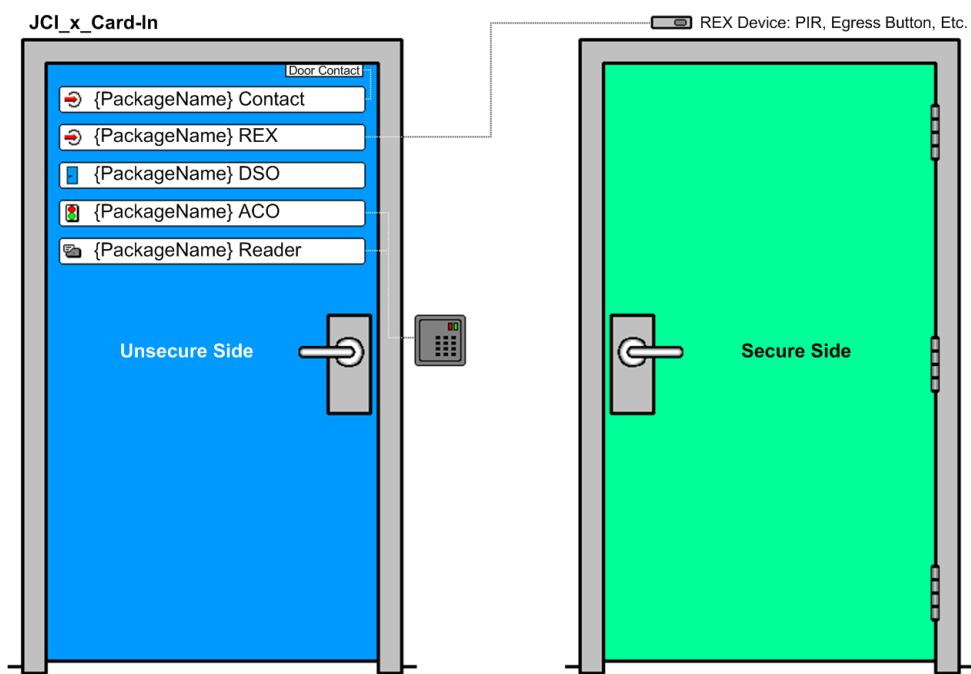
Graphical Representation

Figure 6-24: Graphical Representation of the JCI_x_Card-In Template

Object List and Description

Table 6-23: Object List for the JCI_x_Card-In Template

Type	Name	Description
Access Control	{PackageName} ACO	Controls access control logic for the door. Sends door commands to the {PackageName} DSO object. Destination Objects: {PackageName} DSO {PackageName} Reader
Door Sequence	{PackageName} DSO	Logic for controlling the door hardware. When prompted by the {PackageName} ACO object, the {PackageName} DSO object can control the door hardware, such as the strike or magnetic lock. Source Object: {PackageName} ACO Destination Object: {PackageName} Reader
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.
Security Supervised Input	{PackageName} Contact	Object representing the door contact, which allows the system to determine the state of the active door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device. Connector: Input 1 The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.
Security Supervised Input	{PackageName} REX	Object representing the request to exit (REX) device. See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device. Connector: Input 2 The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.

Table 6-23: Object List for the JCI_x_Card-In Template

Type	Name	Description
S300 Reader Terminal	{PackageName} Reader	<p>Object representing the door's reader.</p> <p>Source Objects: {PackageName} ACO {PackageName} DSO</p> <p>Connector: Reader 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

Non-Default Attributes*Table 6-24: Non-Default Attributes for the JCI_x_Card-In Template*

Object	Attribute	Non-Default Value
{PackageName} ACO	Primary Reader Object	{PackageName} Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} DSO	Reader Terminal Object	{PackageName} Reader
	Timed Override Mode	Timed Override
{PackageName} Contact	Suppress Default	Selected
{PackageName} REX	Debounce Time	50 ms
	Suppress Default	Selected

JCI_x_Card-In_TAMP

This template adds objects for a Card-In door with a reader tamper switch to an existing hardware module. When mapping the field points, the contact, REX, and reader must be mapped to the same terminal (e.g. Terminal 1).

For more information on this template, see the description of the JCI_x_Card-In template (see page 6-40). Both templates are exactly the same, except the JCI_x_Card-In_TAMP includes a **Security Supervised Input** object for a reader tamper switch.

JCI_x_Card-In_IAD

This template adds objects for a Card-In door with an independently reporting inactive door to an existing hardware module. When mapping the field points, the active contact, REX, and reader must be mapped to the same terminal (e.g. Terminal 1).

For more information on this template, see the description of the JCI_x_Card-In_IAD_TAMP template. Both templates are exactly the same, except the JCI_x_Card-In_IAD_TAMP includes a **Security Supervised Input** object for a reader tamper switch.

JCI_x_Card-In_IAD_TAMP

This template adds objects for a Card-In door, with an independently reporting inactive door and a reader tamper switch, to an existing hardware module. When mapping the field points, the active contact, REX, and reader must be mapped to the same terminal (e.g. Terminal 1).

Object Diagram

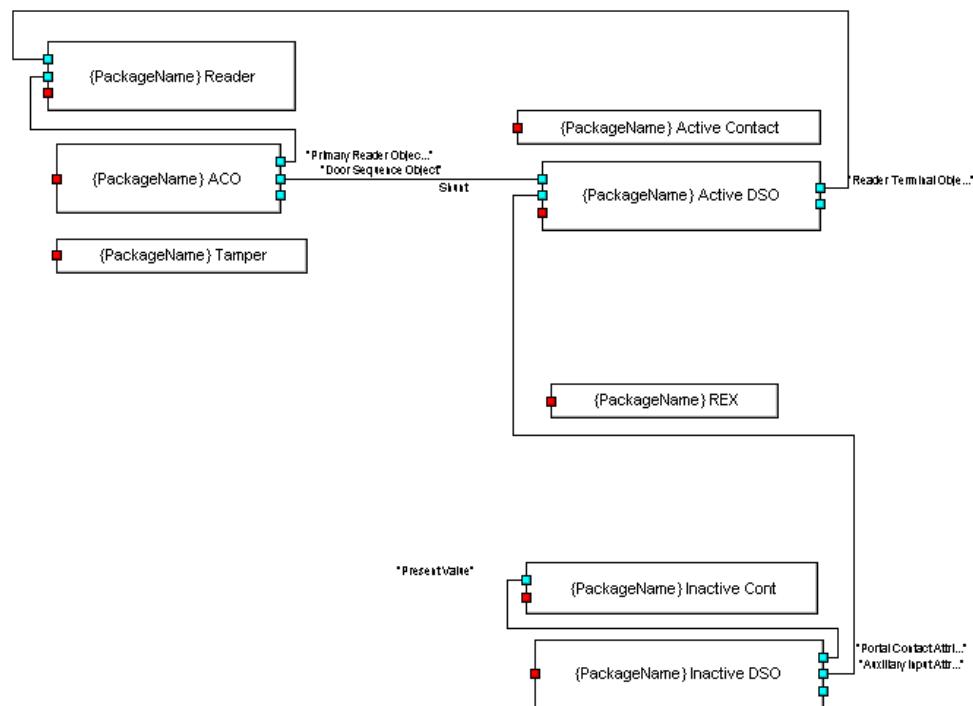


Figure 6-25: Object Diagram for the JCI_x_Card-In_IAD_TAMP Template

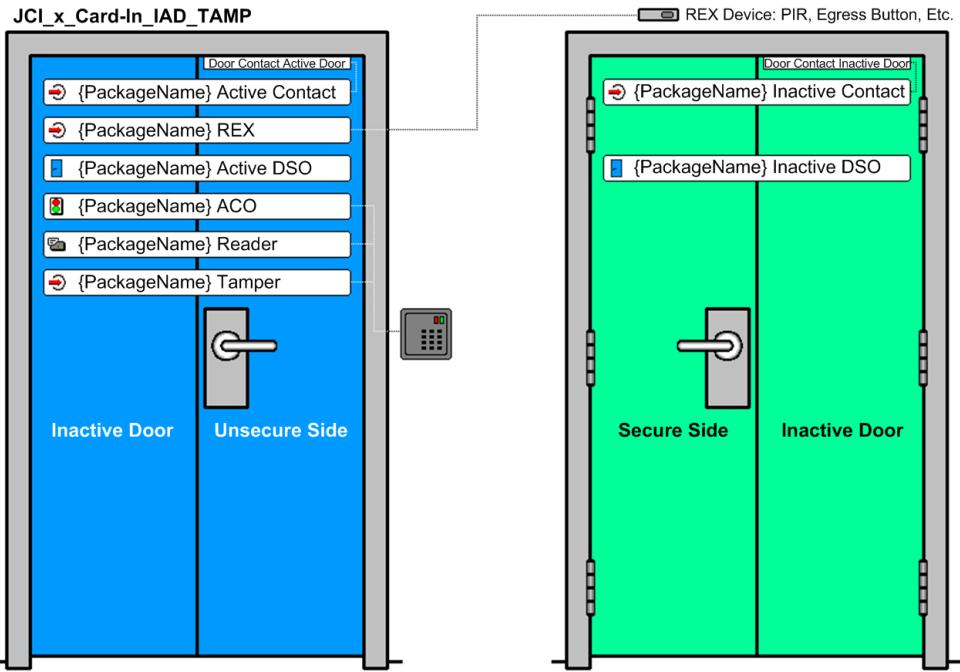
Graphical Representation

Figure 6-26: Graphical Representation of the JCI_x_Card-In_IAD_TAMP Template Object List and Description

Table 6-25: Object List for the JCI_x_Card-In_IAD_TAMP Template

Type	Name	Description
Access Control	{PackageName} ACO	<p>Controls access control logic for the door. Sends door commands to {PackageName} Active DSO object.</p> <p>Destination Objects: {PackageName} Active DSO {PackageName} Reader</p>
Door Sequence	{PackageName} Active DSO	<p>Logic for controlling the active door hardware. When prompted by the {PackageName} ACO object, the {PackageName} DSO object can control the active door hardware, such as the strike or magnetic lock.</p> <p>Source Objects: {PackageName} ACO {PackageName} Inactive DSO</p> <p>Destination Object: {PackageName} Reader</p>

Table 6-25: Object List for the JCI_x_Card-In_IAD_TAMP Template

Type	Name	Description
Door Sequence	{PackageName} Inactive DSO	<p>Logic for monitoring the inactive door hardware, such as the inactive door contact.</p> <p>This DSO shunts the alarm for the inactive door according to the current shunt status of the {PackageName} Active DSO object. If the active door is in Override Mode, for example, the alarm for the active door is shunted. In this case, the inactive door alarm will also be shunted, enabling the building occupants to open and prop both doors during the duration of the Override Mode.</p> <p>Destination Objects: {PackageName} Active DSO {PackageName} Inactive Contact</p>
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.
Security Supervised Input	{PackageName} Tamper	<p>Object representing the tamper input, which enables the system to report possible device tampering.</p> <p>Connector: Input 3</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Active Contact	<p>Object representing the active door contact, which allows the system to determine the state of the active door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device.</p> <p>Connector: Input 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

Table 6-25: Object List for the JCI_x_Card-In_IAD_TAMP Template

Type	Name	Description
Security Supervised Input	{PackageName} REX	<p>Object representing the request to exit (REX) device. See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device.</p> <p>Connector: Input 2</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Inactive Contact	<p>Object representing the inactive door contact, which allows the system to determine the state of the inactive door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device.</p> <p>Source Object: {PackageName} Inactive DSO</p> <p>Connector: Input 5</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
S300 Reader Terminal	{PackageName} Reader	<p>Object representing the door’s reader.</p> <p>Source Objects: {PackageName} ACO {PackageName} Active DSO</p> <p>Connector: Reader 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

*Non-Default Attributes**Table 6-26: Non-Default Attributes for the JCI_x_Card-In_IAD_TAMP Template*

Object	Attribute	Non-Default Value
{PackageName} Active Contact	Suppress Default	Selected
{PackageName} REX	Debounce Time	50 ms
	Suppress Default	Selected
{PackageName} Active DSO	Reader Terminal Object	{PackageName} Reader
	Timed Override Mode	Timed Override

Table 6-26: Non-Default Attributes for the JCI_x_Card-In_IAD_TAMP Template

Object	Attribute	Non-Default Value
{PackageName} ACO	Primary Reader Object	{PackageName} Reader
	Door Sequence Object	{PackageName} Active DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} Inactive Contact	Debounce Time	150 ms
	Suppress Default	Selected
{PackageName} Inactive DSO	Aux Mode	Shunt Only
	Access Time	1 second
	Alternate Access Time	1 second
	Shunt Time	1 second
	Timed Override Mode	Timed Override
	Portal Contact Attribute	{PackageName} Inactive Contact Attribute: Present Value
	Aux Input Attribute	{PackageName} Active DSO Attribute: Shunt

JCI_x_CICO

This template adds objects for a generic hardware module with Reader 1 and Reader 2 enabled for use with a single Card-In-Card-Out door.

NOTE

The wiring of certain modules, like the RDR2S and RDR2S-A, determine the assignment of the REX field points. See “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27 for more information.

Object Diagram

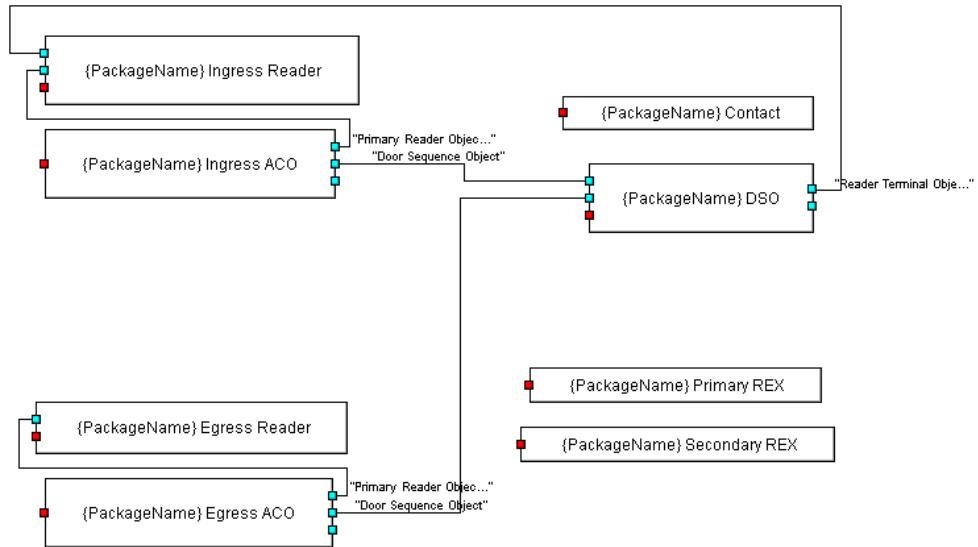


Figure 6-27: Object Diagram for the JCI_x_CICO Template

Graphical Representation

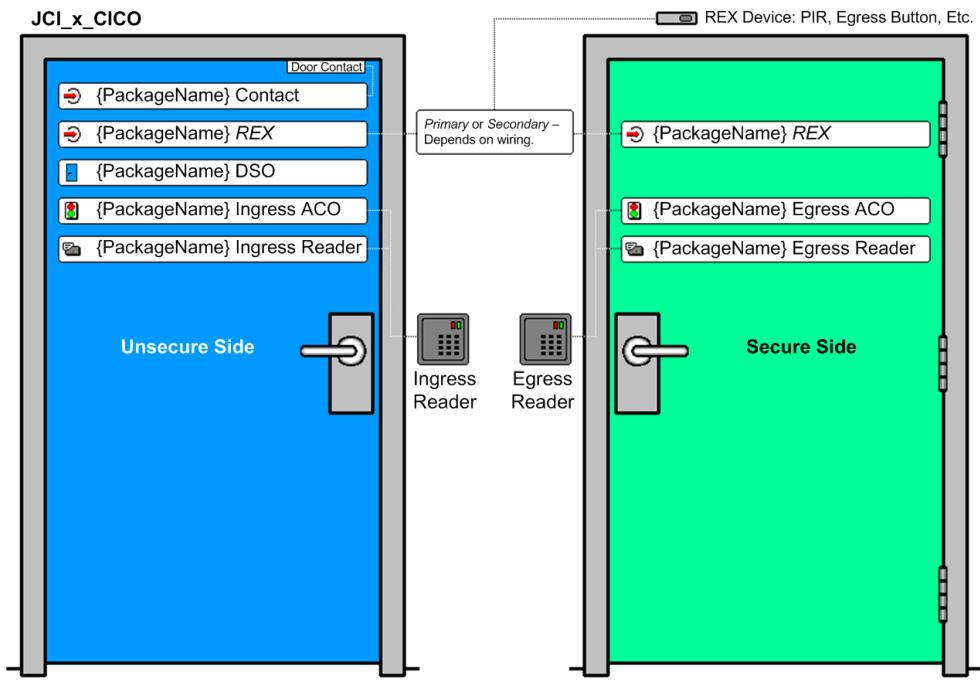


Figure 6-28: Graphical Representation of the JCI_x_CICO Template

Object List and Description

Table 6-27: Object List for the JCI_x_CICO Template

Type	Name	Description
Access Control	{PackageName} Ingress ACO	Controls access control logic for the <i>unsecure</i> side of the door. Sends door commands to {PackageName} DSO object. Destination Objects: • {PackageName} DSO • {PackageName} Ingress Reader
Access Control	{PackageName} Egress ACO	Controls access control logic for the <i>secure</i> side of the door. Sends door commands to {PackageName} DSO object. Destination Objects: • {PackageName} DSO • {PackageName} Egress Reader
Door Sequence	{PackageName} DSO	Logic for controlling the door hardware. When prompted by the {PackageName} Ingress ACO object or {PackageName} Egress ACO, the {PackageName} DSO object can control the door hardware, such as the strike or magnetic lock. Source Objects: • {PackageName} Ingress ACO • {PackageName} Egress ACO Destination Object: • {PackageName} Ingress Reader
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.
Security Supervised Input	{PackageName} Contact	Object representing the door contact, which allows the system to determine the state of the door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device. Connector: Input 1 The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.

Table 6-27: Object List for the JCI_x_CICO Template

Type	Name	Description
Security Supervised Input	{PackageName} Primary REX	<p>Object representing the request to exit (REX) device, if used and if cross-wiring is in place (see “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27). See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device.</p> <p>Connector: Input 2</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Secondary REX	<p>Object representing the request to exit (REX) device, if used and if cross-wiring is not in place (see “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27). See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device.</p> <p>Connector: Input 6</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
S300 Reader Terminal	{PackageName} Ingress Reader	<p>Object representing the door’s ingress reader.</p> <p>Source Objects: {PackageName} Ingress ACO {PackageName} DSO</p> <p>Connector: Reader 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
S300 Reader Terminal	{PackageName} Egress Reader	<p>Object representing the door’s egress reader.</p> <p>Source Objects: {PackageName} Egress ACO {PackageName} DSO</p> <p>Connector: Reader 2</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

Non-Default Attributes

Table 6-28: Non-Default Attributes for the JCI_x_CICO Template

Object	Attribute	Non-Default Value
{PackageName} Ingress ACO	Direction	Ingress
	Primary Reader Object	{PackageName} Ingress Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} Egress ACO	Direction	Egress
	Primary Reader Object	{PackageName} Egress Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} DSO	Reader Terminal Object	{PackageName} Ingress Reader
	Aux Mode	Shunt and Unlock Only
	Timed Override Mode	Timed Override
{PackageName} Contact	Suppress Default	Selected
{PackageName} Ingress Reader	Access Control Feedback	Selected
{PackageName} Egress Reader	Access Control Feedback	Selected
	Portal Contact Connected	Deselected
	Aux Input Connected	Not Connected
{PackageName} Primary REX	Debounce Time	50 ms
	Suppress Default	Selected
{PackageName} Secondary REX	Debounce Time	50 ms
	Suppress Default	Selected

JCI_x_CICO_TAMP

This template adds objects for a Card-In-Card-Out door with tamper switches on each reader to an existing hardware module.

For more information on this template, see the description of the JCI_x_CICO template (see page 6-47). Both templates are exactly the same, except the

JCI_x_CICO_TAMP includes two **Security Supervised Input** objects for reader tamper switches.

- The **{PackageName} Ingress Tamper** object represents the tamper switch for the **ingress** reader.
- The **{PackageName} Egress Tamper** object represents the tamper switch for the **egress** reader.

JCI_x_CICO_IAD

This template adds objects for a Card-In-Card-Out door with an independently reporting inactive door to an existing hardware module.

For more information on this template, see the description of the JCI_x_CICO_IAD_TAMP template. Both templates are exactly the same, except the JCI_x_CICO_IAD_TAMP includes two **Security Supervised Input** objects for reader tamper switches.

JCI_x_CICO_IAD_TAMP

This template adds objects for a Card-In-Card-Out door, with an independently reporting inactive door and tamper switches on each reader, to an existing hardware module.

When mapping the field points, the active contact, primary REX, and ingress reader must be mapped to the same terminal (e.g. Terminal 1). The secondary REX and the egress reader must be mapped to the same respective terminal (e.g. Terminal 2).

NOTE

The wiring of certain modules, like the RDR2S and RDR2S-A, determine the assignment of the REX field points. See “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27 for more information.

Object Diagram

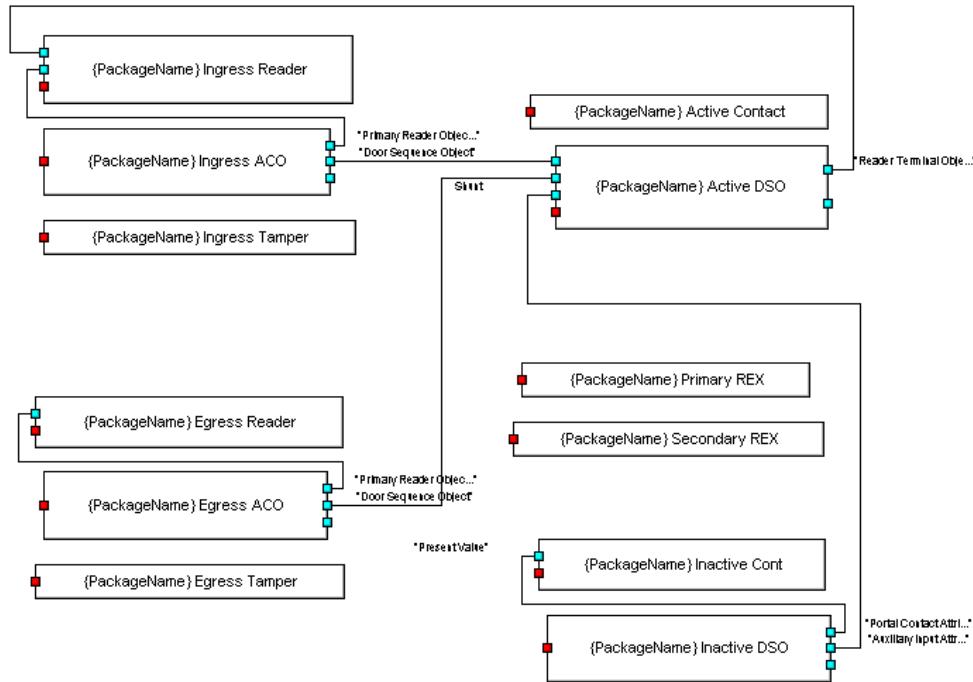


Figure 6-29: Object Diagram for the JCI_x_CICO_IAD_TAMP Template

Graphical Representation

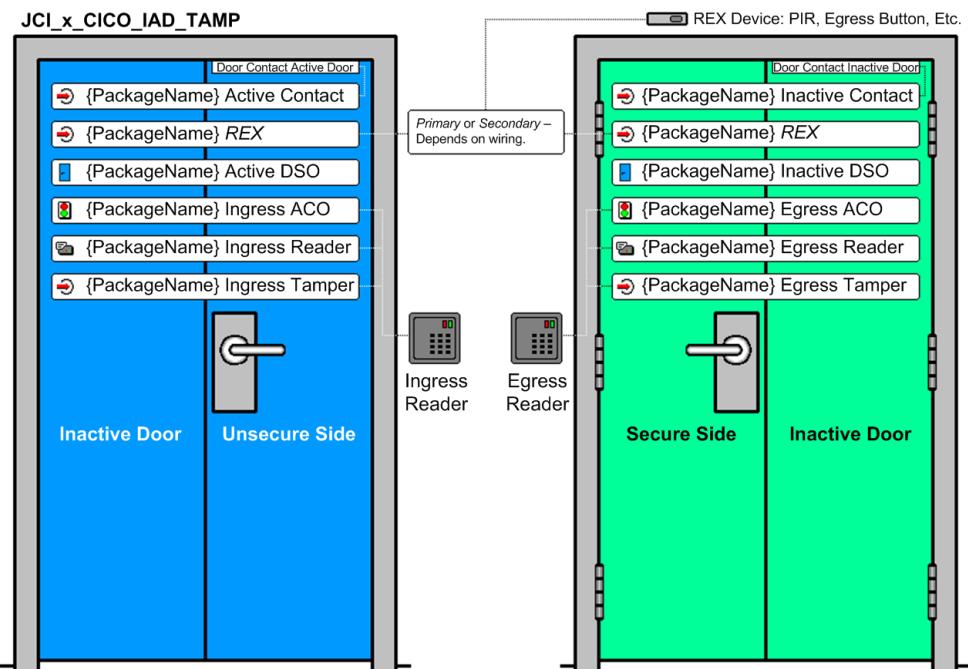


Figure 6-30: Graphical Representation of the JCI_x_CICO_IAD_TAMP Template

Object List and Description

Table 6-29: Object List for the JCI_x_CICO_IAD_TAMP Template

Type	Name	Description
Access Control	{PackageName} Ingress ACO	Controls access control logic for the <i>unsecure</i> side of the door. Sends door commands to the {PackageName} Active DSO object. Destination Objects: {PackageName} Active DSO {PackageName} Ingress Reader
Access Control	{PackageName} Egress ACO	Controls access control logic for the <i>secure</i> side of the door. Sends door commands to the {PackageName} Active DSO object. Destination Objects: {PackageName} Active DSO {PackageName} Egress Reader
Door Sequence	{PackageName} Active DSO	Logic for controlling the active door hardware. When prompted by the {PackageName} Ingress ACO object or {PackageName} Egress ACO, the {PackageName} Active DSO object can control the door hardware, such as the strike or magnetic lock. Source Objects: {PackageName} Ingress ACO {PackageName} Egress ACO Destination Object: {PackageName} Ingress Reader
Door Sequence	{PackageName} Inactive DSO	Logic for monitoring the inactive door hardware, such as the inactive door contact. This DSO shunts the alarm for the inactive door according to the current shunt status of the {PackageName} Active DSO object. If the active door is in Override Mode, for example, the alarm for the active door is shunted. In this case, the inactive door alarm will also be shunted, enabling the building occupants to open and prop both doors during the duration of the Override Mode. Destination Objects: {PackageName} Active DSO {PackageName} Inactive Contact
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.

Table 6-29: Object List for the JCI_x_CICO_IAD_TAMP Template

Type	Name	Description
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.
Security Supervised Input	{PackageName} Active Contact	<p>Object representing the active door contact, which allows the system to determine the state of the active door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device.</p> <p>Connector: Input 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Inactive Contact	<p>Object representing the inactive door contact, which allows the system to determine the state of the inactive door (open, closed, forced, propped). See “Door Contacts” on page 4-13 for details on this type of device.</p> <p>Connector: Input 5</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Primary REX	<p>Object representing the request to exit (REX) device, if used and if cross-wiring is in place (see “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27). See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device.</p> <p>Connector: Input 2</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Secondary REX	<p>Object representing the request to exit (REX) device, if used and if cross-wiring is not in place (see “Wiring Instructions for Offline Mode and When Using an External REX Device” on page 6-27). See also “Request to Exit (REX) or Egress Devices” on page 4-14 for details on this type of device.</p> <p>Connector: Input 6</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

Table 6-29: Object List for the JCI_x_CICO_IAD_TAMP Template

Type	Name	Description
Security Supervised Input	{PackageName} Ingress Tamper	<p>Object representing the tamper input, which enables the system to report possible device tampering, for the ingress reader.</p> <p>Connector: Input 3</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
Security Supervised Input	{PackageName} Egress Tamper	<p>Object representing the tamper input, which enables the system to report possible device tampering, for the egress reader.</p> <p>Connector: Input 7</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
S300 Reader Terminal	{PackageName} Ingress Reader	<p>Object representing the door's ingress reader.</p> <p>Source Objects: {PackageName} Ingress ACO {PackageName} Active DSO</p> <p>Connector: Reader 1</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>
S300 Reader Terminal	{PackageName} Egress Reader	<p>Object representing the door's egress reader.</p> <p>Source Object: {PackageName} Egress ACO</p> <p>Connector: Reader 2</p> <p>The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.</p>

Non-Default Attributes

Table 6-30: Non-Default Attributes for the JCI_x_CICO_IAD_TAMP Template

Object	Attribute	Non-Default Value
{PackageName} Ingress ACO	Direction	Ingress
	Primary Reader Object	{PackageName} Ingress Reader
	Door Sequence Object	{PackageName} Active DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} Egress ACO	Direction	Egress
	Primary Reader Object	{PackageName} Egress Reader
	Door Sequence Object	{PackageName} Active DSO
	Set 1 First Identifier Format	10000 - Default
{PackageName} Active DSO	Reader Terminal Object	{PackageName} Ingress Reader
	Aux Mode	Shunt and Unlock Only
	Timed Override Mode	Timed Override
{PackageName} Inactive DSO	Aux Mode	Shunt Only
	Access Time	1 second
	Alternate Access Time	1 second
	Shunt Time	1 second
	Timed Override Mode	Timed Override
	Portal Contact Attribute	{PackageName} Inactive Contact Attribute: Present Value
	Aux Input Attribute	{PackageName} Active DSO Attribute: Shunt
{PackageName} Active Contact	Suppress Default	Selected
{PackageName} Inactive Contact	Debounce Time	150 ms
	Suppress Default	Selected

Table 6-30: Non-Default Attributes for the JCI_x_CICO_IAD_TAMP Template

Object	Attribute	Non-Default Value
{PackageName} Ingress Reader	Access Control Feedback	Selected
{PackageName} Egress Reader	Access Control Feedback	Selected
	Portal Contact Connected	Deselected
	Aux Input Connected	Not Connected
{PackageName} Primary REX	Debounce Time	50 ms
	Suppress Default	Selected
{PackageName} Secondary REX	Debounce Time	50 ms
	Suppress Default	Selected

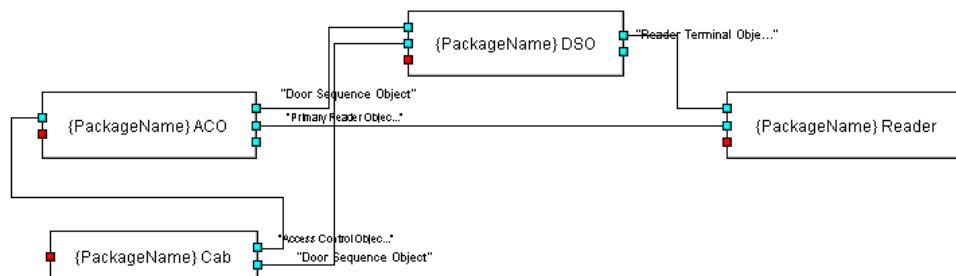
Miscellaneous Templates

Use these templates for applications other than doors or S300 Input/Output modules.

JCI_x_Elevator

This template adds objects for a low level integration elevator with all associated objects, except inputs, outputs, and schedules.

Object Diagram

*Figure 6-31: Object Diagram for the JCI_x_Elevator Template*

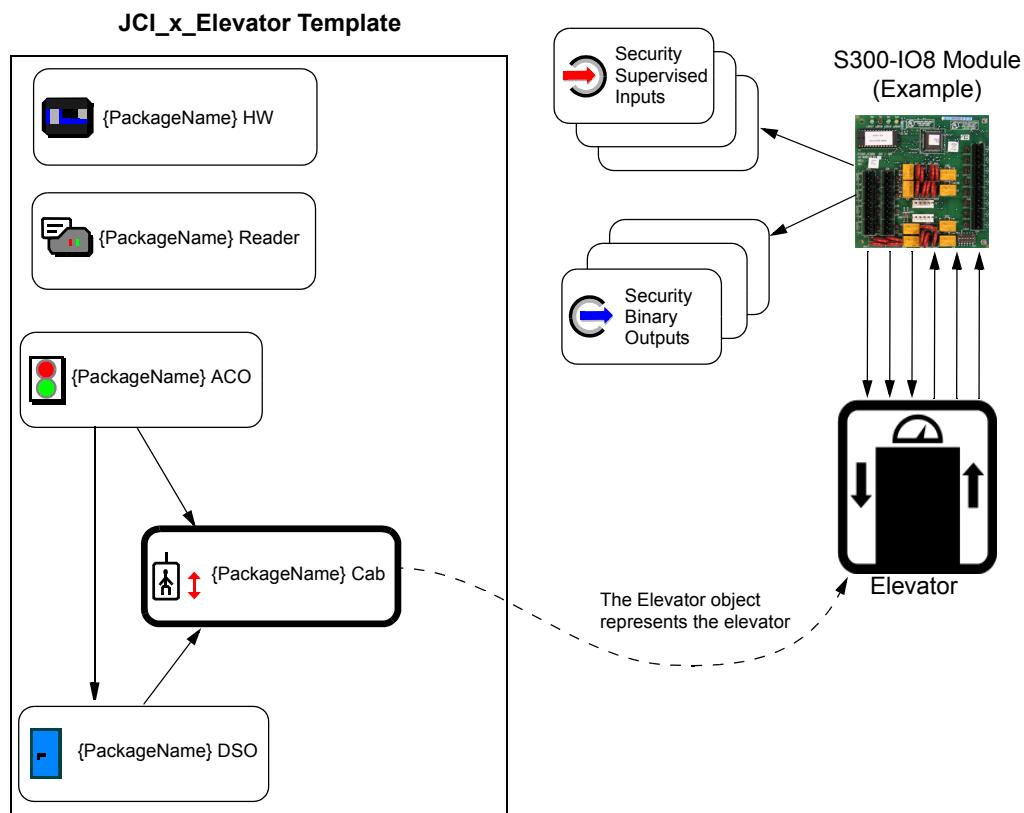
Graphical Representation

Figure 6-32: Graphical Representation of the JCI_x_Elevator Template

Object List and Description

Table 6-31: Object List for the JCI_x_Elevator Template

Type	Name	Description
Access Control	{PackageName} ACO	Controls access control logic for the elevator. Destination Objects: {PackageName} DSO {PackageName} Reader

Table 6-31: Object List for the JCI_x_Elevator Template

Type	Name	Description
Door Sequence	{PackageName} DSO	Monitors and controls elevator functions in conjunction with the {PackageName} ACO, {PackageName} Reader, and {PackageName} Cab objects. For example, the {PackageName} Cab object may monitor the {PackageName} DSO object to detect the override mode. When in override, all floors are accessible as if they were in public access. Source Object: {PackageName} ACO Destination Object: {PackageName} Reader
Elevator	{PackageName} Cab	Object used to interface at a low level with the elevator system. Use this object to define the floor list. Destination Objects: {PackageName} ACO {PackageName} DSO
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{PackageName} HW	S300 Hardware Module object that represents a generic hardware module.
S300 Reader Terminal	{PackageName} Reader	Object representing the elevator's reader. Source Objects: {PackageName} ACO {PackageName} DSO Connector: Reader 1 The connector is selected when inserting a package. Available connectors are determined by the S300 hardware module used.

*Non-Default Attributes**Table 6-32: Non-Default Attributes for the JCI_x_Elevator Template*

Object	Attribute	Non-Default Value
{PackageName} ACO	Primary Reader Object	{PackageName} Reader
	Door Sequence Object	{PackageName} DSO
	Set 1 First Identifier Format	10000 - Default

Table 6-32: Non-Default Attributes for the JCI_x_Elevator Template

Object	Attribute	Non-Default Value
{PackageName} Cab	Access Control Object	{PackageName} ACO
	Door Sequence Object	{PackageName} DSO
	Low Level Mode	Activate All Allowed Floors
{PackageName} DSO	Reader Terminal Object	{PackageName} Reader
	Timed Override Mode	Timed Override
{PackageName} Reader	Portal Contact Connected	Deselected
	Aux Input Connected	Not Connected

JCI_KDM_with-ACO

This template adds objects for a Keypad/Display Module (KDM) hardware module and an Access Control object configured for accessing the intrusion system via the KDM's built-in keypad.

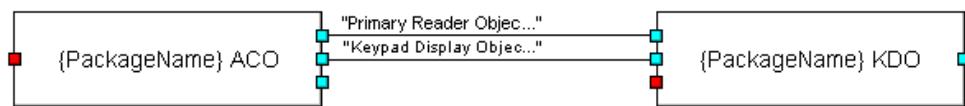
Object Diagram

Figure 6-33: Object Diagram for the JCI_KDM_with-ACO Template

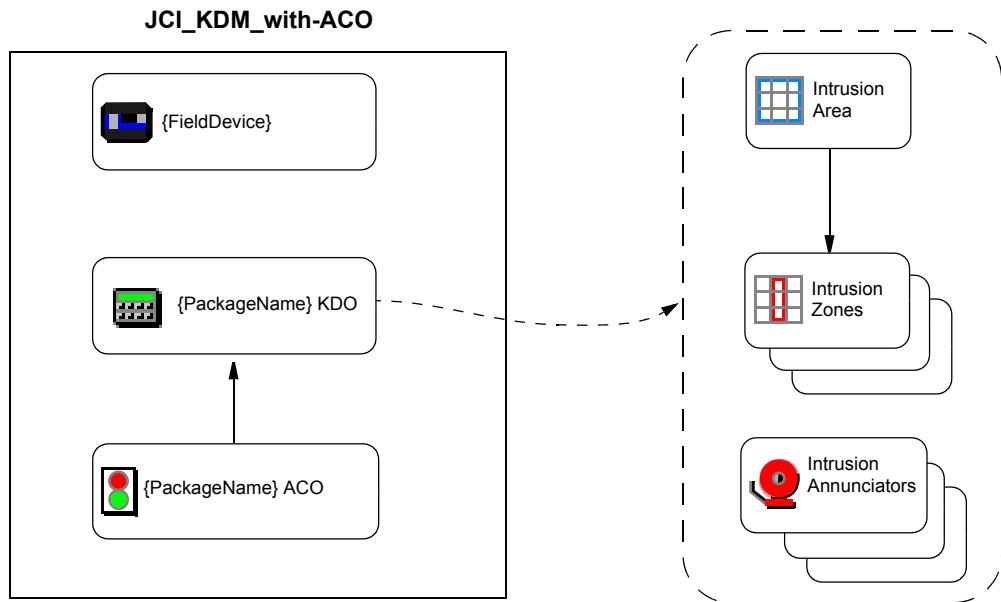
Graphical Representation

Figure 6-34: Graphical Representation of the JCI_KDM_with-ACO Template

Object List and Description

Table 6-33: Object List for the JCI_KDM_with-ACO Template

Type	Name	Description
Access Control	{PackageName} ACO	<p>Provides access control logic for the KDM, enabling operators to access the intrusion system via the KDM's keypad. Can also be used for granting access to an entity who has entered a Card ID code with the KDM's keypad.</p> <p>Destination Objects: {PackageName} KDO (assigned to both <i>Primary Reader Object</i> and <i>Keypad Display Object</i> attributes)</p>
S300 Trunk	S300	Represents the S300 bus for the S300 hardware modules and their input and output points.
S300 Hardware Module	{FieldDevice}	S300 Hardware Module object that represents the Intrusion Keypad/Display hardware module.
Intrusion Keypad/Display	{PackageName} KDO	Interfaces to the Intrusion Keypad/Display module and allows authorized users to control the assigned Intrusion Area, Intrusion Zone, and Intrusion Annunciator objects.

Non-Default Attributes

Table 6-34: Non-Default Attributes for the JCI_KDM_with-ACO Template

Object	Attribute	Non-Default Value
{PackageName} ACO	Primary Reader Object	{PackageName} KDO
	Keypad Display Object	{PackageName} KDO
	Set 1 First Identifier Format	00001 - Card ID

Legacy Templates

JCI Legacy Templates were provided in previous releases of the P2000 SCT software. They do not follow the JCI Standard Template naming convention, do not include package graphics, and are not immediately available using the **Item>Import Item** menu option. The *.ZIP files of these templates are available on the **Firmware Resources CD** in the **Legacy Templates** directory.

To use JCI Legacy Templates, you must manually copy the associated *.ZIP file to the following location on the P2000 server:

Local Disk\Documents and Settings\All Users\Application Data\Johnson Controls\MetasysIII\DatabaseFiles

Once you copy the desired *.ZIP file, you can import the JCI Legacy Template using the **Item>Import Item** menu option.

The following JCI Legacy Templates are available:

Table 6-35: JCI Legacy Templates

Template Name	Description
Bi-directional_Man_Trap	Adds objects associated with implementing a bi-directional man-trap application consisting of two fully-configured doors (with four readers total) controlled by two RDR2S modules.
Eight_Door_Occupancy	Adds eight Access Control objects, eight Door Sequence objects, and one Occupancy object for configuring four occupancy spaces (using four entry doors and four exit doors). This template does not include hardware and I/O objects. These must be added to the application.
Eight_Zone_Area_Keypad_Annun	Adds intrusion objects for an application consisting of eight zones, one area, and one keypad/display module. The zones are grouped into a single area.

Table 6-35: JCI Legacy Templates

Template Name	Description
Eight_Zone_Area_No_Keypad	Adds eight Intrusion Zone objects and one Intrusion Area object for use in an intrusion application. It does not include hardware, such as the Keypad/Display Module, and I/O objects. These must be added to the application.
RDR2_Two_Door	Adds all objects associated with implementing two-door control on a legacy RDR2 module (two readers installed for use with two separate doors – not an Entry/Exit door).
RDR2S_Entry_Exit	Adds objects for the RDR2S module with Reader 1 and Reader 2 enabled for use with a single door (entry/exit).
RDR2S_One_Door	Adds objects for the RDR2S module with Reader 1 enabled for use with a single, fully configured door.
RDR2S_One_Door_App	Adds one Access Control object and one Door Sequence object for configuring a single door. It does not include hardware and I/O objects. These must be added to the application.
RDR2S_One_Door_IO	Adds objects for an RDR2S module for use with one door, consisting of I/O and device object definitions for one door contact and one request to exit (REX) device, including four general purpose inputs and five general purpose outputs. It does not include application objects, such as the Door Sequence object or Access Control object.
RDR2S_SI_O_ONLY	Adds objects for an RDR2S module with no readers enabled (SI/O Mode).
RDR2S_Two_Door	Adds objects for an RDR2S module with Reader 1 and Reader 2 enabled for use with two separate, fully configured doors (not an entry/exit door).
RDR2S_Two_Door_App	Add two sets of Access Control objects and Door Sequence objects for configuring two doors. It does not include hardware and I/O objects. These must be added to the application, as needed.
RDR2S_Two_Door_IO	Adds objects for an RDR2S module for use with two doors, consisting of I/O and device object definitions for two door contacts and two request to exit (REX) devices, including two general purpose inputs and two general purpose outputs. It does not include application objects, such as the Door Sequence object or Access Control object.

Table 6-35: JCI Legacy Templates

Template Name	Description
S300_I16	Adds objects a fully configured S300-I16 unsupervised input module with 16 Security Supervised Input objects for use in an access control or intrusion detection application.
S300_IO8	Adds objects for a fully configured S300-IO8 input/output module with eight Security Supervised Input objects and eight Security Binary Output objects for use in an access control or intrusion detection application.
S300_SI8	Adds objects for a fully configured S300-SI8 input module with eight Security Supervised Input objects for use in an access control or intrusion detection application.
S300_SIO8	Adds objects for a fully configured S300-SIO8 input/output module with eight Security Supervised Input objects and eight Security Binary Output objects for use in an access control or intrusion detection application.
SIO8_Eight_Zone_Outputs	Adds objects for a fully configured S300-SIO8 input/output module with eight Security Supervised Input objects, eight Security Binary Output objects, and eight Intrusion Zone objects for use in an intrusion detection application.
Thirty_Two_Zone_Area_Annun	Adds intrusion objects for an application consisting of 32 zones, 1 area, and 1 Intrusion Announcer object for alarm annunciation. All 32 zones are grouped into a single area.

CREATING JOB-SPECIFIC TEMPLATES

This chapter describes in detail how to create Job-Specific Templates to define various security logic functions using the P2000 SCT. Included are examples for defining access control and intrusion detection functions.

CREATING JOB-SPECIFIC TEMPLATES

A Job-Specific Template is simply a template that has been copied and modified to meet the needs of a particular access control or intrusion application. Creating Job-Specific Templates enables you to automate much of the repetitive steps required to create and apply multiple similar applications.



This chapter describes how to modify an JCI template to design a Job-Specific Template. However, once you modify an original, imported template, you cannot re-import the same template unless you delete or overwrite the existing template in the Templates folder. For this reason, **do not** modify an original template. Before making any modifications to a template, make a copy, give the new template a name that more closely represents the application you are configuring, and modify the new template accordingly. For information on copying items in P2000 SCT, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

► **To create and use a Job-Specific Template, perform the following series of steps:**

1. Determine which one of the existing templates matches the type of Job-Specific Template you wish to create.
The P2000 SCT comes with a set of factory-created templates (see “Chapter 6: JCI Standard Templates”). Other templates may be imported. If no template matches the needs of a particular door, the existing templates can be used as a starting point to create your own templates.
2. Copy the applicable template. See page 7-2.
3. Adapt the new template according to the job. See page 7-3.
4. Update the template graphics, if desired.
You can update the template's graphics file to include a new GIF file for the new template (e.g. Job12345_x_CICO.GIF). For more information on package graphics, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

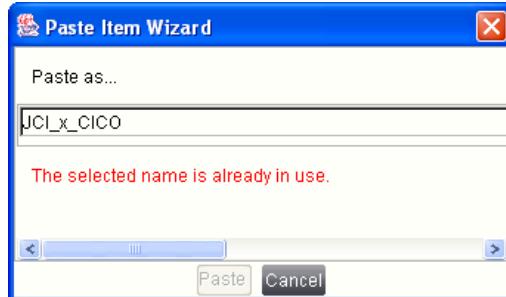
5. Insert a package based on the Job-Specific Template, as needed.

First create a single package with the job-specific template, and test it to ensure that all job-wide attributes are correct. Any errors you encounter can be fixed inside the package until the door works as desired. Then make the same corrections to the job-specific template, so that all subsequently created doors will not require any rework.

Copying Existing Templates as Job-Specific Templates

► To copy a template:

1. In the Navigation Pane, select the template to be copied.
2. From the menu bar, select **Edit>Copy**.
3. In the Navigation Pane, select **Templates Folder**.
4. From the menu bar, select **Edit>Paste**. The Paste Item Wizard appears.



5. Enter a new name for the template.

The names of all JCI Standard Templates, excluding Legacy Templates, start with the job prefix **JCI**, followed by a hardware module descriptor, followed by the name of the application. The job prefix **JCI** should be replaced with a job-specific prefix of up to 8 characters. You may assign a longer job prefix, but it may lead to the truncation of the template name after 32 characters.



6. Click **Paste**.

Adapting the New Template According to the Job

Some attributes are always job specific, such as the *Set 1 First Identifier Format* attribute of an Access Control object. Each job typically uses the same card format for all doors. Any exceptions can be resolved after the database is created.

Double-check the Job-Specific Templates to ensure that all job-wide attributes are correct.

Prime Candidates for Job-Specific Attributes

The following attributes are examples of the most common job-specific attributes:

Table 7-1: Common Job-Specific Attributes

Object	Common Job-Specific Attributes
Access Control	Set 1 First Identifier Format Set 1 PIN Required PIN Digits PIN Duress
Door Sequence	Anti-Tailgating Access Time Shunt Time
S300 Reader Terminal	Offline Facility Code Offline Card Type

If you plan to use other features on a job-wide basis, such as the **Alternate Access** feature, set the corresponding attributes to their job-specific values.

Selecting Package Attributes

Attributes that typically vary by door, such as the *Timed Override Mode* attribute of a Door Sequence object, can be designated as **Package Attributes**. If you designate an attribute as a package attribute, the P2000 SCT will prompt you for to define the value of this attribute for every door being created when inserting a package.

For more information on package attributes, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

ACCESS CONTROL APPLICATION EXAMPLES

This section describes various access control application examples, most of which are common security applications used in the field. Included with these examples are instructions on how to configure the applications using the P2000 SCT.

The examples described in this section also require host software configuration, which is not described in this manual. For information on using the host software, refer to the *P2000AE Software User Manual*.

Card-In Door: Single Reader or Keypad with REX

Application Description

A portal is a door, gate, or other opening by which an entity can enter or leave a protected area of the facility. These areas are the key focus of the security management system, as they must be controlled properly to allow the entry and egress of authorized entities, while at the same time preventing access to unauthorized individuals. This section describes a portal entry application that can be configured using the P2000 SCT.

The most common access control application utilizes the single reader or keypad device to allow an authorized entity to enter a controlled area and a REX device for egress. The following scenario helps describe how this application functions:

1. The entity presents an access badge identifier to a badge reader, or enters a PIN identifier on a keypad.
2. The P2000 SMS authenticates the entity and grants him/her access to the controlled area.
3. The door strike or magnetic lock unlocks the door. The alarm is shunted during the door open timer.
4. The door contact allows the P2000 SMS to monitor how long the door is held open (propped). The system can generate an alarm if the door is held open longer than the maximum time allowed.
5. When the entity leaves, the REX device (e.g. motion detector) prompts the P2000 SMS to unlock the door and shunt the alarm.

Using an Existing Template

Depending on which hardware module you plan to use, copy one of the following templates:

- JCI_x_Card-In (see page 6-40)
- JCI_RDR2SA_Card-In (see page 6-33)
- JCI_RDR2S_Card-In (see page 6-24)

NOTE

Depending on your exact application requirements, your template may require modifications.

Single Portal Entry with Two Readers

Application Description

This application uses two entry (ingress) readers, either of which may be used to gain access through a single portal. The following scenario helps describe how this application functions:

1. A vehicle entry point into a parking lot is controlled by a gate operator.
2. To prompt the P2000 system to open the gate, an entity must present his/her identifier to one of two readers mounted in front of the gate.
3. One reader is installed at a height specifically for drivers of standard vehicles (sedans, compact cars, etc.).
4. The other reader is installed at a height specifically designed for drivers of higher profile vehicles (e.g. trucks, large sport utility vehicles, etc.).

Using an Existing Template

The following instructions enable you to define a Single Portal Entry with Two Readers application using an RDR2S-A hardware module.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33), and follow the instructions in this section to modify the copied template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-2: Object Additions (Single Portal Entry with Two Readers Application)

Object	Name ¹	Description
S300 Reader Terminal	{PackageName} Reader2	Object representing the second reader. Source Object: {PackageName} ACO ² Parent Hardware: {FieldDevice} Connector: DATA0 / DATA1 (bottom) Portal Contact Connected: Deselect check box

1. The names in this column are suggestions. You may enter any name you wish.
2. See Table 7-3.

Object Modifications

Modify the following object(s) in the template:

Table 7-3: Object Modifications (Single Portal Entry with Two Readers Application)

Object	Attribute	Modification
{PackageName} ACO	Secondary Reader Object	Select the {PackageName} Reader2 object.

Object Deletions

No objects need to be deleted from the JCI_RDR2SA_Card-In template.

Object Hierarchy

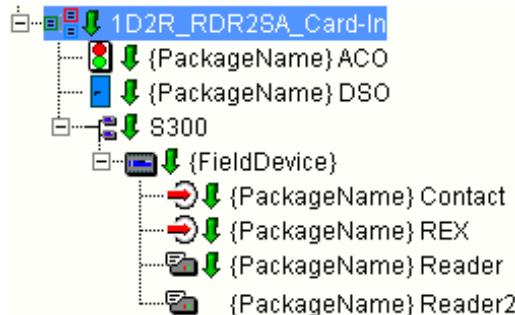


Figure 7-1: Object Hierarchy for Single Portal Entry with Two Readers Application

Application Notes

The RDR2S-A can support a maximum of two readers. As a result of adding an additional reader, Reader 2 Contact, Reader 2 REX, Reader 2 Red LED, Reader 2 Green LED, Reader 2 Shunt, and Reader 2 Strike input and output terminals are no longer available as general purpose I/Os.

The Access Control object must be linked to the new S300 Reader Terminal object (Reader2), which is the Secondary Reader object.

The {PackageName} DSO object does not have to be linked with the second S300 Reader Terminal object – {PackageName} Reader2. When using the RDR2S-A hardware module, the Door Sequence object connects to the reader that is wired to the door contact. In this example, that reader is {PackageName} Reader. Reader 2 is not wired to the door contact, and therefore is not connected to the Door Sequence object.

Portal Entry: Card Reader and Keypad Combination

Application Description

This application is similar to the “Single Portal Entry with Two Readers” application (see page 7-5). In this application, however, the reader module can read identifier badges and has a built-in keypad for entering Personal Identification Number (PIN) codes. An entity must present an access badge identifier to the reader and enter a PIN identifier on the reader’s keypad before entering a protected area.

Using an Existing Template

Depending on which hardware module you plan to use, copy one of the following templates:

- JCI_x_Card-In (see page 6-40)
- JCI_RDR2SA_Card-In (see page 6-33)
- JCI_RDR2S_Card-In (see page 6-24)

NOTE

Depending on your exact application requirements, your template may require modifications.

Application Notes

There is little or no difference between the Single Portal Entry with Two Readers application and this application when configuring them in the P2000 SCT. When using a combination reader/keypad device, you only need a single S300 Reader Terminal object for the device in the P2000 SCT.

Portal Entry and Exit (Card-In-Card-Out)

Application Description

For a description of this type of application, see “Portal Entry and Exit (Card-In-Card-Out)” on page 4-16.

Using an Existing Template

The JCI_RDR2SA_Card-In-Card-Out template enables you to define this application using an RDR2S-A module. See page 6-33.

NOTE

Depending on your exact application requirements, your template may require modifications.

Emergency Exit Portal

Application Description

An emergency exit portal provides an egress option in the event of an emergency and signals the person attempting to exit that the door should only be opened in an emergency. The type of egress hardware commonly used with this application is a crash bar. See “Panic or Crash Bars” on page 4-15 for more information.

The following scenario helps describe how this application functions:

1. The occupant attempts to leave through an emergency exit portal.
2. He pushes the crash bar, which is wired to an input terminal on the security field device.
3. The field device is also wired to an output device, in this case, a sounder (buzzer).
4. The field device activates the sounder.
5. The field device keeps the door locked for 5 seconds while the buzzer sounds to warn the occupant.
 - If the person is attempting to exit during a non-emergency, and he releases the crash bar without opening the door before the 5 second period elapses, the door remains locked and the sounder is deactivated as soon as he releases the crash bar.
 - If the person holds down the crash bar for the full 5 seconds, the door will unlock and the sounder will be deactivated as soon as he releases the crash bar.

Using an Existing Template

The following instructions enable you to define an Emergency Exit Portal application using an S300-SIO8 module. This application requires one general purpose input, two general purpose outputs, and an Interlock object to link the I/Os.

Start by importing and copying the JCI_SIO8_Full-IO template (see page 6-12), and follow the instructions in this section to modify the copied template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

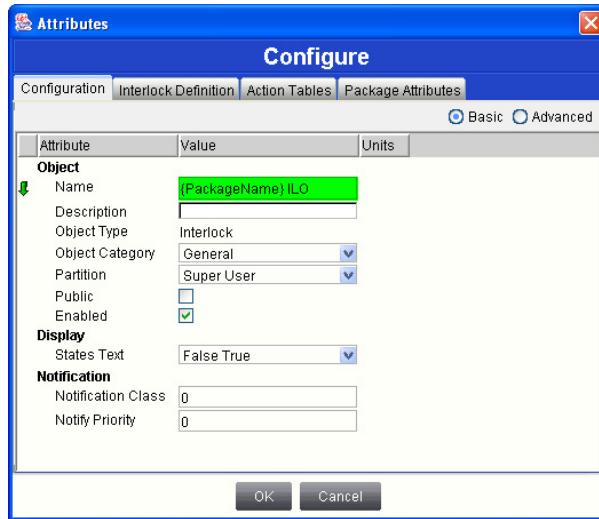
Table 7-4: Object Additions (Emergency Exit Portal Application)

Object	Name ¹	Description
Interlock	{PackageName} ILO	Links the input and output devices for control purposes. An input in an alarm state can activate an output (i.e. sound a buzzer).

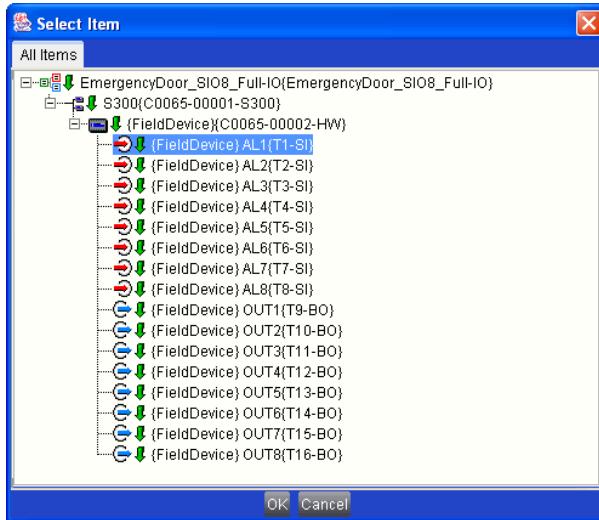
1. The names in this column are suggestions. You may enter any name you wish.

➤ To add the Interlock object:

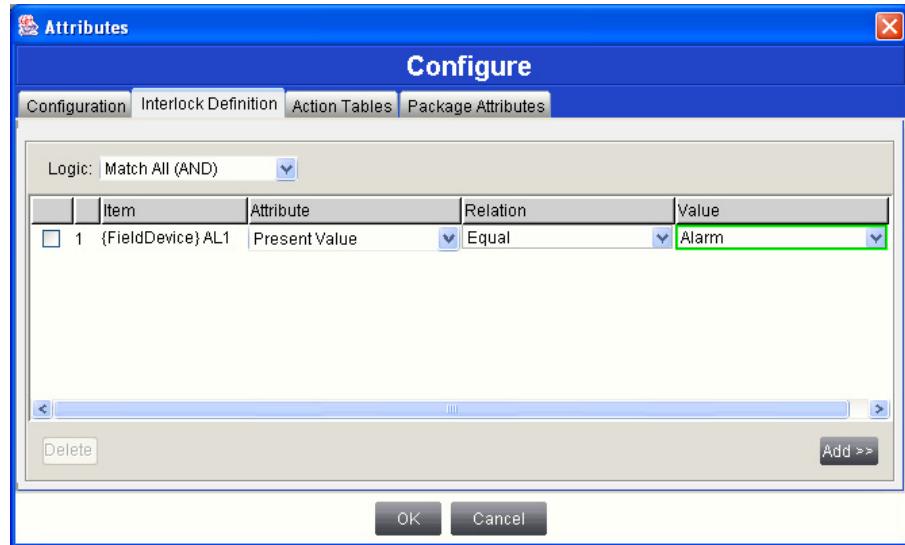
1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.
For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.
2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} ILO, in the **Name** field.



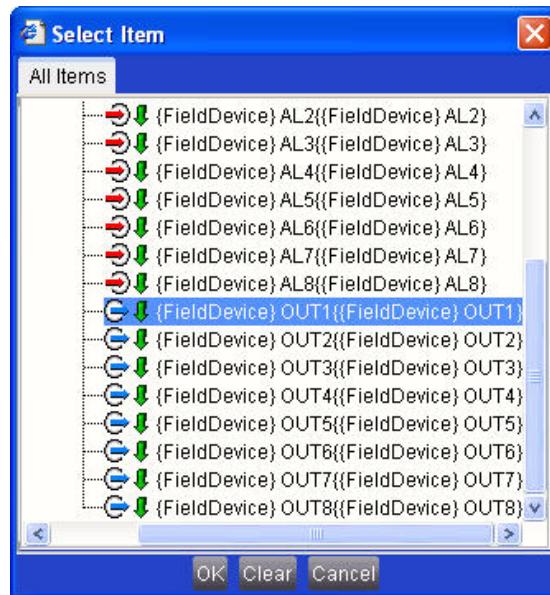
3. Select the **Interlock Definition** tab.
4. Verify the **Match All (AND)** option is selected in the **Logic** drop-down list.
5. Click **Add**.
6. Expand the object tree, select **{FieldDevice} AL1**, and click **OK**.



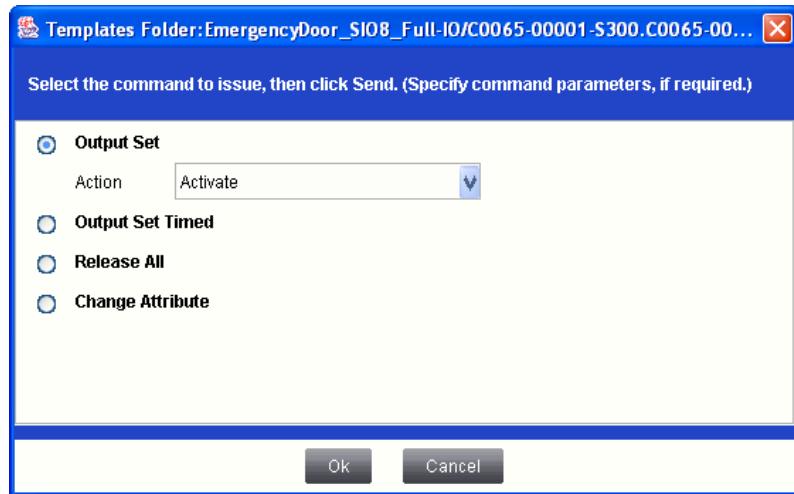
- In the Logic table, select **Alarm** under the **Value** column for {FieldDevice} AL1.



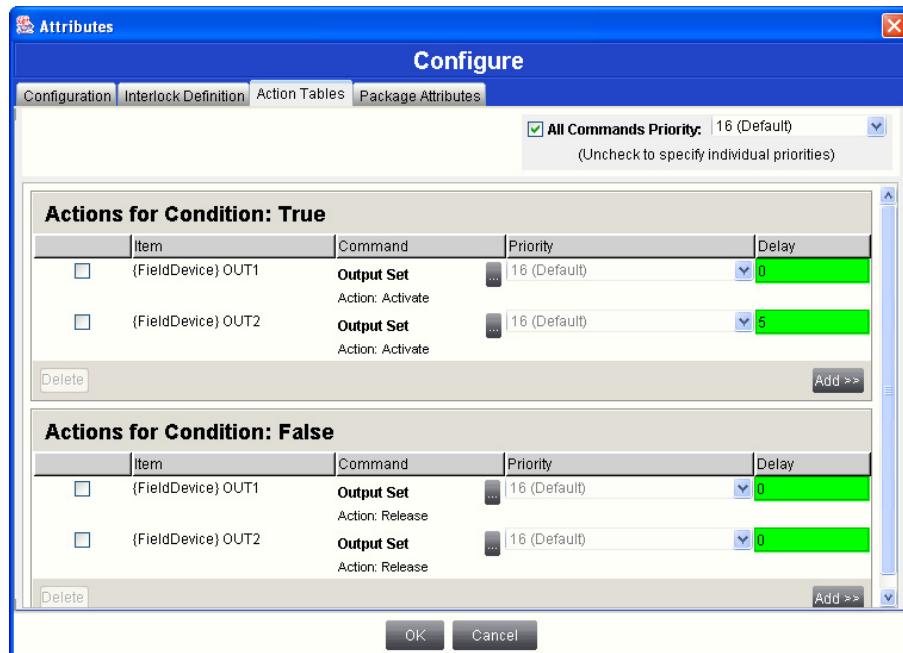
- Select the **Action Tables** tab.
- In the Actions for Condition: True area, click **Add**.
- Select {FieldDevice} OUT1 and click **OK**.



- Under the **Command** column, click the **Browse** button [...].
- Select the **Output Set** radio button. Then select **Activate** in the **Action** drop-down list and click **OK**.



13. Repeat steps 9-12, but select {FieldDevice} OUT2 and enter a **Delay** time of **5** seconds.
14. Add Actions for Condition: False entries for the same outputs, but with the following changes:
 - The **Output Set** command value must be set to **Release** for OUT1 and OUT2.
 - The **Delay** time for OUT2 must be **0** seconds.



15. Click **OK**.
16. Click **Save**.

Object Modifications

No object modifications are required.

Object Deletions

No objects need to be deleted from the JCI_SIO8_Full-IO template.

Object Hierarchy

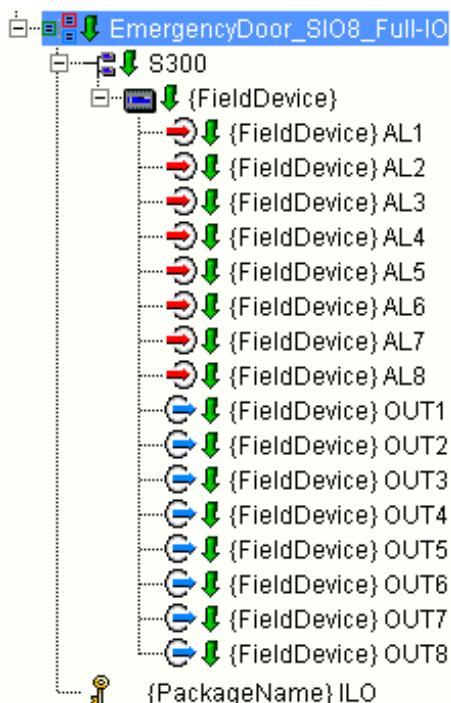


Figure 7-2: Object Hierarchy for Emergency Exit Portal Application

Application Notes

The S300-SIO8 module supports up to eight general purpose inputs and eight general purpose outputs. This module allows you to control up to four emergency exit portals.

The inputs and outputs for the Emergency Exit Portal application have the following functions:

AL1 / OUT1 – The AL1 input is wired to the door contact and has a default debounce time of 100 ms (0.1 second). When the door is opened, AL1 receives the input from the contact, and by way of the Interlock object, causes the OUT1 output to sound a buzzer. There is a 100 ms delay from the time the door is opened to the time OUT1 activates the buzzer.

AL1 / OUT2 – The Interlock object delay function allows you to set the AL1 with a delay time of 5000 ms (5.0 seconds) before it activates OUT2 (unlock door). When the door is opened, AL1 receives the input from the contact, and by way of the

Interlock object, causes the OUT2 output to unlock the door once the 5 second delay period elapses. There is a 5 second delay from the time the door is opened to the time OUT2 unlocks the door.

Reader with Tamper Switch

Application Description

This type of reader includes a tamper switch that detects when the reader housing has been opened, thereby signaling the P2000 SMS that someone may be tampering with the reader to gain access to an unauthorized area of the facility.

The following scenario helps describe how this application functions:

1. An unauthorized individual unlawfully breaks open the housing to a reader in an attempt to tamper with the controls and gain access to a controlled area.
2. The field device senses that the switch's state has changed.
3. A notification is generated in the P2000 SMS.
4. The field device is also wired to an output device, in this case, a sounder (buzzer).
5. The field device activates the sounder.
6. The buzzer continues to sound until an authorized individual manually deactivates it.

Using an Existing Template

The following instructions enable you to define a Reader with Tamper Switch application using an RDR2S-A module.

NOTE

JCI x-Templates with a TAMP indicator in their name include a Security Supervised Input object dedicated as a reader tamper input; however, all field points for x-Templates are assigned when inserting the template as a package.

Apart from the standard door object requirements, this application requires one general purpose input, one general purpose output, and an Interlock object to link the I/O objects.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33) and follow the instructions in this section to modify the copied template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

NOTE

Add the Interlock object last, since this object references the Security Supervised Input and Security Binary Output objects you are adding.

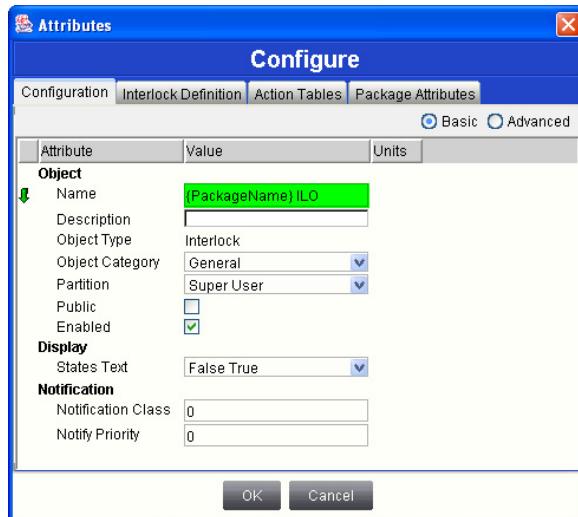
Table 7-5: Object Additions (Reader with Tamper Switch Application)

Object	Name ¹	Description
Security Supervised Input	{PackageName} Tamper	Monitors the reader's tamper switch. When an alarm occurs, {PackageName} OUT01 activates a local alarm device, such as a sounder. Parent Hardware: {FieldDevice} Connector: Reader 1 Tamper A
Security Binary Output	{PackageName} Alarm	When a reader tamper alarm occurs, this output activates a local alarm device, such as a sounder. Parent Hardware: {FieldDevice} Connector: Reader 2 Green LED ²
Interlock	{PackageName} ILO	Links the input and output devices for control purposes. An input in an alarm state can activate an output (i.e. sound a buzzer).

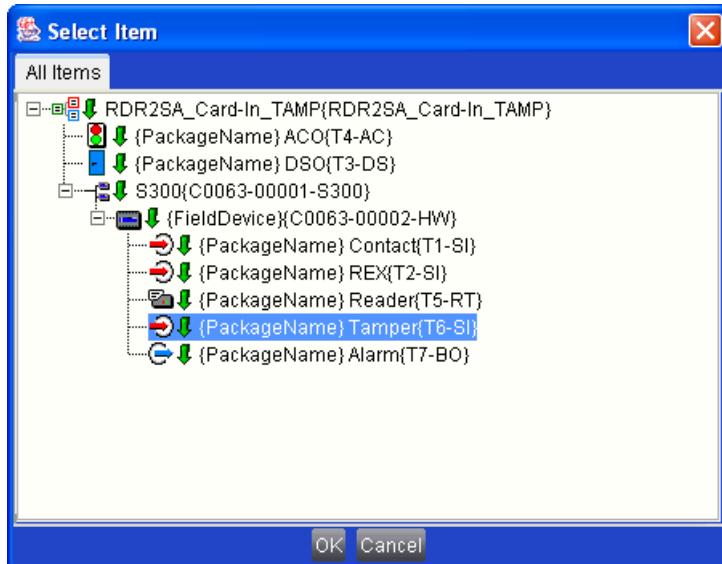
1. The names in this column are suggestions. You may enter any name you wish.
2. Selecting an output from the Reader 2 terminal is the safest approach, since this template is designed only for a single reader. See "How to "Steal" Unused RDR2S-A Field Points" on page 19.

► **To add the Interlock object:**

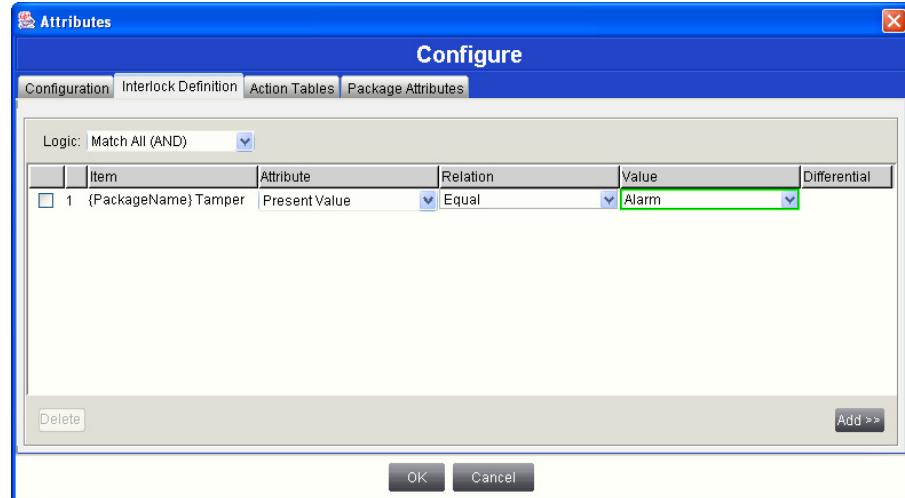
1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.
For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.
2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} ILO, in the **Name** field.



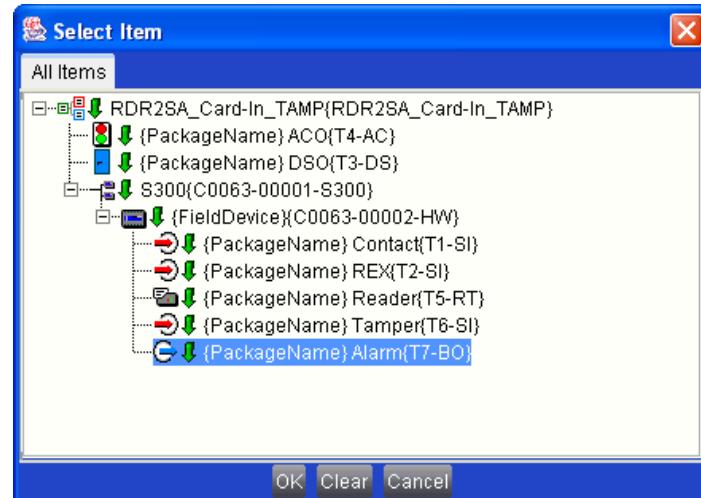
3. Select the **Interlock Definition** tab.
4. Verify the **Match All (AND)** option is selected in the **Logic** drop-down list.
5. Click **Add**.
6. Expand the object tree, select **{PackageName} Tamper**, and click **OK**.



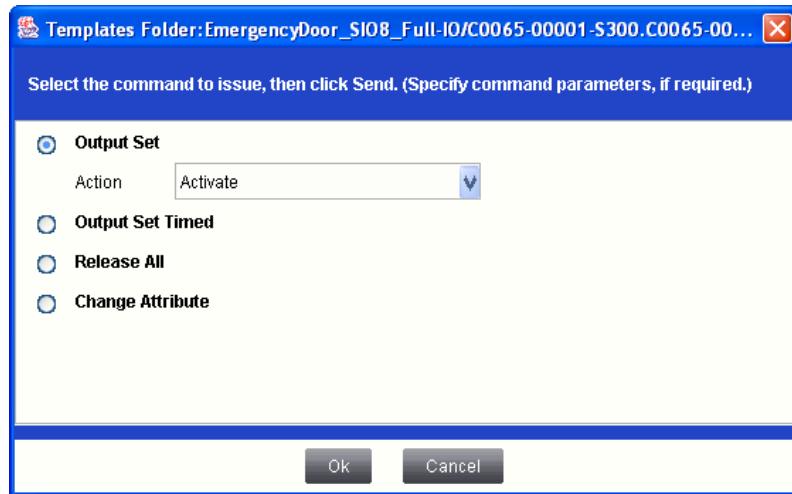
7. In the Logic table, select **Alarm** under the **Value** column for {PackageName} Tamper.



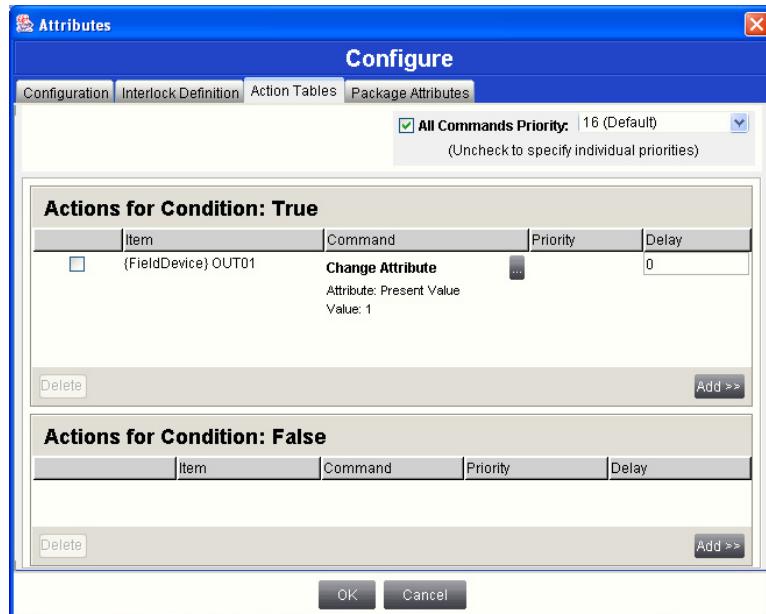
8. Select the **Action Tables** tab.
9. In the Actions for Condition: True area, click **Add**.
10. Select {PackageName} Alarm and click **OK**.



11. Under the **Command** column, click the **Browse** button [...].
12. Select the **Output Set** radio button. Then select **Activate** in the **Action** drop-down list.



13. Click **OK**.



14. Click **OK**.

15. Click **Save**.

Object Modifications

No object modifications are required.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

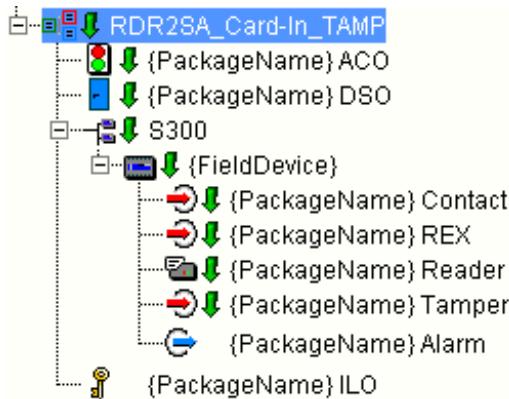


Figure 7-3: Object Hierarchy for Reader with Tamper Switch Application

Application Notes

The input and output for the Reader with Tamper Switch application have the following functions:

Reader 1 Tamper A / Reader 2 Green LED – The Reader 1 Tamper A input is wired to the reader tamper switch. When the reader housing is opened, Reader 1 Tamper A receives the input from the reader, and by way of the interlock object, causes the Reader 2 Green LED output to sound a buzzer.

Portal with Timed Anti-Passback

This section describes how to develop an application that incorporates the anti-passback time rule. This rule is described on page 4-19.

The following scenario helps describe how this application functions:

1. An entity presents a valid identifier badge to a reader so he may be granted access through a door.
2. This door has an anti-passback time of ten minutes.
3. The security system authenticates the entity and unlocks the door.
4. The Access Control object reports the access event to the Anti-Passback object, which notes the time of entry.
5. The entity passes his identifier badge to his friend (an unauthorized individual) through a facility window.
6. His friend presents the badge to the reader in an attempt to gain unauthorized access to the facility.
7. Three minutes have elapsed since the identifier was last presented to the reader.
8. The system denies access to the unauthorized individual.

Using an Existing Template

The following instructions enable you to define a Portal with Timed Anti-Passback application using an RDR2S-A module. This application requires one Anti-Passback object, in addition to the other objects required for door control.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-6: Object Additions (Portal with Timed Anti-Passback Application)

Object	Name ¹	Description
Anti-Passback	{PackageName} AP Time	<p>Consulted for anti-passback information. Before the Access Control object grants an entity access, it verifies with the Anti-Passback object whether the entity has violated an anti-passback rule.</p> <p>Anti-Passback Requirements: Elapsed time only</p> <p>Entry Anti-Passback Time: 600 seconds</p>

1. The names in this column are suggestions. You may enter any name you wish.

Object Modifications

Modify the following object(s) in the template:

Table 7-7: Object Modifications (Portal with Timed Anti-Passback Application)

Object	Attribute	Modification
{PackageName} ACO	Anti-Passback Check (Access tab)	Enforce, only when operational Violations will be reported to the host software and access will be denied, but only if the system is operational.
	Anti-Passback In Object (Access tab)	Select the Anti-Passback object (see Table 7-6).

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

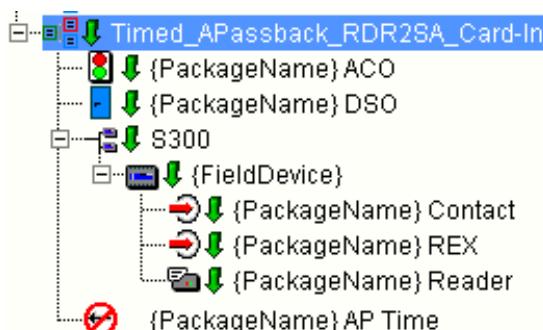


Figure 7-4: Object Hierarchy for Portal with Timed Anti-Passback Application

One Portal with Entry/Exit Anti-Passback

This section describes how to develop an application that incorporates the anti-passback entry/exit rule with a single entry/exit portal. This rule is described on page 4-19. Information for configuring both types of entry/exit anti-passback applications (hard and soft) is also provided.

The following scenario helps describe how these applications function:

1. An entity presents a valid identifier badge to an entry reader so he may be granted access through Door A.
2. The security system authenticates the entity and unlocks the door.
3. The Access Control object reports the access event to the Anti-Passback object, which notes the location.

4. The entity should exit by presenting a valid access badge identifier to the exit reader of Door A.
5. However, he decides to exit the building through a different door without presenting a valid identifier to exit the facility.
6. When he returns the next day and presents his identifier to the entry reader of Door A:
 - **Hard Entry/Exit Rule:** The system *denies* the entity access to the facility and reports the violation to the host software.
 - **Soft Entry/Exit Rule:** The system *grants* the entity access to the facility, but still reports the violation to the host software.

Using an Existing Template

The following instructions enable you to define an entry/exit anti-passback application using an RDR2S-A module. This application requires one Anti-Passback object, in addition to the other objects required for entry/exit door control.

Start by importing and copying the JCI_RDR2SA_Card-In-Card-Out template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-8: Object Additions (One Portal with Entry/Exit Anti-Passback Application)

Object	Name ¹	Description
Anti-Passback	{PackageName} AP Entry-Exit	Consulted for anti-passback information. Before the Access Control object grants an entity access, it verifies with the Anti-Passback object whether the entity has violated an anti-passback rule.

1. The names in this column are suggestions. You may enter any name you wish.

Object Modifications

Modify the following object(s) in the template:

Table 7-9: Object Modifications (One Portal with Entry/Exit Anti-Passback Application)

Object	Attribute	Modification
{PackageName} Ingress ACO	Anti-Passback Check (Access tab)	Monitor only (Soft Entry/Exit Rule) Violations will be reported to the host software and access will be granted.
	Anti-Passback In Object (Access tab)	Select the Anti-Passback object (see Table 7-8).
	Anti-Passback Transition (Access tab)	On Granted For a description of the other values, refer to the <i>Anti-Passback Object Manual</i> .
{PackageName} Egress ACO	Anti-Passback Check (Access tab)	Enforce, only when operational (Hard Entry/Exit Rule) Violations will be reported to the host software and access will be denied, but only if the system is not operational.
	Anti-Passback Out Object (Access tab)	Select the Anti-Passback object (see Table 7-8).
	Anti-Passback Transition (Access tab)	On Granted For a description of the other values, refer to the <i>Anti-Passback Object Manual</i> .

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

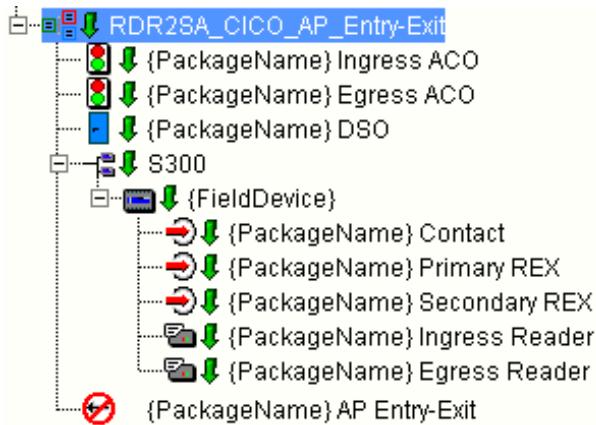


Figure 7-5: Object Hierarchy for One Portal with Entry/Exit Anti-Passback Application

Anti-Loitering Area

This section describes how to develop an application that incorporates the anti-loitering feature with a single Card-In-Card-Out portal. This feature is described on page 4-19.

The following scenario helps describe how the anti-loitering application functions:

1. An entity presents a valid access badge identifier to an entry reader so he may be granted access through a Card-In-Card-Out door.
2. The security system authenticates the entity and unlocks the door.
3. The Access Control object reports the access event to the Anti-Loitering object, which notes the time the entity has been granted access.
4. The *Anti-Loitering Time* attribute of the Anti-Loitering object is set to 600 seconds (10 minutes).
 - If the entity exits the area before the 10-minute time period elapses, the system **generates** an alarm.
 - If the entity has not exited the area when the 10-minute time period elapses, the system **does not generate** an alarm.

Using an Existing Template

The following instructions enable you to define an anti-loitering application using an RDR2S-A module. This application requires one Anti-Loitering object, in addition to the other objects required for Card-In-Card-Out door control.

Start by importing and copying the JCI_RDR2SA_Card-In-Card-Out template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-10: Object Additions (Anti-Loitering Area Application)

Object	Name ¹	Description
Anti-Loitering	{PackageName} Anti-Loitering	Consulted for anti-loitering information. If an entity has violated an anti-loitering rule, the system generates a notification. Enter 600 into the <i>Anti-Loitering Time</i> attribute. Default = 300 seconds (5 minutes)

1. The names in this column are suggestions. You may enter any name you wish.

Object Modifications

Modify the following object(s) in the template:

Table 7-11: Object Modifications (Anti-Loitering Area Application)

Object	Attribute	Modification
{PackageName} Ingress ACO	Anti-Loitering Transition (Access tab)	On Granted For a description of the other values, refer to the <i>Anti-Loitering Object Manual</i> .
	Anti-Loitering In Object (Access tab)	Select the Anti-Loitering object (see Table 7-10).
{PackageName} Egress ACO	Anti-Loitering Transition (Access tab)	On Granted For a description of the other values, refer to the <i>Anti-Loitering Object Manual</i> .
	Anti-Loitering Out Object (Access tab)	Select the Anti-Loitering object (see Table 7-10).

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

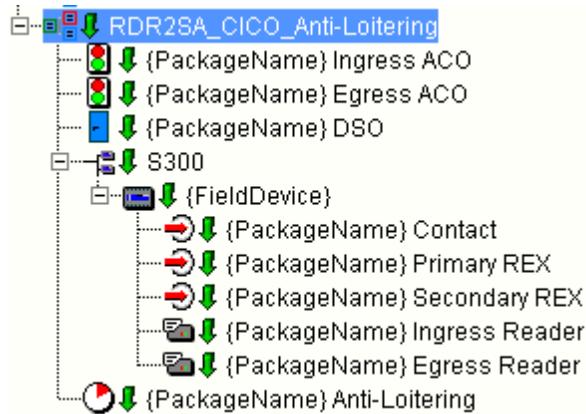


Figure 7-6: Object Hierarchy for Anti-Loitering Area Application

Occupancy: Parking Lot with “LOT FULL” Sign

Application Description

Parking lot access control applications consist of a gate operator and other accessories to control the entry and egress of vehicles. There are three main types of gate operators: slide, swing, and barrier. Barrier gate operators, which consist of a barrier arm that rotates up and down, are commonly used in high-traffic parking lots, as they allow vehicles to more quickly enter and exit the lot. Slide and swing gate operators typically take longer to open and close, and are thus better suited for low-to-medium traffic areas, such as residential properties.

In some cases, parking lot managers want to track the number of vehicles entering and exiting the lot to prevent vehicles from entering when there are not enough parking spaces to accommodate them, based on the current number of vehicles in the lot.

The P2000 Security Management System can track the number of vehicles entering and exiting a parking lot and can perform one or more actions (activate a “LOT FULL” sign; deny access to the next vehicle) as a result of someone attempting to enter a full lot.

The following scenario helps describe how this application functions:

1. A parking lot has one lane to enter the lot and one lane to exit the lot. Each lane is controlled by a gate operator.
2. The entry lane has a reader. The driver must present a valid access badge identifier to the reader in order for the gate operator open and allow access into the lot.
3. The reader is wired to an RDR2S-A hardware module of a P2000 Security Management System, which manages the entity information.

4. The entry lane has a vehicle loop just beyond the gate. This loop detects when a vehicle is present and allows the system to count the number of cars entering the lot.
5. The exit lane has a vehicle loop that detects when a vehicle is present and ready to exit the lot. The loop signals the gate operator to open the gate and allow the vehicle to exit.
6. There are 500 parking spaces in the lot.
7. Each time the *entry* loop detects a vehicle, the Occupancy counter *increases* by one.
8. Each time the *exit* loop detects a vehicle, the Occupancy counter *decreases* by one.
9. When the Occupancy counter reaches 500, the P2000 SMS triggers an output and an electronic “LOT FULL” sign is activated, informing motorists that there are currently no vacant spaces in the lot.
10. If a motorist attempts to enter the lot by presenting his/her identifier, and even if the identifier is valid, the P2000 SMS will deny access to the motorist until the occupancy counter falls below 500.

Using an Existing Template

The following instructions enable you to define an Occupancy: Parking Lot with “LOT FULL” Sign application using an RDR2S-A module. Apart from the standard door object requirements, this application requires one general purpose input, one general purpose output, three Interlock objects, and one Occupancy object.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33) and follow the instructions in this section to modify the copied template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

NOTE

Add the Interlock objects last, since these objects reference the Security Supervised Input, Security Binary Output, and Occupancy objects you are adding.

Table 7-12: Object Additions (Occupancy: Parking Lot with “LOT FULL” Sign Application)

Object	Name ¹	Description
Security Supervised Input	{PackageName} Entry Loop	Object representing the entry vehicle loop input device. Parent Hardware: {FieldDevice} Connector: Reader 1 Spare
Security Supervised Input	{PackageName} Exit Loop	Object representing the exit vehicle loop input device. Parent Hardware: {FieldDevice} Connector: Reader 2 Spare
Security Binary Output	{PackageName} Sign	Object representing the output that will trigger a “LOT FULL” sign. Parent Hardware: {FieldDevice} Connector: Reader 2 Green LED ²
Occupancy	{PackageName} Occupancy	Tracks the number of vehicles entering and exiting the parking lot. Maximum Occupancy: 500
Interlock	{PackageName} ILO	This object will turn on the “LOT FULL” sign when the occupancy counter reaches 500. See page 7-28 for configuration instructions.
Interlock	{PackageName} Entry ILO	This object increases the occupancy count by one when a vehicle is detected by the Entry Loop input device. See page 7-31 for configuration instructions.
Interlock	{PackageName} Exit ILO	This object decreases the occupancy count by one when a vehicle is detected by the Exit Loop input device. See page 7-34 for configuration instructions.

1. The names in this column are suggestions. You may enter any name you wish.
2. Selecting an output from the Reader 2 terminal is the safest approach, since this template is designed only for a single reader. See “How to “Steal” Unused RDR2S-A Field Points” on page 19.

► To add the {PackageName} ILO object:

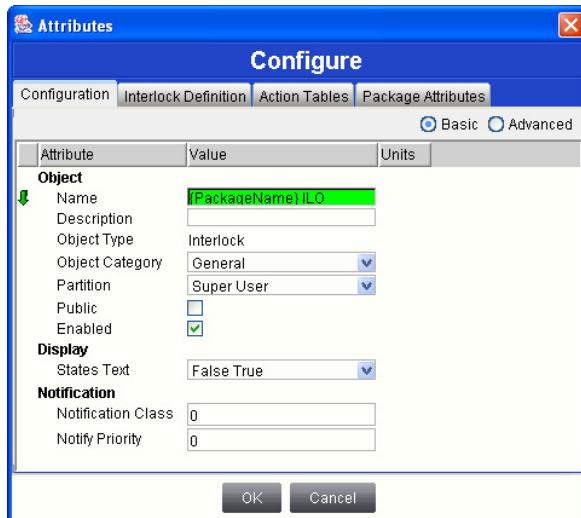
NOTE

Add and save the Security Supervised Input, Security Binary Output, and Occupancy objects for this template prior to performing the following instructions. See Table 7-12.

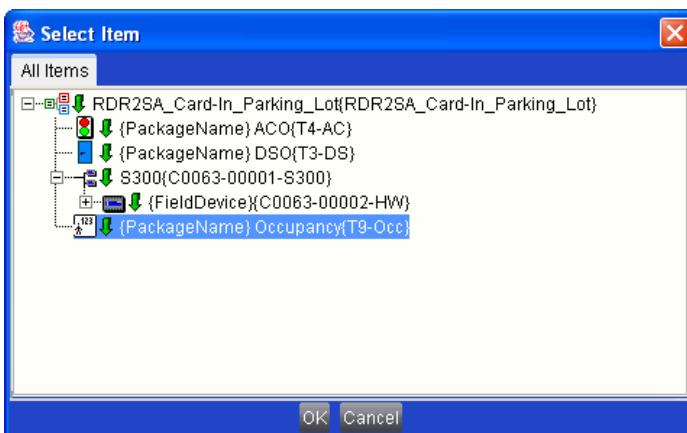
1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.

For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

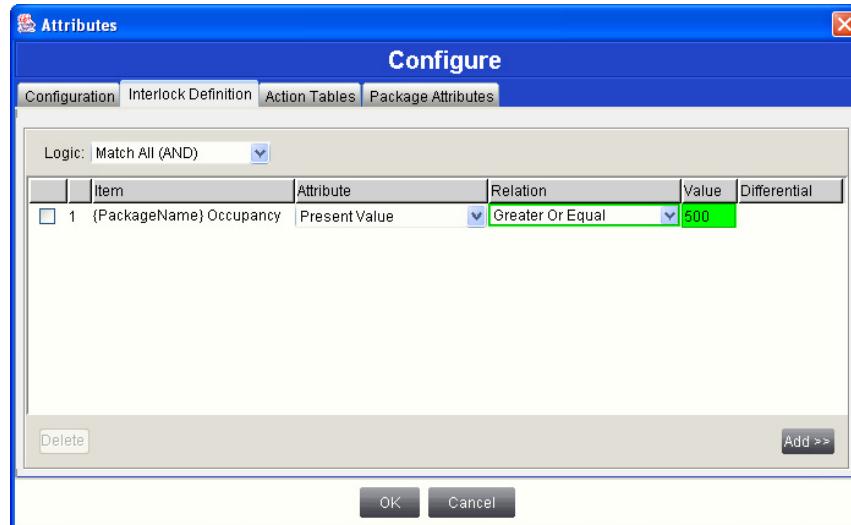
2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} ILO, in the **Name** field.



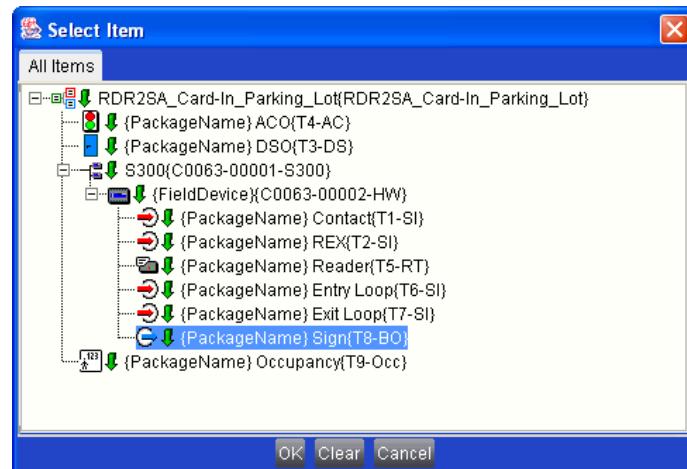
3. Select the **Interlock Definition** tab.
4. Verify the **Match All (AND)** option is selected in the **Logic** drop-down list.
5. Click **Add**.
6. Select **{PackageName} Occupancy** and click **OK**.



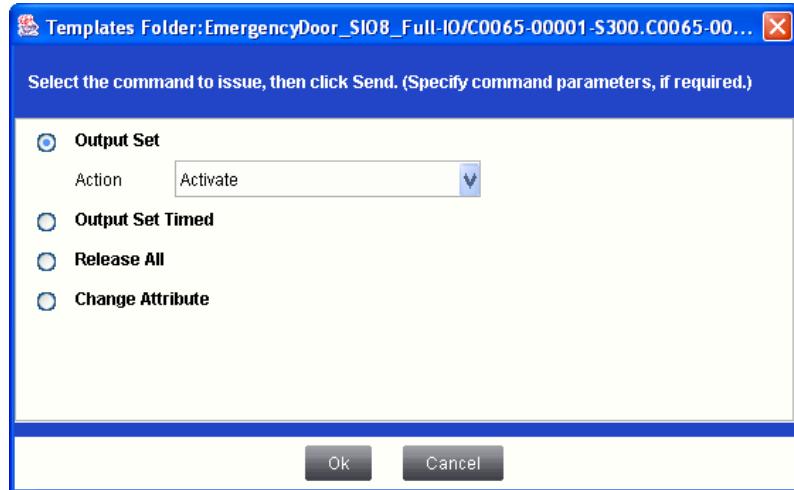
7. In the Logic table, select **Greater or Equal** under the **Relation** column.
8. Enter **500** under the **Value** column.



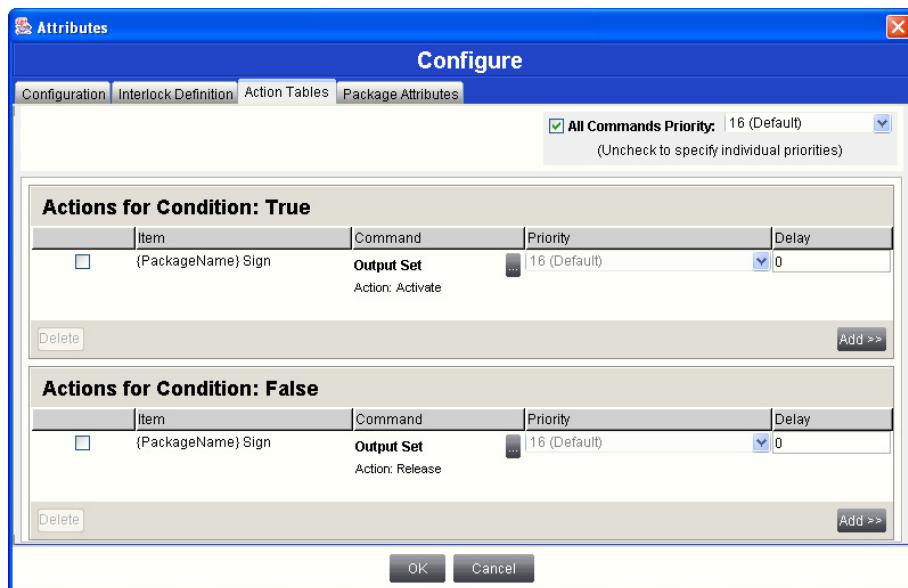
9. Select the **Action Tables** tab.
10. In the Actions for Condition: True area, click **Add**.
11. Expand the object tree, select **{PackageName} Sign**, and click **OK**.



12. Under the **Command** column, click the **Browse** button [...].
13. Select the **Output Set** radio button. Then select **Activate** in the **Action** drop-down list.



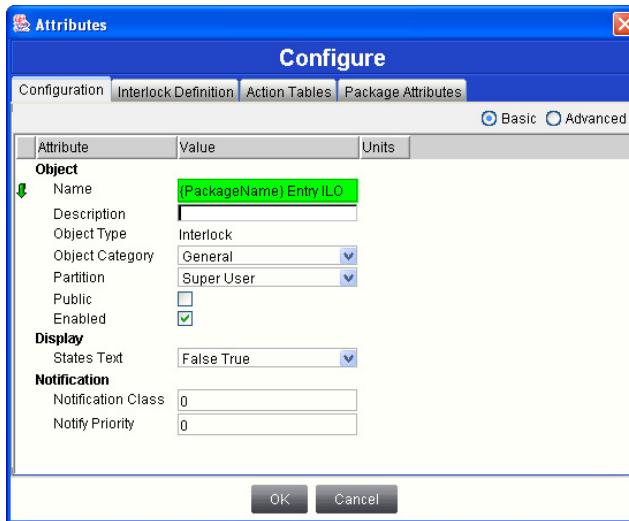
14. Click **OK**.
15. Under Actions for Condition: False, click **Add** and repeat steps 11-14, but select **Release** as the **Output Set** action.



16. Click **OK**.
17. Click **Save**.

➤ **To add the {PackageName} Entry ILO object:**

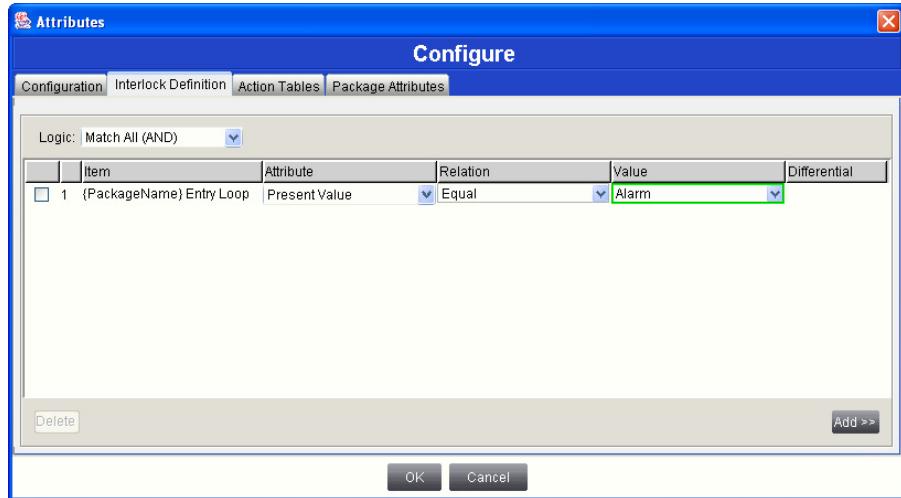
1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.
For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.
2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} Entry ILO, in the **Name** field.



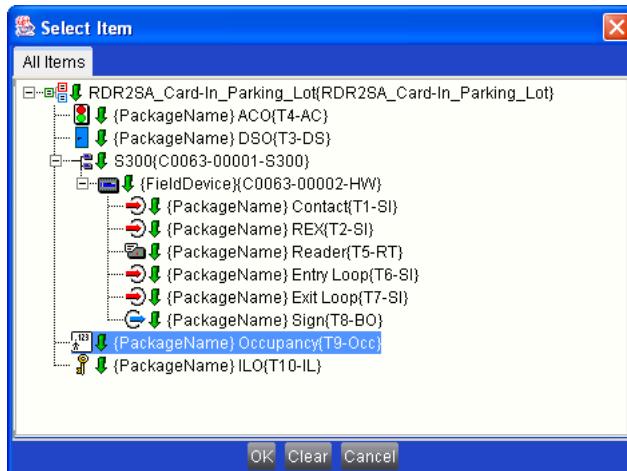
3. Select the **Interlock Definition** tab.
4. Verify the **Match All (AND)** option is selected in the **Logic** drop-down list.
5. Click **Add**.
6. Select **{PackageName} Entry Loop** and click **OK**.



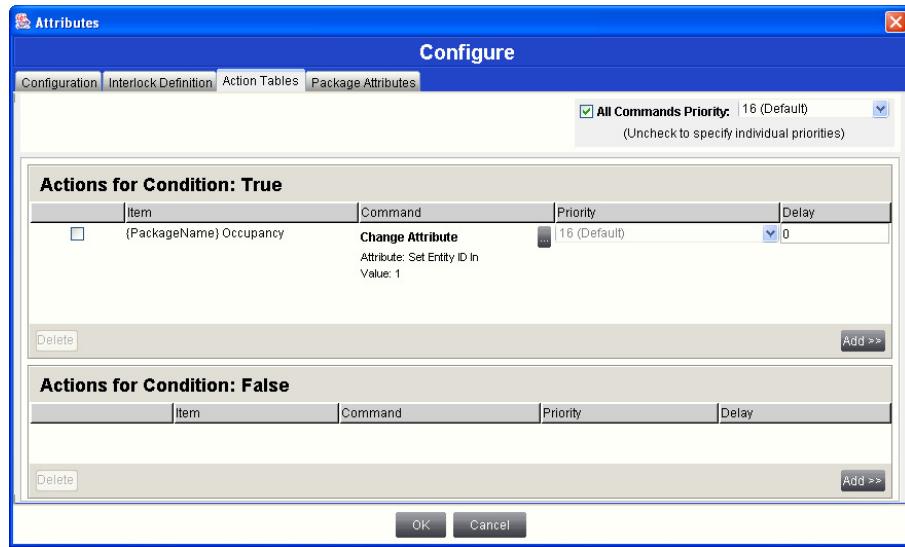
7. In the Logic table, select **Alarm** under the **Value** column.



8. Select the **Action Tables** tab.
9. In the Actions for Condition: True area, click **Add**.
10. Select **{PackageName} Occupancy** and click **OK**.



11. Under the **Command** column, click the **Browse** button [...].
12. Select **Set Entity ID In** in the **Attribute** drop-down list.
13. Enter a **Value** of 1 and click **OK**.



14. Click **OK**.

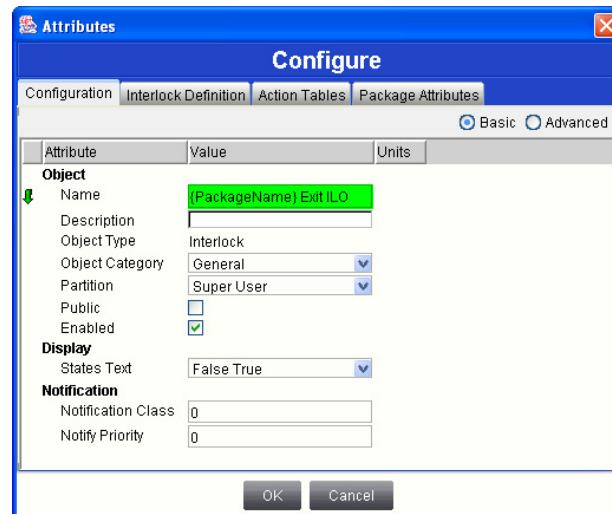
15. Click **Save**.

➤ **To add the {PackageName} Exit ILO object:**

1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.

For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} Exit ILO, in the **Name** field.



3. Select the **Interlock Definition** tab.
4. Verify the **Match All (AND)** option is selected in the **Logic** drop-down list.
5. Click **Add**.

6. Select **{PackageName}** Exit Loop and click **OK**.
7. In the Logic table, select **Alarm** under the **Value** column.
8. Select the **Action Tables** tab.
9. In the Actions for Condition: True area, click **Add**.
10. Select **{PackageName} Occupancy** and click **OK**.
11. Under the **Command** column, click the **Browse** button [...].
12. Select **Set Entity ID Out** in the **Attribute** drop-down list.
13. Enter a **Value** of **1** and click **Ok**.
14. Click **OK**.
15. Click **Save**.

Object Modifications

Modify the following object(s) in the template:

Table 7-13: Object Modifications (Occupancy: Parking Lot with “LOT FULL” Sign Application)

Object	Attribute	Modification
{PackageName} ACO	Occupancy In Object (Access tab)	Select the {PackageName} Occupancy object.
	Occupancy Check (Access tab)	Enforce, Only When Operational Violations will be reported to the host software and access will be denied, but only if the system is operational.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

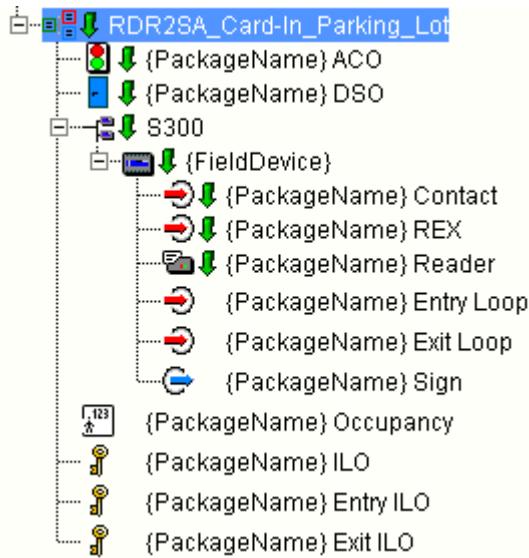


Figure 7-7: Object Hierarchy for Occupancy: Parking Lot with “LOT FULL” Sign Application

Application Notes

The inputs and outputs for the parking lot application have the following functions:

Reader 1 Spare – The Reader 1 Spare input point is wired to the entry vehicle loop. When a vehicle passes over the loop, Reader 1 Spare receives the input from the device, and by way of the {PackageName} Entry ILO object, causes the Occupancy object to increase its count by one.

Reader 2 Spare – The Reader 2 Spare input point is wired to the exit vehicle loop. When a vehicle passes over the loop, Reader 2 Spare receives the input from the device, and by way of the {PackageName} Exit ILO object, causes the Occupancy object to decrease its count by one.

Reader 2 Green LED – The Reader 2 Green LED output point is wired to the “LOT FULL” sign. When the maximum occupancy number reaches 500 or more entities, the Reader 2 Green LED output will activate the sign. When the number falls below 500, the output will deactivate the sign.

Occupancy: Counting People with Turnstiles

Application Description

When facility managers wish to count the actual number of people entering their facility, they commonly use turnstiles for high-traffic applications. Each time a person passes through a turnstile to enter or exit a facility, the P2000 SMS can track how many persons are currently in the facility and generate a notification if the

number of occupants equals or surpasses the maximum number allowed in the facility.

The following scenario helps describe how this application functions:

1. A facility has four entry turnstiles and four exit turnstiles, consisting of eight inputs.
2. These inputs are wired to the S300-SI8 input module.
3. The maximum number of occupants allowed in the facility at one time is 1000.
4. An Occupancy object counts the number of guests entering and exiting the facility.
5. When a visitor passes through any of the entry turnstiles, the number of occupants increases by one.
6. When a visitor passes through any of the exit turnstiles, the number of occupants decreases by one.
7. When the number of visitors in the facility equals or exceeds 1000, the P2000 SMS generates a notification.
8. The guard manned at the facility entrance sees the notification on his P2000 workstation and begins refusing new visitors until the number of occupants decreases to below 1000.

Using an Existing Template

The following instructions enable you to define an Occupancy: Counting People with Turnstiles application using an S300-SI8 supervised input module. This application requires eight general purpose inputs, one Interlock object, and one Occupancy object.

Start by importing and copying the JCI_SI8_Full-IO template (see page 6-9) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-14: Object Additions (Occupancy: Counting People with Turnstiles Application)

Object	Name ¹	Description
Occupancy	{PackageName} Occupancy	Tracks the number of visitors entering and exiting the facility. Maximum Occupancy: 1000
Interlock	{PackageName} Entry ILO	This object increases the occupancy count by one when a person passes through one of the four entry turnstiles. See page 7-38 for configuration instructions.
Interlock	{PackageName} Exit ILO	This object decreases the occupancy count by one when a person passes through one of the four exit turnstiles. See page 7-41 for configuration instructions.

1. The names in this column are suggestions. You may enter any name you wish.

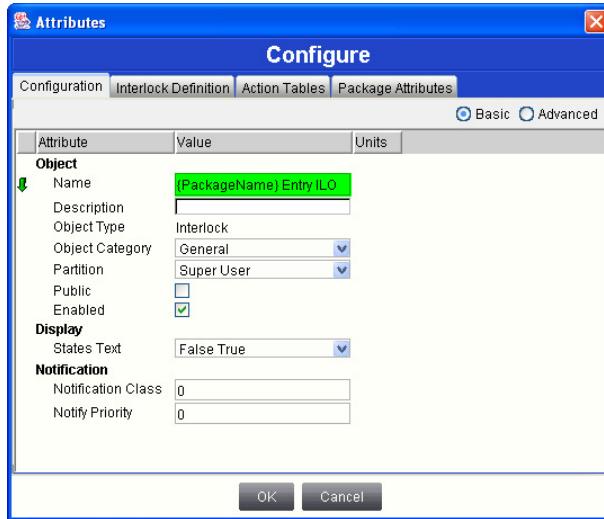
► **To add the {PackageName} Entry ILO object:**

NOTE

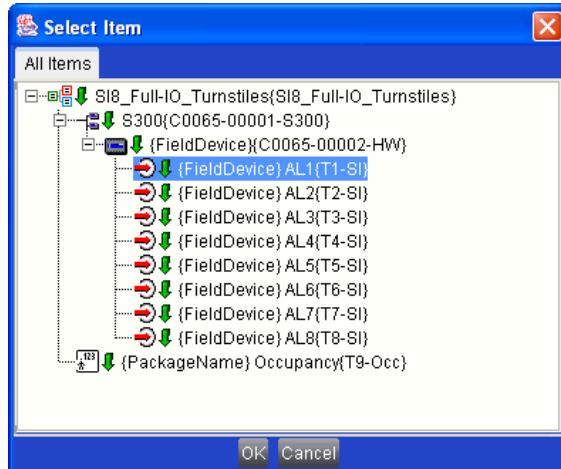
To complete the following steps, you must first add and save the Occupancy object listed in Table 7-14.

1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.
For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} Entry ILO, in the **Name** field.

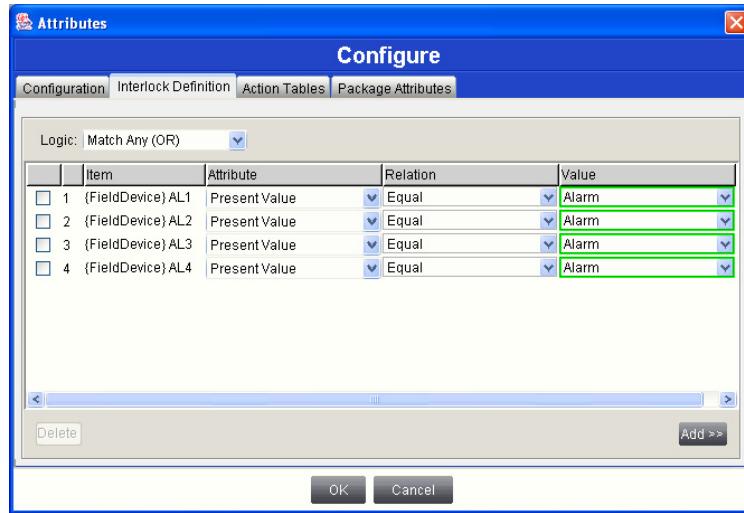


3. Select the **Interlock Definition** tab.
4. Select **Match All (OR)** in the **Logic** drop-down list.
5. Click **Add**.
6. Expand the object tree, select **{FieldDevice} AL1**, and click **OK**.

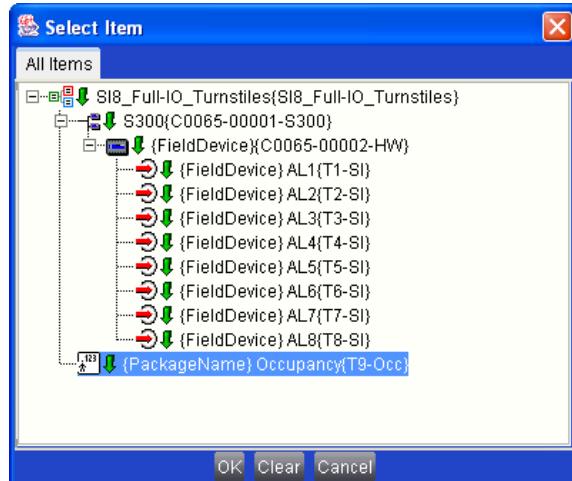


7. In the Logic table, select **Alarm** under the **Value** column.

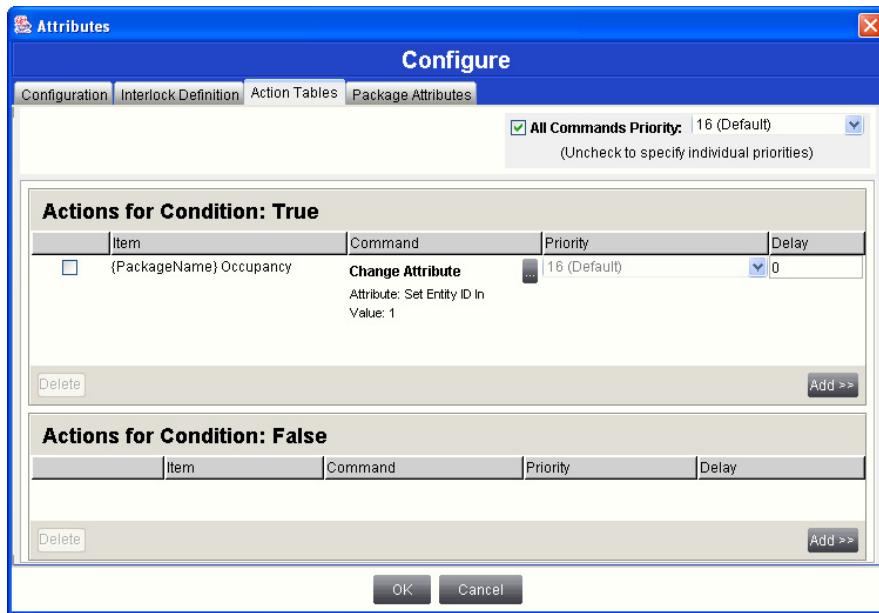
8. Repeat steps 5-7 for the {FieldDevice} AL2, {FieldDevice} AL3, and {FieldDevice} AL4 input points.



9. Select the **Action Tables** tab.
10. In the Actions for Condition: True area, click **Add**.
11. Select **{PackageName} Occupancy** and click **OK**.



12. Under the **Command** column, click the **Browse** button [...].
13. Select **Set Entity ID In** in the **Attribute** drop-down list.
14. Enter a **Value** of **1** and click **Ok**.



15. Click **OK**.

16. Click **Save**.

➤ **To add the {PackageName} Exit ILO object:**

1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.

For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

2. On the **Configuration** tab, enter a name for the Interlock object, such as {PackageName} Exit ILO, in the **Name** field.
3. Select the **Interlock Definition** tab.
4. Select **Match All (OR)** in the **Logic** drop-down list.
5. Click **Add**.
6. Expand the object tree, select {FieldDevice} AL5, and click **OK**.
7. In the Logic table, select **Alarm** under the **Value** column.
8. Repeat steps 5-7 for the {FieldDevice} AL6, {FieldDevice} AL7, and {FieldDevice} AL8 input points.
9. Select the **Action Tables** tab.
10. In the Actions for Condition: True area, click **Add**.
11. Select **{PackageName} Occupancy** and click **OK**.
12. Under the **Command** column, click the **Browse** button [...].
13. Select **Set Entity ID Out** in the **Attribute** drop-down list.
14. Enter a **Value** of **1** and click **Ok**.
15. Click **OK**.
16. Click **Save**.

Object Modifications

No objects need to be modified in the template.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

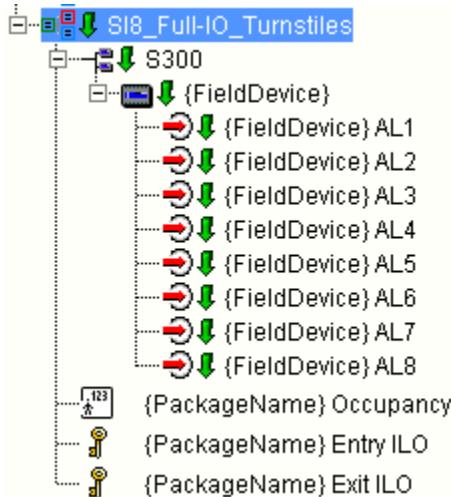


Figure 7-8: Object Hierarchy for Occupancy: Counting People with Turnstiles Application

Application Notes

The inputs for the turnstile application have the following functions:

AL1 through AL4 – These input points are wired to the entry turnstiles. When a person passes through one of these turnstiles, the system receives the input from the device, and by way of the {PackageName} Entry ILO object, causes the Occupancy object to increase its count by one.

AL5 through AL8 – These input points are wired to the exit turnstiles. When a person passes through one of these turnstiles, the system receives the input from the device, and by way of the {PackageName} Exit ILO object, causes the Occupancy object to decrease its count by one.

Asset Protection

Application Description

The asset protection application enables you to deter the theft of assets from a facility by assigning assets to owners (entities), and tracking whether an owner is present if the asset is detected near a portal. Asset protection applications consist of the following:

- A Radio Frequency Identification (RFID) identifier tag attached/adhered to the asset. This tag is similar to an identifier badge, but has a greater detection range. See “Radio Frequency Identification (RFID) Tags” on page 4-11 for more information.
- A long-range RFID reader that can detect an asset’s RFID tag.
- Security system that can generate an alarm and/or prevent egress by placing the door(s) in lockdown mode when it determines that an asset is unaccompanied by the proper entity.

In the P2000 SMS, an asset can be accompanied by a one or more entities, or by an entity group. Defining asset entities and assigning owners is accomplished with the P2000 host software. However, the asset protection application must first be properly configured with the P2000 SCT.

The following scenario helps describe how the asset protection application functions:

1. John has been assigned a notebook computer (Asset). Both John and the notebook computer are considered entities in the P2000 SMS.
2. In the P2000 host software, John is assigned as the only owner of the notebook computer. He is the only person allowed to remove the notebook from the building.
3. The entry/exit door separating the suite where John works and the building’s front lobby has an entry reader and an exit reader. Occupants must badge to enter and exit the suite.
4. The entry/exit door also includes a long-range RFID reader installed on the exit side of the door. This reader will detect John’s notebook if it comes within 10 feet of the reader.
5. When John attempts to leave with his notebook, the long-range RFID reader detects the notebook. According to the *Asset Timeout* attribute of the Access Control object, John has 5 seconds (default) to present his identifier to the proximity exit reader before the system goes into alarm.
6. The next day, Jane attempts to leave with John’s notebook, and she is not assigned as the notebook’s owner.
7. When she nears the long-range RFID reader, it detects the notebook and starts the *Asset Timeout* countdown.
8. She presents her identifier to the exit reader.

9. The P2000 SMS determines that she is not the owner of the notebook and immediately locks the door, so that it cannot be opened from the entry and exit side, and generates an alarm notification.

NOTE

Entities with Executive privileges are exempt from the asset-owner rule. In other words, if Jane has Executive privileges, she would be able to leave with the notebook, and the system would not generate an alarm.

Using an Existing Template

The following instructions enable you to define an Asset Protection application using two RDR2S-A modules. Apart from the standard entry/exit door object requirements, this application requires one additional Access Control object, one additional RDR2S-A module, and one additional S300 Reader Terminal object for the long-range RFID reader.

Start by importing and copying the JCI_RDR2SA_Card-In-Card-Out template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-15: Object Additions (Asset Protection Application)

Object	Name ¹	Description
S300 Hardware Module	{FieldDevice} Asset	Object representing the RDR2S-A hardware module. This additional RDR2S-A module will be wired to the long-range RFID reader.
S300 Reader Terminal	{PackageName} LR_Rdr	Object representing the long-range RFID reader. Parent Object: {FieldDevice} Asset

Table 7-15: Object Additions (Asset Protection Application)

Object	Name ¹	Description
Access Control	{PackageName} Asset ACO	This object is responsible for tracking assets and determining whether the other Access Control objects will grant access based on whether an asset is accompanied or unaccompanied. Primary Reader Object (Configuration tab): {PackageName} LR_Rdr Asset Processing (Processing tab): Asset Tracking Only

1. The names in this column are suggestions. You may enter any name you wish.

Object Modifications

Modify the following object(s) in the template:

Table 7-16: Object Modifications (Asset Protection Application)

Object	Attribute	Modification
{PackageName} Ingress ACO	Tracking Object (Processing tab)	Select the {PackageName} Asset ACO object. See Table 7-15.
	Asset Processing (Processing tab)	Select Deny All on Unaccompanied Asset .
{PackageName} Egress ACO	Tracking Object (Processing tab)	Select the {PackageName} Asset ACO object. See Table 7-15.
	Asset Processing (Processing tab)	Select Deny All on Unaccompanied Asset .

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

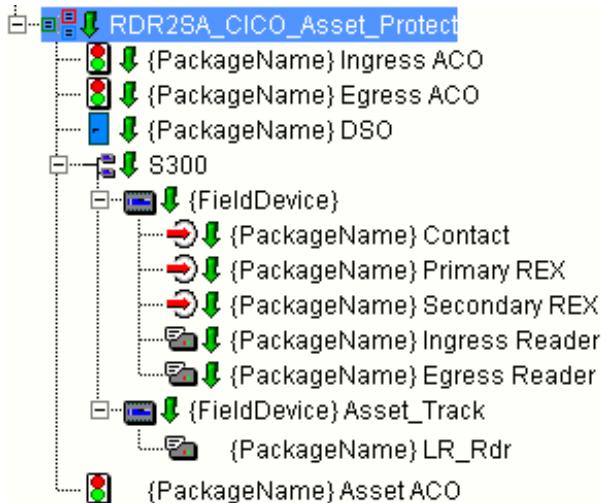


Figure 7-9: Object Hierarchy for Asset Protection Application

Application Notes

Since the RDR2S-A module can only support up to two readers, and because this application example requires three readers, you must add another RDR2S-A module to accommodate the third reader. When using a long-range RFID reader in this type of application, the door contact and REX device are not used, and therefore these input objects do not have to be inserted into the P2000 SCT. However, the RDR2S-A inputs for Reader 2 are available for use on a fourth reader or for general purpose inputs.

Assisted Access

NOTE

*This type of application is **not** supported on RDR2 hardware modules with firmware version PS-201D or earlier.*

Application Description

In this application, the door can open for an extended time based on the characteristics of the presented identifier. It can also operate an ADA Relay upon presentation of an identifier. The *Alternate Access Time* attribute determines the duration of the access time. The assisted shunt time will automatically be adjusted to exceed the configured *Shunt Time* by the same amount as the *Alternate Access Time* exceeds the *Access Time*.

Alternate Access can be always enabled, never enabled, or enabled by mask.

This feature satisfies the ADA (Americans with Disabilities Act) requirements for assisted access.

ADA Relay

An external ADA Relay can be controlled by an output through the use of the reader module. The activation can be delayed without the use of external additional hardware (unlike when using S300-SIO8 modules). The delay is necessary to avoid operating a door-opening device before the door is fully unlocked.

The ADA relay can be wired to the green light or the shunt connector of the supported S300 hardware module. On the S300 Reader Terminal object, the *ADA Relay Connector* attribute should then be set to “Green” or “Shunt,” respectively. The “None” option disables the ADA relay.

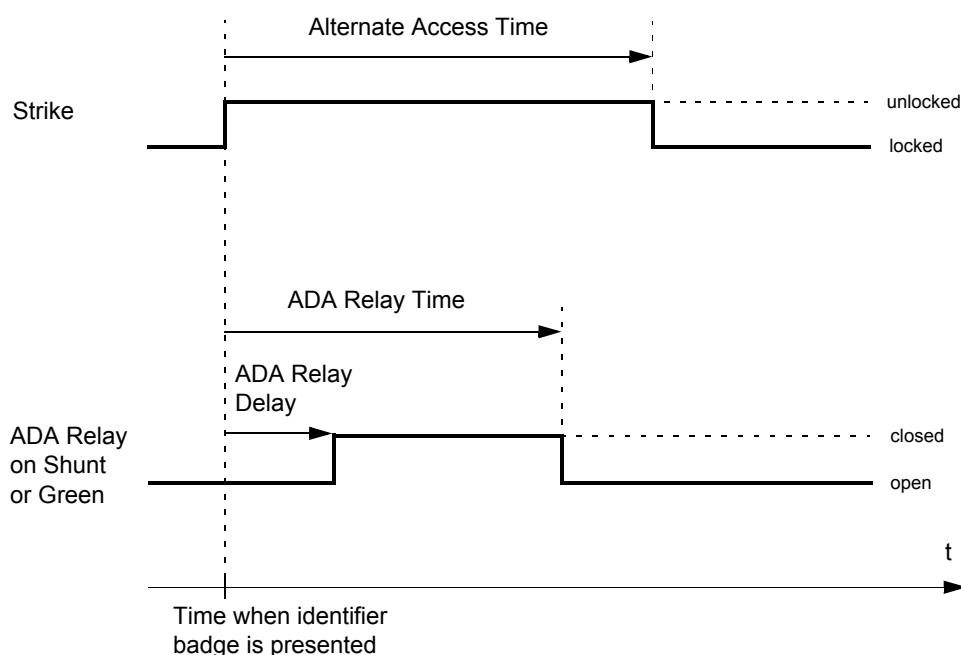


Figure 7-10: Alternate Access Timing Diagram

The following scenario helps describe how the Assisted Access application functions:

1. A building's front entrance can only be unlocked by presenting a valid identifier to the portal's reader.
2. The door also includes an automatic opening mechanism for entities needing assistance (e.g. entities using wheelchairs, elderly entities).
3. All entities needing assistance have **Special Access A** privileges assigned to their profile.

4. When an entity needing assistance presents his identifier to the door's reader, the following occurs:
 - The P2000 SMS recognizes the entity as having Special Access A privileges.
 - The door immediately unlocks and remains unlocked for 15 seconds (according to the *Alternate Access Time* attribute setting).
 - The *ADA Relay Delay* time (500 ms) keeps the door opening mechanism from activating to prevent the mechanism from opening the door before it is unlocked.
 - For 2 seconds (*ADA Relay Time* attribute setting), the P2000 SMS triggers the relay output for the door opening mechanism.
 - The door opening mechanism opens the door for the entity needing assistance.

Using an Existing Template

The following instructions enable you to define an Assisted Access application using an RDR2S-A hardware module.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

No objects need to be added to the template.

Object Modifications

Modify the following object(s) in the template:

Table 7-17: Object Modifications (Assisted Access Application)

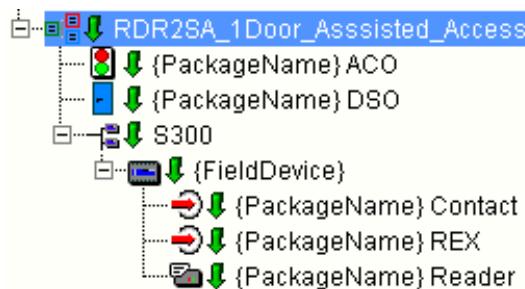
Object	Attribute	Modification
{PackageName} ACO	Alternate Access Mask (Privileges tab)	Select the Special Access A check box.

Table 7-17: Object Modifications (Assisted Access Application)

Object	Attribute	Modification
{PackageName} Reader	ADA Relay Connected	To Green Light Output
	ADA Relay Time	2 Seconds
	ADA Relay Delay	500 ms

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy*Figure 7-11: Object Hierarchy for the Assisted Access Application***Elevator Low Level Interface for Floor-by-Floor Control*****Application Description***

This application utilizes the JCI_RDR2SA_Card-In template (see page 6-33) to develop a low level elevator interface with the P2000 Security Management System. In this application, the P2000 SMS interfaces with the elevator system through a series of Security Supervised Input objects and Security Binary Output objects to provide floor-by-floor control throughout the building.

The following scenario helps describe how this application functions:

1. The elevator system of a residential/commercial building stops at 12 different floors. See Table 7-18.
2. The building has a single elevator, and a single reader is installed inside the elevator cab.
3. Each time a floor button is pressed inside the elevator cab, the P2000 SMS receives this information as an input for tracking purposes. Therefore, each floor button has a dedicated Security Supervised Input object in the P2000 SCT.

4. Anyone can access floors designated as public access floors during the *active* day/time of a time zone. During *inactive* days/times of a time zone, an entity must present a valid access badge identifier to the elevator cab's reader to access a floor. In addition, the entity must be assigned the proper permissions to access a particular floor. Refer to the *P2000AE Software User Manual* for information on assigning access permissions to an entity record.
5. One output is assigned to every floor button in the elevator cab. The P2000 SMS grants access to a floor by enabling the corresponding car-call button when an entity presents an access badge identifier at the reader installed in the elevator cab. Therefore, each floor button has a dedicated Security Binary Output object in the P2000 SCT.
6. Anyone can call the elevator from the lobby and access the Lobby floor from within the elevator. There are no restrictions to this floor. See Table 7-18 and Table 7-19.
7. Floors 2 (Mezzanine) through 10 are commercial businesses. Anyone can access these floors during normal business hours, Monday through Friday. During non-public access hours, entities with valid access badge identifiers can access these floors according to their access rights.
8. Floor 11 (Restaurant) is also a commercial business. Anyone can access this floor during the restaurant's designated open hours (11:00 AM to 10:00 PM) any day of the week. During non-public access hours, entities with valid access badge identifiers can access this floor according to their access rights.
9. Floor 12 (Penthouse) is the only residential floor of the building. There is no public access to this floor. Residents of the building are the only entities allowed to access the floor.
10. Each time zone requires a Schedule object in the P2000 SCT. This application uses four different time zones defined in the P2000 host software, so four Schedule objects will be defined in the P2000 SCT.

Floor Description

Floor numbers, floor names, and time zones are defined in the P2000 host software. Refer to the *P2000AE Software User Manual* for instructions.

Table 7-18: Floors of Commercial/Residential Building

Floor Number	Floor Name	Public Access?	Time Zone	Commercial/Residential
1	Lobby	Yes	Always	Commercial
2	Mezzanine	Yes	Business Hours	Commercial
3	Level 3	Yes	Business Hours	Commercial
4	Level 4	Yes	Business Hours	Commercial
5	Level 5	Yes	Business Hours	Commercial
6	Level 6	Yes	Business Hours	Commercial

Table 7-18: Floors of Commercial/Residential Building

Floor Number	Floor Name	Public Access?	Time Zone	Commercial/Residential
7	Level 7	Yes	Business Hours	Commercial
8	Level 8	Yes	Business Hours	Commercial
9	Level 9	Yes	Business Hours	Commercial
10	Level 10	Yes	Business Hours	Commercial
11	Restaurant	Yes	Restaurant Hours	Commercial
12	Penthouse	No	Never	Residential

Time Zones

Add the time zones listed in Table 7-19 using the P2000 host software. Refer to the *P2000AE Software User Manual* for instructions.

Table 7-19: Time Zones

Time Zone	Active Days	Active Times
Always	All	All
Business Hours	Monday through Friday	8:00 AM to 6:00 PM
Restaurant Hours	All	11:00 AM to 10:00 PM
Never	None	None

Using an Existing Template

The following instructions enable you to define an Elevator Low Level Interface for Floor-by-Floor Control application using an RDR2S-A hardware module and an S300-SIO8 hardware module.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-20: Object Additions (Elevator Low Level Interface for Floor-by-Floor Control Application)

Object	Name ¹	Description
S300 Hardware Module	{FieldDevice} Elevator I/O	<p>Object representing the S300-SIO8 hardware module.</p> <p>Destination: S300 Trunk object Hardware Module Type: SIO8</p>
Security Supervised Input	{FieldDevice} IN1 {FieldDevice} IN2 {FieldDevice} IN3 {FieldDevice} IN4 {FieldDevice} IN5 {FieldDevice} IN6 {FieldDevice} IN7 {FieldDevice} IN8	<p>Add eight Security Supervised Input objects. These objects represent inputs AL1 through AL8 on the SIO8 module.</p> <p>Parent Hardware: {FieldDevice} Elevator I/O</p> <p>Connector: Select connector AL1 for {FieldDevice} IN1, AL2 for {FieldDevice} IN2, and so on for all eight inputs.</p>
Security Supervised Input	{FieldDevice} IN9 {FieldDevice} IN10 {FieldDevice} IN11 {FieldDevice} IN12	<p>Add four additional Security Supervised Input objects for the elevator cab button inputs for floors 9 through 12.</p> <p>Parent Hardware (All Inputs): {FieldDevice}</p> <p>{FieldDevice} IN9 Connector: Reader 1 Spare</p> <p>{FieldDevice} IN10 Connector: Reader 2 Spare</p> <p>{FieldDevice} IN11 Connector: Reader 2 Door Contact</p> <p>{FieldDevice} IN12 Connector: Reader 2 REX</p>
Security Binary Output	{FieldDevice} OUT1 {FieldDevice} OUT2 {FieldDevice} OUT3 {FieldDevice} OUT4 {FieldDevice} OUT5 {FieldDevice} OUT6 {FieldDevice} OUT7 {FieldDevice} OUT8	<p>Add eight Security Binary Output objects. These objects represent OUTPUT 1 NO/NC through OUTPUT 8 NO/NC on the SIO8 module.</p> <p>Parent Hardware: {FieldDevice} Elevator I/O</p> <p>Connector: Select connector OUTPUT 1 NO/NC for {FieldDevice} OUT1, OUTPUT 2 NO/NC for {FieldDevice} OUT2, and so on for all eight outputs.</p>

Table 7-20: Object Additions (Elevator Low Level Interface for Floor-by-Floor Control Application)

Object	Name ¹	Description
Security Binary Output	{FieldDevice} OUT9 {FieldDevice} OUT10 {FieldDevice} OUT11 {FieldDevice} OUT12	Add four additional Security Binary Output objects for the elevator cab button outputs for floors 9 through 12. Parent Hardware (All Outputs): {FieldDevice} {FieldDevice} OUT9 Connector: Reader 2 RED LED {FieldDevice} OUT10 Connector: Reader 2 Green LED {FieldDevice} OUT11 Connector: Reader 2 Shunt {FieldDevice} OUT12 Connector: Reader 2 Strike
Schedule	{Schedule} Always	Object representing the Always time zone. See Table 7-19 on page 7-51. Time Zone: Always
Schedule	{Schedule} 8AM to 6PM Mon-Fri	Object representing the Business Hours time zone. See Table 7-19 on page 7-51. Time Zone: Business Hours
Schedule	{Schedule} 11AM-10PM All Week	Object representing the Restaurant Hours time zone. See Table 7-19 on page 7-51. Time Zone: Restaurant Hours
Schedule	{Schedule} Never	Object representing the Never time zone. See Table 7-19 on page 7-51. Time Zone: Never
Elevator	{PackageName} Elevator	Object used to interface at a low level with the elevator system. Access Control Object: {PackageName} ACO Door Sequence Object: {PackageName} DSO Floor List: See “Floor List Attribute Configuration” on page 54.

1. The names in this column are suggestions. You may enter any name you wish.

NOTE

64 Schedule objects are automatically created when creating a CK722 Device object in the P2000 SCT database; however, this does not prevent you from inserting this type of object into a template and loading it as part of a package. As an alternative, you may use the existing Schedule objects instead of adding ones to this template.

Floor List Attribute Configuration

Configure the Elevator object's *Floor List* attribute according to Table 7-21.

Table 7-21: Floor List Attribute Configuration

Floor No.	Public Access Attribute	Output Attribute	Input Attribute
1	Schedule (Always)	{FieldDevice} OUT1	{FieldDevice} IN1
2	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT2	{FieldDevice} IN2
3	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT3	{FieldDevice} IN3
4	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT4	{FieldDevice} IN4
5	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT5	{FieldDevice} IN5
6	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT6	{FieldDevice} IN6
7	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT7	{FieldDevice} IN7
8	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT8	{FieldDevice} IN8
9	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT9	{FieldDevice} IN9
10	Schedule (8AM to 6PM Mon-Fri)	{FieldDevice} OUT10	{FieldDevice} IN10
11	Schedule (11AM-10PM All Week)	{FieldDevice} OUT11	{FieldDevice} IN11
12	Schedule (Never)	{FieldDevice} OUT12	{FieldDevice} IN12

Object Modifications

Modify the following object(s) in the template:

Table 7-22: Object Modifications (Elevator Low Level Interface for Floor-by-Floor Control Application)

Object	Attribute	Modification
{FieldDevice} OUT1 {FieldDevice} OUT2 {FieldDevice} OUT3 {FieldDevice} OUT4 {FieldDevice} OUT5 {FieldDevice} OUT6 {FieldDevice} OUT7 {FieldDevice} OUT8 {FieldDevice} OUT01 {FieldDevice} OUT02 {FieldDevice} OUT21 {FieldDevice} OUT22	Duration	1.0 seconds See "Application Notes" on page 7-56 for an explanation.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

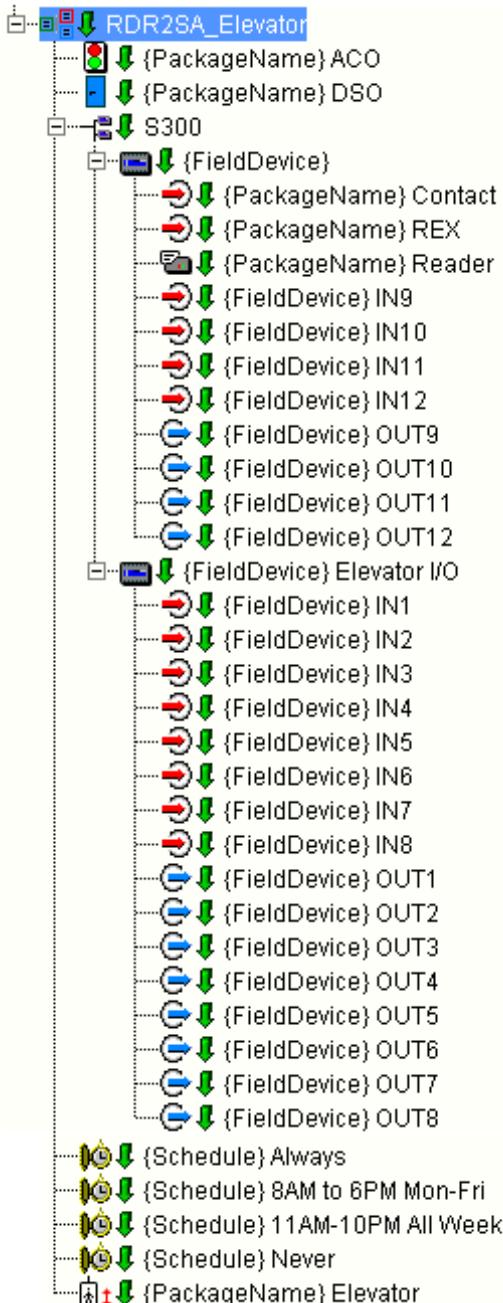


Figure 7-12: Object Hierarchy for Elevator Low Level Interface for Floor-by-Floor Control Application

Application Notes

Activate Only Selected Floor – When the Elevator object's *Low Level Mode* attribute is set to Activate Only Selected Floor, the *Duration* attribute for all Security Binary Output objects assigned to the *Floor List* attribute must be set to 1.0 seconds. In an Activate Only Selected Floor application, when an entity presents his/her identifier to the elevator cab's reader, the buttons inside the cab *will not* become lit to indicate which floor(s) the entity can access. When the entity presses a floor button, the signal is sent to the access controller, which then determines whether to grant or deny the entity access to the selected floor. If the entity is granted access, the S300-SIO8 module signals the elevator controller that this button has now been pressed, and the elevator will take the entity to the selected floor. The additional second in the duration is needed to signal the elevator controller that the floor button has been pressed.

This method is preferred, as only a single output needs to be activated for each allowed floor button that is pressed. However, the entity receives no indication in the cab as to which floors he/she can access.

Activate All Allowed Floors – If the *Low Level Mode* attribute is set to Activate All Allowed Floors, the output duration must remain at the default setting: 0 seconds. In an Activate All Allowed Floors application, when an entity presents his/her identifier to the elevator cab's reader, the buttons inside the cab *will* become lit to indicate which floor(s) the entity can access. Pressing an active, lit floor button prompts the elevator controller to immediately take the elevator to the selected floor. Pressing an inactive, unlit floor button does nothing, as the elevator controller does not receive a signal that this button has been pressed.

This method provides an indication to the entity in the cab of which floors are allowed, but potentially requires the activation and deactivation of all outputs for that elevator for each access request.

INTRUSION DETECTION EXAMPLES

This section describes various intrusion detection application examples, most of which are common applications used in the field. Included with these examples are instructions on how to configure the applications using the P2000 SCT.

The examples described in this section also require host software configuration, which is not described in this manual. For information on using the host software, refer to the *P2000AE Software User Manual*.

One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application

Application Description

This application utilizes the JCI_RDR2SA_Card-In template (see page 6-33) to demonstrate how access control and intrusion detection applications can be integrated. In this application, an entity with intrusion rights (defined in the P2000 host software) is authorized to arm and disarm the P2000 intrusion detection system.

The following scenario helps describe how this application functions:

1. The entity Jane Doe is authorized to arm and disarm her company's intrusion detection system.
2. The intrusion detection system consists of 16 input sensors, eight zones, one area, one keypad/display module, and one output annunciation device (sounder).
3. The keypad/display module is located in the lobby.
4. Jane Doe arrives in the morning and must disarm the system.
5. She first presents her identifier at the lobby door's reader.
6. The Entry input device (a motion detector over the door) detects Jane's movement as she passes through the door.
7. The Intrusion Zone object associated with the sensor that has just been tripped has an *Entry Time* of 45 seconds. Jane has 45 seconds to disarm the area using the keypad/display module before the system goes into alarm.
8. She disarms the area.
9. At the end of the day, she must arm the area. She does this using the keypad/display module.
10. When she arms the area, she has 45 seconds to exit the facility (*Exit Time*) before the entire area is armed.

Using an Existing Template

The following instructions enable you to define a One Door, Eight Zone, One Area Intrusion with Entry/Exit Time application using an RDR2S-A reader module and an S300-I16 input module.

Start by importing and copying the JCI_RDR2SA_Card-In template (see page 6-33) and follow the instructions in this section to modify the template. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-23: Object Additions (One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application)

Object	Name ¹	Description
S300 Hardware Module	{FieldDevice} Intrusion	Object representing the S300-I16 module. Destination: S300 Trunk object Hardware Module Type: I16
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8 {FieldDevice} AL9 {FieldDevice} AL10 {FieldDevice} AL11 {FieldDevice} AL12 {FieldDevice} AL13 {FieldDevice} AL14 {FieldDevice} AL15 {FieldDevice} AL16	Add sixteen Security Supervised Input objects. These objects represent inputs AL1 through AL16 on the I16 module. Parent Hardware: {FieldDevice} Intrusion Connector: The connector selected should correspond to the name assigned to the object. For example, for {FieldDevice} AL1, select AL1 for the <i>Connector</i> attribute. See also the JCI_I16_Full-IO template on page 6-5.
Security Binary Output	{FieldDevice} OUT1 {FieldDevice} OUT2	Objects representing the annunciators that will sound when the system is in alarm. Parent Hardware: {FieldDevice} Connectors: Reader 2 Shunt for {FieldDevice} OUT1; Reader 2 Green LED for {FieldDevice} OUT2
Intrusion Zone	{Zone01} Zone	Monitors and controls a group of sensors used in the intrusion detection system. Entry Time: 45 seconds Exit Time: 45 seconds Entry Input Attribute List: {FieldDevice} AL1 – Assuming that the AL1 input (S300-I16 module) is the entry sensor for the door closest to the keypad/display module. Alarm Input Attribute List: {FieldDevice} AL2 – Assuming that the AL2 input (S300-I16 module) is a sensor near the keypad/display module.
Intrusion Zone	{Zone02} Zone	Alarm Input Attribute List: {FieldDevice} AL3 {FieldDevice} AL4

Table 7-23: Object Additions (One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application)

Object	Name ¹	Description
Intrusion Zone	{Zone03} Zone	Alarm Input Attribute List: {FieldDevice} AL5 {FieldDevice} AL6
Intrusion Zone	{Zone04} Zone	Alarm Input Attribute List: {FieldDevice} AL7 {FieldDevice} AL8
Intrusion Zone	{Zone05} Zone	Alarm Input Attribute List: {FieldDevice} AL9 {FieldDevice} AL10
Intrusion Zone	{Zone06} Zone	Alarm Input Attribute List: {FieldDevice} AL11 {FieldDevice} AL12
Intrusion Zone	{Zone07} Zone	Alarm Input Attribute List: {FieldDevice} AL13 {FieldDevice} AL14
Intrusion Zone	{Zone08} Zone	Alarm Input Attribute List: {FieldDevice} AL15 {FieldDevice} AL16
Intrusion Area	{Area} Intrusion Area	Arms and disarms its associated Intrusion Zone objects. Zone Object List: {Zone01} Zone {Zone02} Zone {Zone03} Zone {Zone04} Zone {Zone05} Zone {Zone06} Zone {Zone07} Zone {Zone08} Zone
S300 Hardware Module	{FieldDevice} Keypad	Object representing the Keypad/Display Module (KDM). Destination: S300 Trunk object Hardware Module Type: KDM
Intrusion Keypad/Display	{Area} Keypad Display	Interfaces with the Keypad/Display module and allows authorized users to control Intrusion Area, Intrusion Zone, and Intrusion Annunciator objects. Parent Hardware: {FieldDevice} Keypad

Table 7-23: Object Additions (One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application)

Object	Name ¹	Description
Intrusion Annunciator	{Area} Annunciator	<p>Resets associated output points. It may be silenced by authorized users via the Keypad/Display module or by users via the host (P2000 server).</p> <p>Annunciator Output Attribute List: {FieldDevice} OUT1 {FieldDevice} OUT2</p> <p>Assuming that these devices represent the annunciators that will sound when the system is in alarm.</p>

1. The names in this column are suggestions. You may enter any name you wish.

Object Modifications

No objects need to be modified in the template.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

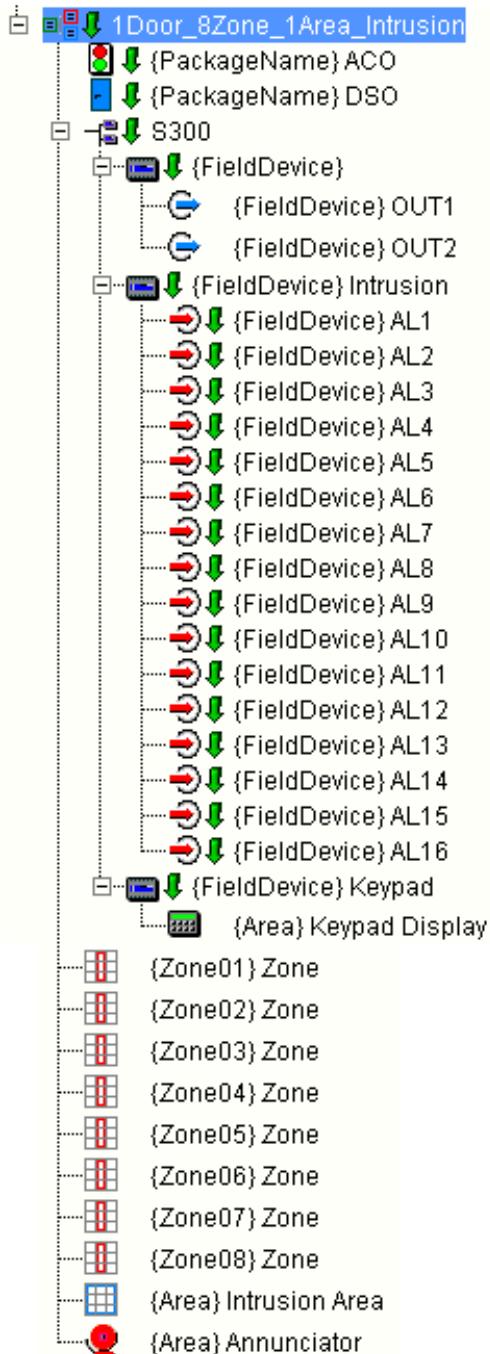


Figure 7-13: Object Hierarchy for One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application

Application Notes

The following describes the reasoning behind the actions taken to develop this application.

Entry Time – The *Entry Time* attribute for the **{Zone01} Zone** object is set to 45 seconds. If a sensor in Zone 1 is tripped while the system is armed, the system will go into alarm unless it is disarmed before the 45 second time period elapses. The Entry Time for Zones 2-8 is set to 0 seconds. If a sensor in Zones 2-8 is tripped, the intrusion detection system will immediately go into alarm (no delay).

NOTE

This application example does not incorporate the Delay Announcer Output Attribute of the Intrusion Zone object. This attribute enables you to select a Security Binary Output object that represents an annunciation device that is activated for the duration of the Entry Time and Exit Time. For example, if the Entry Time is 45 seconds, a buzzer sounds to warn the person entering the area that an alarm will be activated unless it is disarmed before the Entry Time expires.

{Area} Announcer – In this application example, there are two annunciation devices, both of which are controlled by the **{Area}** Announcer object. Both devices are wired to the RDR2S-A module (Reader 2 Shunt and Reader 2 Green LED).

Intrusion Zone objects – Each zone represents two alarm inputs of the S300-I16 module. However, this does not indicate the actual number of sensors that are wired to these unsupervised inputs, as they may be wired in series when run to the S300-I16 module's alarm inputs. For example, AL2 may be wired to a series of 10 sensors. AL2 is an alarm input for Zone02. If any one of the 10 sensors is tripped, the P2000 SMS senses the change in status through the AL2 input and goes into alarm and indicates that a Zone02 sensor has been tripped.

Supervised Alarm, Tamper and Trouble Inputs Application

NOTE

This application uses the Eight_Zone_Area_Keypad_Annun legacy template. In order to create this application, you will need to import this template. For more information on legacy templates and how to import them, see "Legacy Templates" on page 63.

Application Description

This application utilizes the Eight_Zone_Area_Keypad_Annun legacy template to demonstrate the design of an intrusion system that consists of the following inputs:

- **Alarm** – An alarm input is triggered when the sensor detects an intrusion (e.g. when a motion detector detects motion).

- **Tamper** – A tamper input is triggered when the sensor detects that someone is attempting to tamper with the sensor. Tamper switches detect when a detector's enclosure is opened.
- **Trouble** – A trouble input is triggered when the sensor detects a trouble condition (e.g. an obscured lens on the detector).

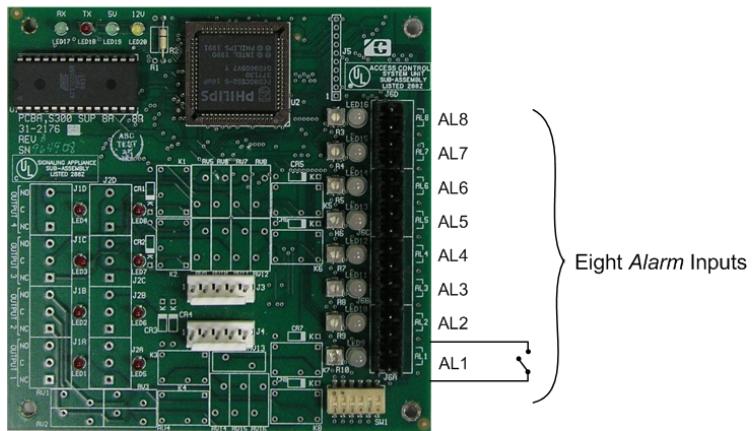
Each alarm, tamper, and trouble input should be wired to separate inputs, so a sensor with all three inputs must be wired to three different terminals on the input module, such as the S300-SI8. See Figure 7-14.

The following application example consists of the following:

- One area
- One keypad/display module
- Eight zones
- Eight supervised input sensors
- Tamper and trouble inputs for each sensor

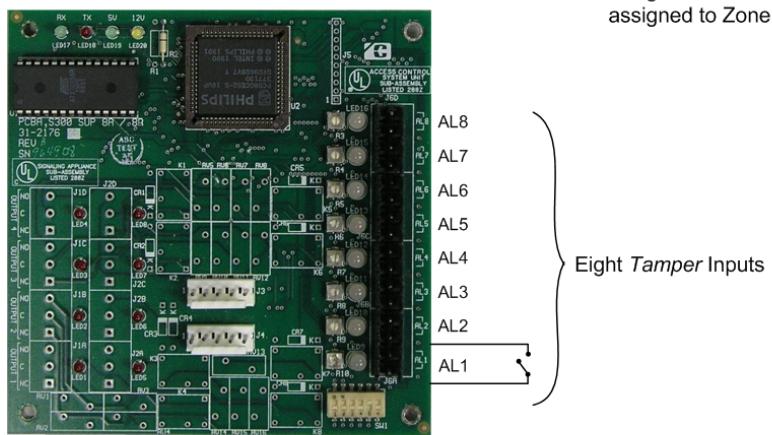
To have eight supervised input sensors with tamper and trouble inputs, the application requires a total of 24 inputs. Three S300-SI8 modules can provide enough inputs to cover this number (each module provides eight supervised inputs). See Figure 7-14.

S300-SI8 Module #1



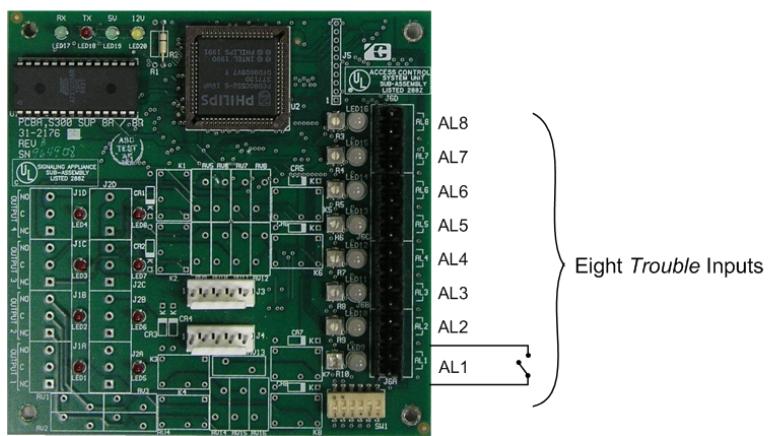
Eight Alarm Inputs

S300-SI8 Module #2



Eight Tamper Inputs

S300-SI8 Module #3



Eight Trouble Inputs

There are a total of twenty-four (24) inputs divided into eight (8) zones. All three AL1 inputs are assigned to Zone 1. All three AL2 inputs are assigned to Zone 2, and so on.

Figure 7-14: Supervised Alarm, Tamper and Trouble Inputs Application

Using an Existing Template

The following instructions enable you to define a Supervised Alarm, Tamper and Trouble Inputs application using three S300-SI8 input modules.

Start by importing and copying the Eight_Zone_Area_Keypad_Annun legacy template and follow the instructions in this section to modify it. See page 6-63 for information on how to import legacy templates. See also page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

For the purposes of this application, this section directs you to add three S300 Hardware Module objects representing three S300-SI8 modules and eight Security Supervised Input objects for each of the SI8 Hardware Module objects, so that the template has a total of 24 Security Supervised Input objects.

NOTE

Another method not covered in these instructions would be to use a single JCI_SI8_Full-IO template, load three copies of the template as a package into a CK722 Device object, and then modify the Intrusion Zone objects after they have been loaded as a package into the same CK722 Device object.

Add the following object(s) to the template:

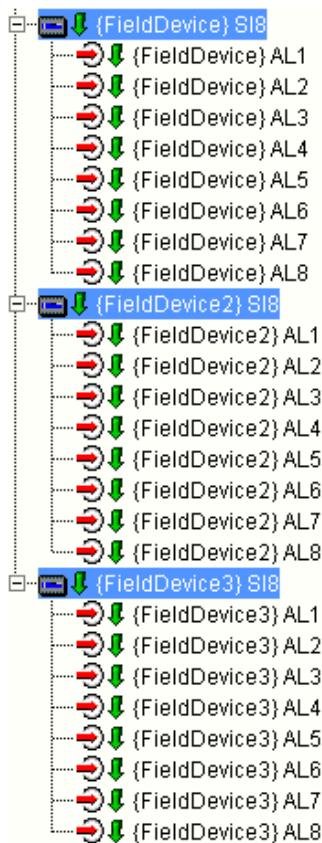
Table 7-24: Object Additions (Supervised Alarm, Tamper and Trouble Inputs Application)

Object	Name ¹	Description
S300 Hardware Module	{FieldDevice} SI8 {FieldDevice2} SI8 {FieldDevice3} SI8	Add three S300 Hardware Module objects. Each one represents the S300-SI8 module. Destination: S300-1 Trunk object Hardware Module Type: SI8

Table 7-24: Object Additions (Supervised Alarm, Tamper and Trouble Inputs Application)

Object	Name ¹	Description
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8	Add eight Security Supervised Input objects. These objects represent inputs AL1 through AL8 on the SI8 module. Parent Hardware: {FieldDevice} SI8 Connector: The connector selected should correspond to the name assigned to the object. For example, for {FieldDevice} AL1, select AL1 for the <i>Connector</i> attribute. See also the JCI_SI8_Full-IO template on page 6-9.
Security Supervised Input	{FieldDevice2} AL1 {FieldDevice2} AL2 {FieldDevice2} AL3 {FieldDevice2} AL4 {FieldDevice2} AL5 {FieldDevice2} AL6 {FieldDevice2} AL7 {FieldDevice2} AL8	Add eight Security Supervised Input objects. These objects represent inputs AL1 through AL8 on the SI8 module. Parent Hardware: {FieldDevice2} SI8 Connector: The connector selected should correspond to the name assigned to the object. For example, for {FieldDevice2} AL1, select AL1 for the <i>Connector</i> attribute. See also the JCI_SI8_Full-IO template on page 6-9.
Security Supervised Input	{FieldDevice3} AL1 {FieldDevice3} AL2 {FieldDevice3} AL3 {FieldDevice3} AL4 {FieldDevice3} AL5 {FieldDevice3} AL6 {FieldDevice3} AL7 {FieldDevice3} AL8	Add eight Security Supervised Input objects. These objects represent inputs AL1 through AL8 on the SI8 module. Parent Hardware: {FieldDevice3} SI8 Connector: The connector selected should correspond to the name assigned to the object. For example, for {FieldDevice3} AL1, select AL1 for the <i>Connector</i> attribute. See also the JCI_SI8_Full-IO template on page 6-9.

1. The names in this column are suggestions. You may enter any name you wish.



Object Modifications

Modify the following object(s) in the template:

Table 7-25: Object Modifications (Supervised Alarm, Tamper and Trouble Inputs Application)

Object	Attribute	Modification
{Zone01} Zone	Alarm Input Attribute List	{FieldDevice} AL1
	Tamper Input Attribute List	{FieldDevice2} AL1
	Trouble Input Attribute List	{FieldDevice3} AL1
{Zone02} Zone	Alarm Input Attribute List	{FieldDevice} AL2
	Tamper Input Attribute List	{FieldDevice2} AL2
	Trouble Input Attribute List	{FieldDevice3} AL2
{Zone03} Zone	Alarm Input Attribute List	{FieldDevice} AL3
	Tamper Input Attribute List	{FieldDevice2} AL3
	Trouble Input Attribute List	{FieldDevice3} AL3

Table 7-25: Object Modifications (Supervised Alarm, Tamper and Trouble Inputs Application)

Object	Attribute	Modification
{Zone04} Zone	Alarm Input Attribute List	{FieldDevice} AL4
	Tamper Input Attribute List	{FieldDevice2} AL4
	Trouble Input Attribute List	{FieldDevice3} AL4
{Zone05} Zone	Alarm Input Attribute List	{FieldDevice} AL5
	Tamper Input Attribute List	{FieldDevice2} AL5
	Trouble Input Attribute List	{FieldDevice3} AL5
{Zone06} Zone	Alarm Input Attribute List	{FieldDevice} AL6
	Tamper Input Attribute List	{FieldDevice2} AL6
	Trouble Input Attribute List	{FieldDevice3} AL6
{Zone07} Zone	Alarm Input Attribute List	{FieldDevice} AL7
	Tamper Input Attribute List	{FieldDevice2} AL7
	Trouble Input Attribute List	{FieldDevice3} AL7
{Zone08} Zone	Alarm Input Attribute List	{FieldDevice} AL8
	Tamper Input Attribute List	{FieldDevice2} AL8
	Trouble Input Attribute List	{FieldDevice3} AL8

Object Deletions

Although not required, you may delete the Eight_Zone_Area_No_Keypad template's {Area} Annunciator object.

Object Hierarchy

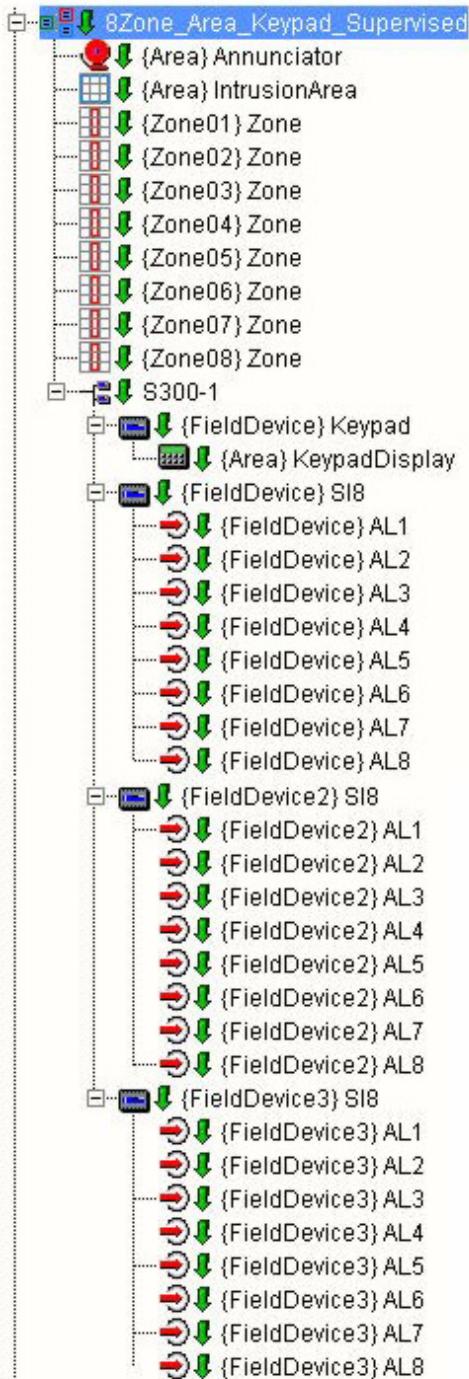


Figure 7-15: Object Hierarchy for Supervised Alarm, Tamper and Trouble Inputs Application

Double Knock and Paired Sensors

Application Description

Intrusion detection systems can be configured to go into alarm when two sensors are triggered (paired sensors) or when a single sensor is triggered twice (double knock). This helps prevent false alarms. See “Paired Sensors or Double Knock Detection” on page 4-27 for more information.

Configuring the CK722 Intrusion Detection System to use the double knock or paired sensors feature is a simple matter of editing the Intrusion Zone object’s *Operation Mode* attribute. This attribute has the following selectable values:

- **Momentary Alarms** – When a single sensor is tripped in the zone, the system goes into alarm.
- **Paired Alarms** – When two sensors are tripped within the zone, the system goes into alarm. These sensors can be two physically separate detectors or a dual detector (a single detector with two different sensors).
- **Double Knock** – When the same sensor is triggered twice in succession within the zone, the system goes into alarm.

Edit each Intrusion Zone object, as needed.

Using an Existing Template

Any of the existing intrusion legacy templates can be used to develop this application. When all areas, zones, annunciators, inputs and outputs are configured, simply modify the Intrusion Zone object’s *Operation Mode* attribute accordingly.

For more information on legacy templates, see page 6-63.

Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application

NOTE

This application uses the Eight_Zone_Area_Keypad_Annun legacy template. In order to create this application, you will need to import this template. For more information on legacy templates and how to import them, see “Legacy Templates” on page 63.

Application Description

This application demonstrates how multiple areas can be utilized and configured in the CK722 intrusion detection system (i.e. how a single area can control multiple areas). The following intrusion detection system application consists of the following:

- One building
- Four different tenant areas
- A fifth area that controls the arming/disarming for the entire building
- Four keypad/display modules, one for each tenant area
- 8 zones per area – 32 zones total
- 32 input sensors per area – 128 total
- Groups of two sensors wired in series to a single alarm input

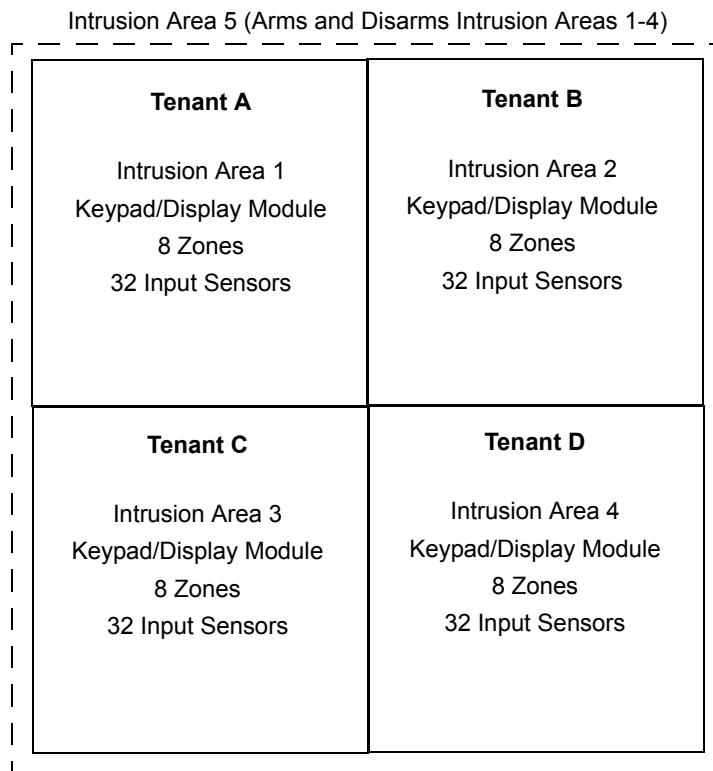


Figure 7-16: Controlling Multiple Intrusion Areas with a Single Area Across Multiple CK722 Controllers

The following scenario helps describe how this application functions:

1. The building owner has leased four areas of his building to four different tenants.
2. Each tenant requires intrusion detection.

3. The building owner divides the building into four intrusion areas, and each area consists of one keypad/display module, eight zones, and 32 input sensors.
4. Each tenant will have intrusion control over their own area.
5. However, the building owner wants the ability to arm/disarm all of the areas with a single command.
6. He configures this ability by adding another area (Intrusion Area 5), which will be used to control the other four areas.

NOTE

Areas may also be controlled across multiple controllers. For example, if Tenants A and B are controlled by CK722-1, and Tenants C and D are controlled by CK722-2, all areas managed by these controllers can be armed/disarmed via a single Intrusion Area object.

Using an Existing Template

The following instructions enable you to define a Multiple Intrusion Areas application using S300-I16 input modules and keypad/display modules.

Start by importing and copying the Eight_Zone_Area_Keypad_Annun legacy template and follow the instructions in this section to modify it. See page 6-63 for information on how to import legacy templates. See also page 7-2 for instructions on copying a template.

NOTE

Annunciation is not covered in this application example, even though the Eight_Zone_Area_Keypad_Annun template includes an Intrusion Announcer object. See the “One Door, Eight Zone, One Area Intrusion with Entry/Exit Time Application” on page 7-57 for an application example incorporating annunciation.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-26: Object Additions (Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application)

Object	Name ¹	Description
S300 Hardware Module	{FieldDevice} I16	Object representing the S300-I16 module. Destination: S300-1 Trunk object Hardware Module Type: I16
Security Supervised Input	{FieldDevice} AL1 {FieldDevice} AL2 {FieldDevice} AL3 {FieldDevice} AL4 {FieldDevice} AL5 {FieldDevice} AL6 {FieldDevice} AL7 {FieldDevice} AL8 {FieldDevice} AL9 {FieldDevice} AL10 {FieldDevice} AL11 {FieldDevice} AL12 {FieldDevice} AL13 {FieldDevice} AL14 {FieldDevice} AL15 {FieldDevice} AL16	Add sixteen Security Supervised Input objects. These objects represent inputs AL1 through AL16 on the I16 module. Parent Hardware: {FieldDevice} I16 Connector: The connector selected should correspond to the name assigned to the object. For example, for {FieldDevice} AL1, select AL1 for the <i>Connector</i> attribute. See also the JCI_I16_Full-IO template on page 6-5.

1. The names in this column are suggestions. You may enter any name you wish.

NOTE

No further additions are required to this template. When inserting the template, you will load four instances of this template, one for each tenant. The fifth Intrusion Area object (not part of the template) is added to the site once the template has been instantiated.

Object Modifications

Modify the following object(s) in the template:

Table 7-27: Object Modifications (Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application)

Object	Attribute	Modification
{Zone01} Zone	Entry Time	60 seconds
	Exit Time	60 seconds
	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL1 {FieldDevice} AL2 Two sensors are wired in series to the S300-I16 module's AL1 and AL2 terminal inputs for a total of four sensors.
{Zone02} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL3 {FieldDevice} AL4 Two sensors are wired in series to the S300-I16 module's AL3 and AL4 terminal inputs for a total of four sensors.
{Zone03} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL5 {FieldDevice} AL6 Two sensors are wired in series to the S300-I16 module's AL5 and AL6 terminal inputs for a total of four sensors.
{Zone04} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL7 {FieldDevice} AL8 Two sensors are wired in series to the S300-I16 module's AL7 and AL8 terminal inputs for a total of four sensors.

Table 7-27: Object Modifications (Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application)

Object	Attribute	Modification
{Zone05} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL9 {FieldDevice} AL10 Two sensors are wired in series to the S300-I16 module's AL9 and AL10 terminal inputs for a total of four sensors.
{Zone06} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL11 {FieldDevice} AL12 Two sensors are wired in series to the S300-I16 module's AL11 and AL12 terminal inputs for a total of four sensors.
{Zone07} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL13 {FieldDevice} AL14 Two sensors are wired in series to the S300-I16 module's AL13 and AL14 terminal inputs for a total of four sensors.
{Zone08} Zone	Alarm Input Attribute List	Select the following objects under the {FieldDevice1} I16 object: {FieldDevice} AL15 {FieldDevice} AL16 Two sensors are wired in series to the S300-I16 module's AL15 and AL16 terminal inputs for a total of four sensors.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

Figure 7-17 displays the object hierarchy prior to loading the template. During the instantiation process, four instances of the template will be loaded into the site.

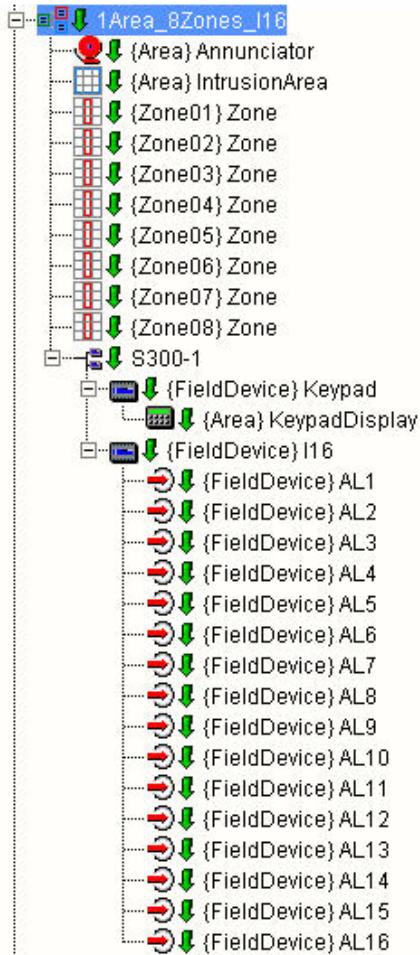


Figure 7-17: Object Hierarchy for Multiple Intrusion Areas and Unsupervised Inputs Wired in Series Application

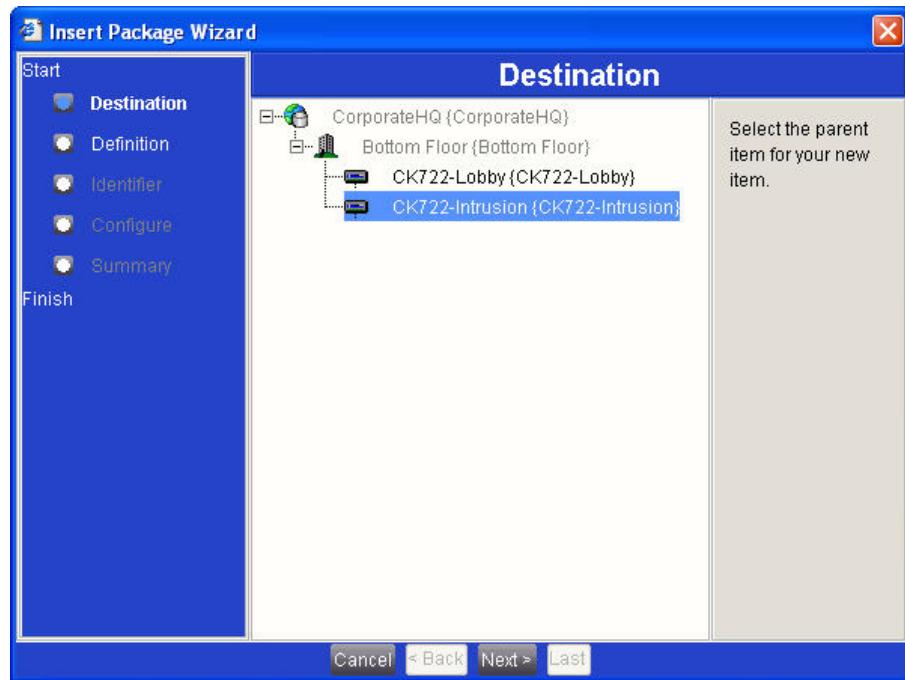
Template Instantiation (Loading)

Load the template for each of the four tenants. Confirm that you have at least one CK722 Device object available under the Site object for use when loading the template. If a CK722 Device object is not available, you must insert one prior to loading the template. For information on inserting a CK722 Device object, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

► To insert the template as a package:

1. From the P2000 SCT menu bar, select **Insert>Package**. The Insert Package Wizard appears.

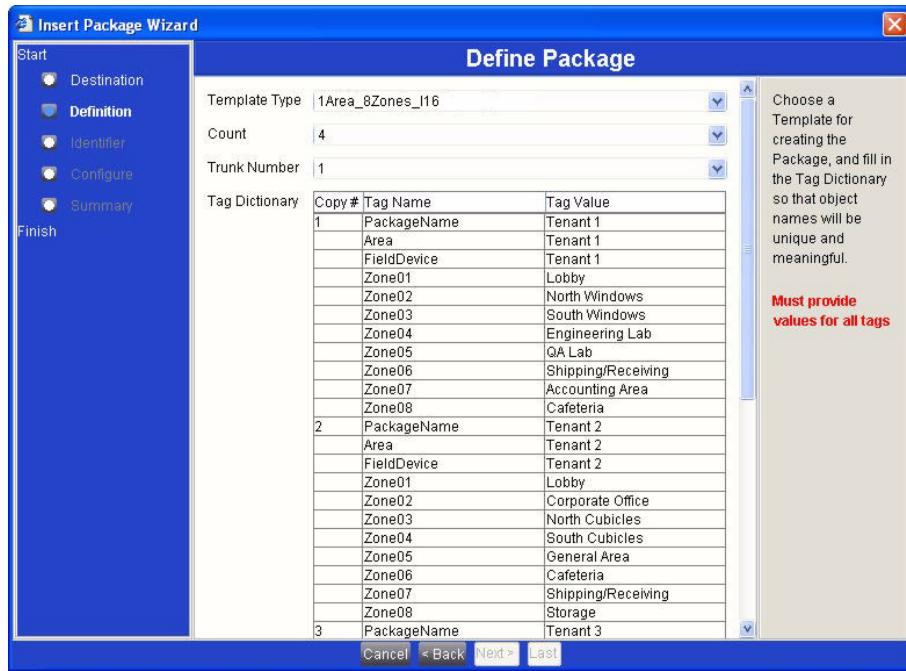
2. Select the CK722 controller that will receive the package and click **Next**.



3. On the Define Package screen, modify the fields according to Table 7-28:

Table 7-28: Modifications (Define Package Screen)

Field	Action/Modification
Template Type	Select the new template you created.
Count	4 (if loading four instances of the template)
Trunk Number	1 or 2, as needed.
Tag Dictionary	Assign tag values to each of the objects, including the PackageName field. If loading four instances, the PackageName can reflect each tenant (e.g. Tenant 1, Tenant 2, etc.). For more information on using the tag dictionary, refer to the <i>P2000AE System Configuration Tool (SCT) Manual</i> .



4. Click **Next**.
5. On the Define Points screen, click **Next**.
6. The Configure screen will be blank, unless you selected package attributes when defining the template's objects. See the *P2000AE System Configuration Tool (SCT) Manual* for information on package attributes.
7. Click **Next**.
8. Click **Finish**.

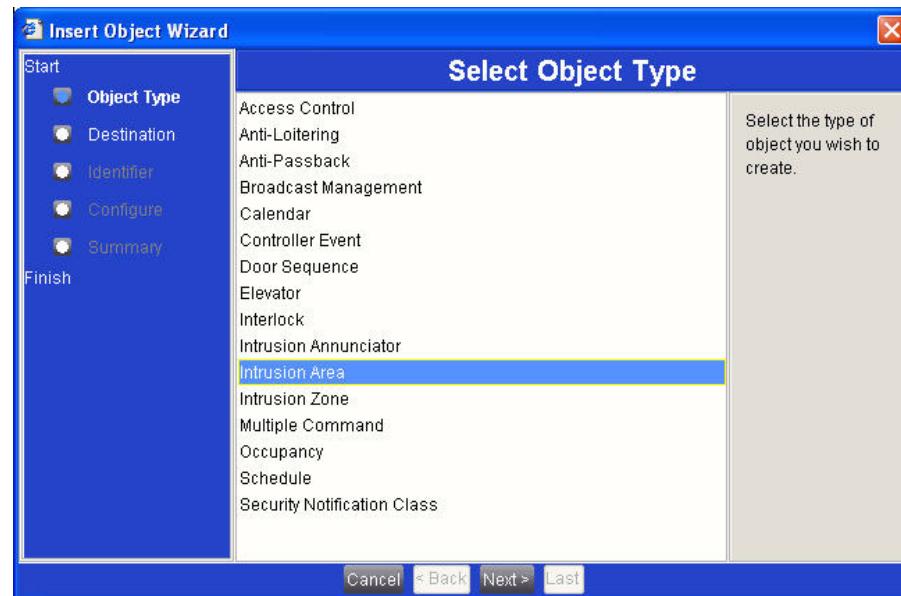
The inserted package appears under the selected CK722 Device object.

Adding Intrusion Area 5 to the Site

The last step for this application requires you to add an Intrusion Area object that can be used to arm and disarm the other four areas assigned to the tenants.

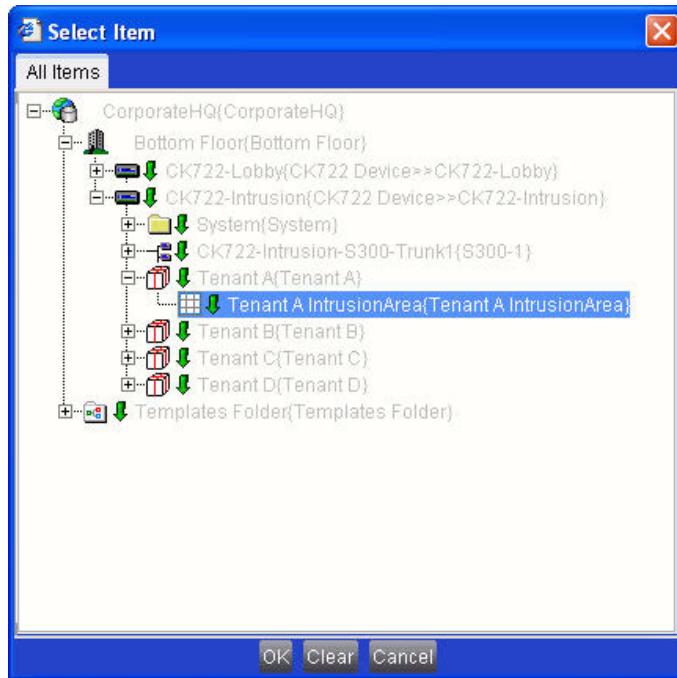
➤ To add an Intrusion Area to the site:

1. From the P2000 SCT menu bar, select **Insert>Object**. The Insert Object Wizard appears.
2. Select **Intrusion Area** and click **Next**.

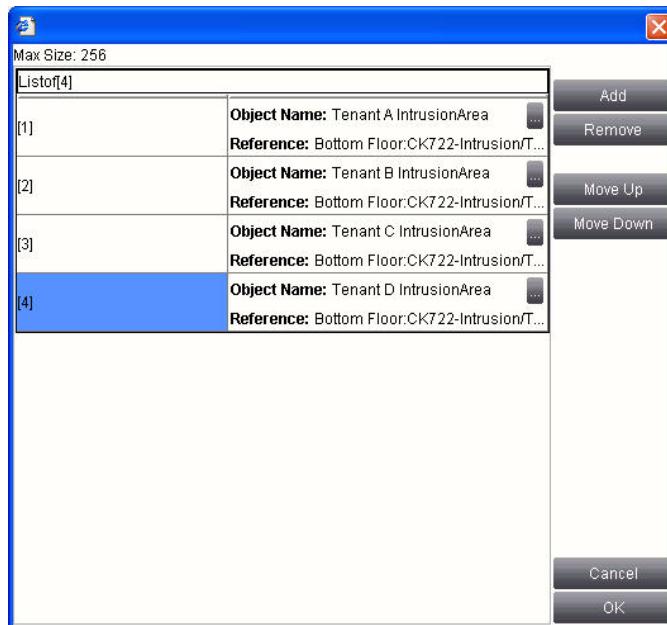


3. On the Destination screen, select the CK722 Device object that contains the inserted package.
4. Click **Next**.
5. On the Configure screen, assign a name to the Intrusion Area object (e.g. Intrusion Area 5).
6. Click the **Browse** button [...] next to the **Area Object List** field.
7. Click **Add**.
8. Click the **Browse** button [...] for row 1.
9. Select the Intrusion Area object assigned to the first tenant and click **OK**.

Make sure you select the Intrusion Area object that has been instantiated for the first tenant. Do not select an Intrusion Area object in the template.



10. Repeat steps 7 through 9 for the other three Intrusion Area objects assigned to the tenants.



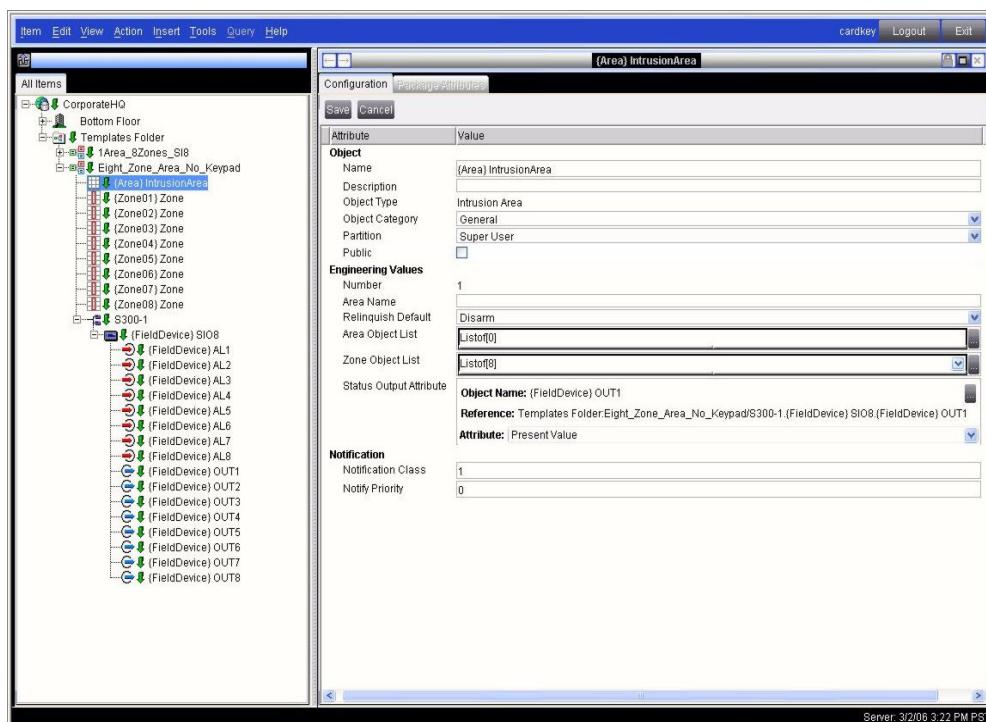
11. Click **OK**.
12. Click **Next** and then **Finish**.

Lighted Display Signaling Area Arm/Disarm

Application Description

This application enables a building guard or other authorized individual to see which areas are currently armed or disarmed, based on a display panel with LEDs that turn on when areas are armed, and turn off when areas are disarmed. The LEDs on this type of panel are triggered by the Intrusion Area object's *Status Output Attribute*. This attribute is linked to a Security Binary Output object. When the area is disarmed, the output is inactive. When the area is armed, the output becomes active.

As an example, if your intrusion system has eight areas, and you wish to use an LED display panel to view the armed/disarmed status of these areas, you would need eight Intrusion Area objects and eight Security Binary Output objects. Each Intrusion Area object's *Status Output Attribute* would be linked to one of the Security Binary Output objects.



Using an Existing Template

Any of the existing intrusion legacy templates can be used to develop this application. When all areas, zones, annunciators, inputs and outputs are configured, simply modify the Intrusion Area object's *Status Output Attribute* accordingly.

For more information on legacy templates, see page 6-63.

One Area, Seven Zones, No Keypad, with Keylock Arming/Disarming

Application Description

This application utilizes the JCI_SI8_Full-IO template (see page 6-9) and demonstrates how to configure the P2000 SCT to allow arming and disarming of the intrusion detection system using a keylock.

The following scenario helps describe how this application functions:

1. The entity Jane Doe is authorized to arm and disarm her company's intrusion detection system.
2. The intrusion detection system consists of twenty-one supervised input sensors (three are wired in series for each of the seven available alarm inputs), seven zones, one area, and a keylock input device (no keypad display module).
3. The keylock device is located in the lobby.
4. She arrives in the morning and must disarm the system.
5. She inserts her key into the keylock, rotates it 180 degrees counter-clockwise, and removes the key. The keylock is now in a secure state.
6. Through the use of an Interlock object, the P2000 SMS detects the keylock input is set to secure. The Interlock object sends the command to **Disarm** the area.
7. The P2000 SMS disarms the area.
8. When Jane Doe leaves, she must arm the area.
9. After the doors are locked, she inserts her key into the keylock, rotates it 180 degrees clockwise, and removes the key. The keylock is now in an alarm state.
10. Through the use of an Interlock object, the P2000 SMS detects the keylock input is set to alarm. The Interlock object sends the command to **Arm** the area.

Using an Existing Template

The following instructions enable you to define a One Area, Eight Zone, No Keypad, with Keylock Arming/Disarming application using an S300-SI8 input module. This application requires the JCI_SI8_Full-IO template.

Start by importing and copying the JCI_SI8_Full-IO template (see page 6-9) and follow the instructions in this section to modify it. See page 7-2 for instructions on copying a template.

Once you have completed the object additions, modifications, and deletions, insert the package into a CK722 Device object.

NOTE

Depending on your exact application requirements, your template may require additional modifications.

Object Additions

Add the following object(s) to the template:

Table 7-29: Object Additions (One Area, Seven Zones, No Keypad, with Keylock Arming/Disarming Application)

Object	Name ¹	Description
Intrusion Zone	{Zone01} Zone {Zone02} Zone {Zone03} Zone {Zone04} Zone {Zone05} Zone {Zone06} Zone {Zone07} Zone	Monitors and controls a group of sensors. Alarm Input Attribute List: For {Zone01} Zone, select {FieldDevice} AL1, for {Zone02} Zone, select {FieldDevice} AL2, etc. The {FieldDevice} AL8 object is used for the Keylock input device and is not assigned to a zone.
Intrusion Area	{Area} Intrusion Area	Arms and disarms its associated zone objects. Zone Object List: {Zone1} Zone {Zone2} Zone {Zone3} Zone {Zone4} Zone {Zone5} Zone {Zone6} Zone {Zone7} Zone
Interlock	{Keylock} Intrusion ILO	This object modifies the Intrusion Area object to arm and disarm the area. See page 7-83 for configuration instructions.

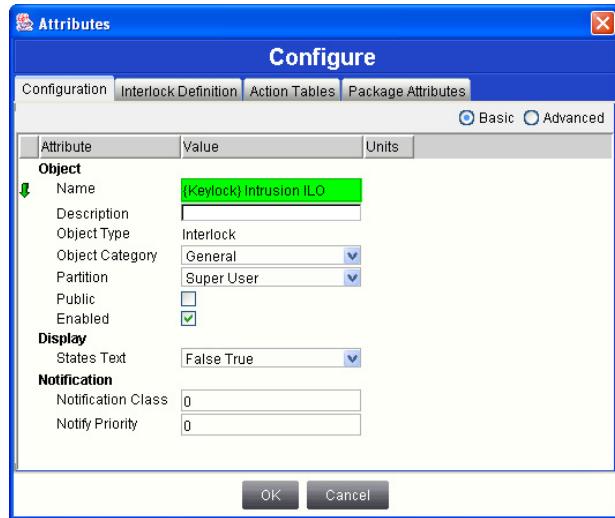
1. The names in this column are suggestions. You may enter any name you wish.

► **To add the {Keylock} Intrusion ILO object:**

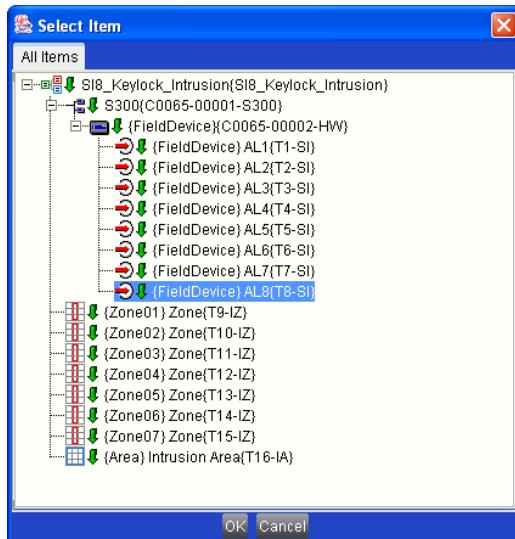
1. Drag and drop an **Interlock** object from the Object Palette to the Object Logic Diagram.

For information on inserting objects with the Object Palette, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

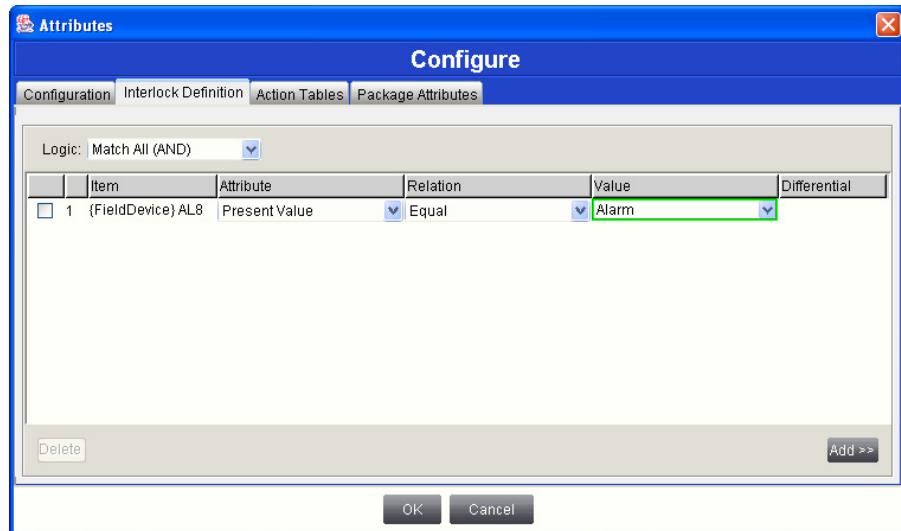
2. On the **Configuration** tab, enter a name for the Interlock object, such as {Keylock} Intrusion ILO, in the **Name** field.



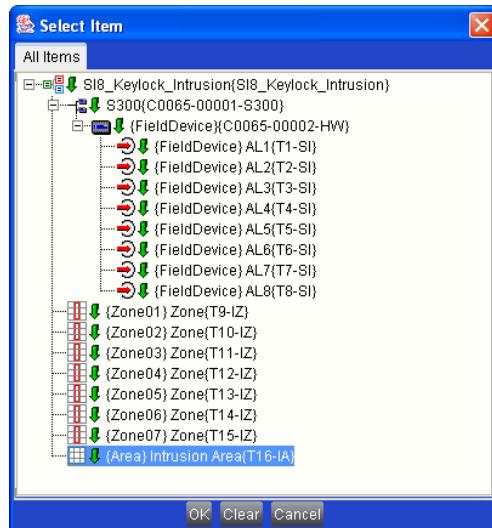
3. Select the **Interlock Definition** tab.
4. Verify the **Match All (AND)** option is selected in the **Logic** drop-down list.
5. Click **Add**.
6. Select the **{FieldDevice} AL8** object and click **OK**. This Security Supervised Input object represents the keylock input device.



7. In the Logic table, verify that **Equal** is listed under the **Relation** column.
8. Select **Alarm** under the **Value** column.

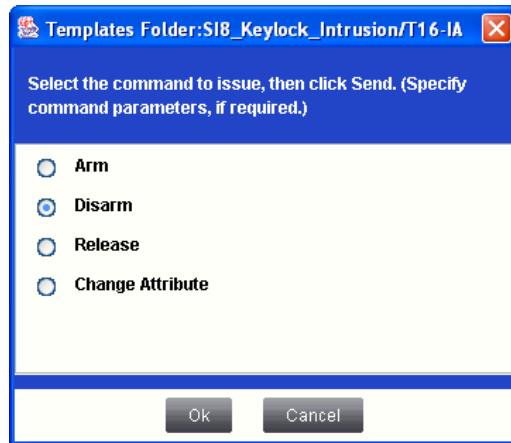


9. Select the **Action Tables** tab.
10. In the Actions for Condition: True area, click **Add**.
11. Select **{Area} Intrusion Area** and click **OK**.

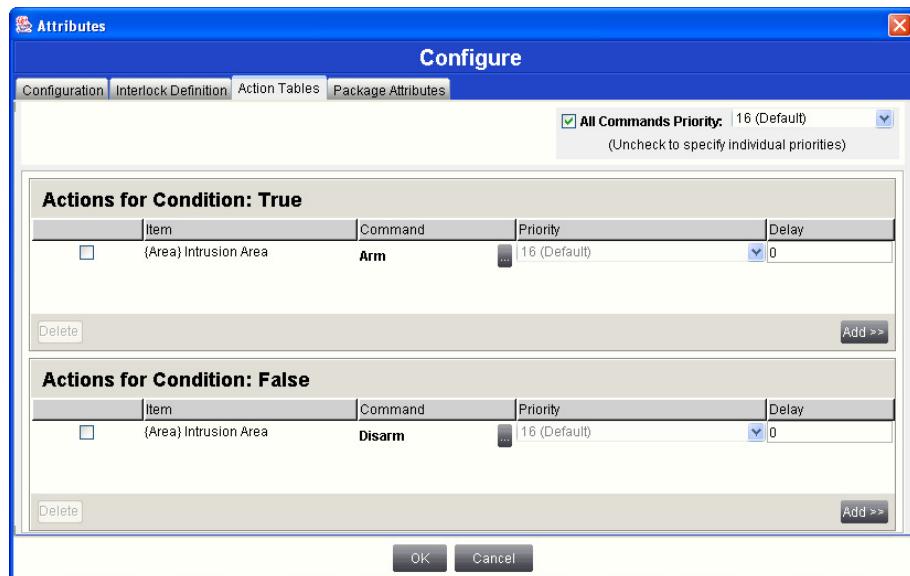


Do not change the default command (**Arm**) for the Actions for Condition: True.

12. Under Actions for Condition: False, click **Add**.
13. Select **{Area} Intrusion Area** and click **OK**.
14. Under the **Command** column, click the **Browse** button [...].
15. Select the **Disarm** radio button.



16. Click **Ok**.



17. Click **OK**.

18. Click **Save**.

Object Modifications

No objects need to be modified in the template.

Object Deletions

No objects need to be deleted from the template.

Object Hierarchy

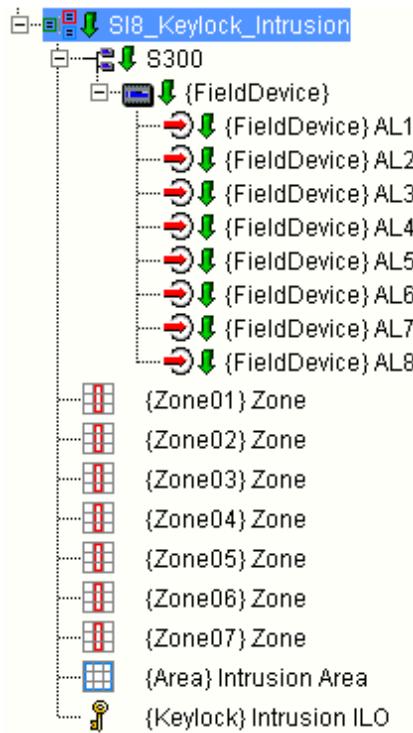


Figure 7-18: Object Hierarchy for One Area, Seven Zones, No Keypad, with Keylock Arming/Disarming Application

SCHEDULING

The scheduling feature allows you to automate certain functions such as unlocking a door for a defined amount of time during defined days of the week. For example, if the lobby door will only be unlocked during normal business hours, you could create a schedule to override the lobby's door's default settings and unlock the lobby door (Override - Open Mode) at 8:00 and lock it (Normal – Access Mode) at 17:00, Monday through Friday.

Each schedule consists of a **Weekly Schedule** and an **Exception Schedule** that use Time/Value Pairs (Events) for defining when events are applied to the defined list of Scheduled Items. The scheduling feature's user interface provides a graphical view for configuring the schedule and displaying when events are scheduled to occur.

SCHEDULE

The Schedule object resides in the CK722 controller and is the internal basis for the Scheduling feature. Create a Schedule and use the Scheduling feature to define the items and times when the Schedule writes values to the referenced attributes of the scheduled items. Each schedule includes the following:

- A weekly schedule that uses a set of time/value pairs for each day (Monday through Sunday) to configure the times when the Schedule writes values to its referenced attributes (Scheduled Items)
- An exception schedule that uses time/value pairs to configure exceptions to the weekly schedule (optional)
- A list of items to which the schedule applies to (referenced by the schedule, called Scheduled Items)
- An effective period that optionally allows you to enable/disable the weekly schedule for a selected period of time

Use the scheduling feature to:

- Create new schedules
- Edit existing schedules
- Add references to items for scheduling to a schedule using a Navigation Tree of the site
- Delete references to items for scheduling from a schedule

- Copy and paste a schedule to another device or item
- Enable or disable a schedule
- Change the effective period of a Schedule

Using a P2000 Time Zone

As a suggested alternative, you may select an existing P2000 Time Zone defined in the P2000 host software. This saves duplicate scheduling work if you wish to add a schedule that has already been defined in the host software. To select a P2000 Time Zone, select the time zone from the Schedule object's *Time Zone* attribute, located on the Configuration tab. See also "Linking Schedule Objects to P2000 Time Zones" on page 5-5.

Once you select a time zone, you will not be able to configure a Weekly Schedule or Exception Schedule on the Schedule object's Schedule tab.

NOTE

Before selecting a P2000 time zone in a Schedule object, do not define a Weekly Schedule or Exception Schedule, or remove the schedule data from their respective tabs. Once you select a P2000 time zone, the P2000 SCT prohibits edits to the Weekly Schedule and Exception Schedule tabs. If you have already defined a weekly schedule and exception schedule, and you select a P2000 time zone, edit the object, select <none> as the Time Zone attribute, remove the schedule data from the Weekly Schedule and Exception Schedule tabs, and re-select the desired time zone from the Time Zone attribute.

CALENDAR

The Calendar object resides in a CK722 controller and is the internal basis for the calendar functionality of the scheduling feature. Create a Calendar and use it to define exceptions to the weekly schedule. The schedule executes an exception schedule based on the list of dates defined in the calendar. For example, use a calendar to define different security operations on days, such as holidays, when the building is not occupied.

A schedule can reference calendars on any CK722 on the site.

Use the calendar feature to:

- Create new calendars
- Edit existing calendars
- Copy and paste a calendar to another device or item
- Toggle the calendar view between a graphical, one-month view, or 12-month view

- Enable or disable a calendar

Table 8-1 describes the functions of the Calendar's buttons.

Table 8-1: Calendar Button Functions

Button	Function
	Shows the Date, Date Range, and Week and Day details of the calendar entries and allows you to create new entries or edit existing entries when in Edit mode.
	Increases or decreases the currently displayed year by one year.
	Increases or decreases the currently displayed month by one month.
	Returns the display to the current month.
	Toggles the calendar display between one-month and 12-month view.

WEEKLY SCHEDULE

The scheduling feature has a weekly schedule that provides a schedule for each day of the week, Monday through Sunday. Each day consists of a set of time/value pairs that initiates the writing of values to the referenced attributes of the scheduled items based on the day of the week. For example, you can schedule a door to unlock at 8:00 and lock at 17:00, Monday through Friday. To do this, you use two time/value pairs, one to unlock the door at 8:00 in the morning and another to lock the door at 17:00 in the evening.

EXCEPTION SCHEDULE

The exception schedule provides a list of exceptions to the weekly schedule. Exceptions define a different set of time/value pairs for particular days in the weekly schedule. An exception can be defined for a date, a date range, a week of the month and day of the week (week and day) entry, or a reference to a Calendar. Table 8-2 contains information on the four types of exceptions.

Table 8-2: Exception Types

Exception Type	Description
Date	Occurs on a specific date.
Date Range	Occurs during a range of dates.
Week and Day	Occurs on certain weekdays during certain weeks of certain months. For example, the second Tuesday in any month or any Wednesday in June.
Calendar Reference	Occurs during selected dates in a calendar. Associating an exception with a calendar allows you to reference a large number of dates.

Each exception has a set of time/value pairs that run during the exception time period. Recall the example in the “Weekly Schedule” section where you scheduled a door to unlock at 8:00 and lock at 17:00, every Monday through Friday. You can create a date type exception to this weekly schedule if you want the door to unlock and lock at different times for a particular day, date range, week and day, or days defined in the calendar. For example, if a business will be open fewer hours than normal during a particular week, you can create a date range exception for the business week starting Monday, December 11, 2006 until Friday, December 15, 2006, and then define two time/value pairs for the new door unlock and lock times (e.g. unlock at 10:00 and lock at 14:00).



You can assign a priority level (1-16, where 1 is the highest priority) to each exception, regardless of type, to determine the order of execution. If you define two or more exceptions for the same date, the exception with the highest priority executes. For example, an exception with a priority of 2 executes over an exception with a priority of 8 if they are both defined for the same date.

If all exceptions for the same date have the same priority, the exception that appears first in the exception list executes. Therefore, if you have multiple exceptions defined for one day, the exceptions apply by priority level and then by the first item in the exception list. Avoid this type of situation by not assigning more than one exception on the same day with the same priority.

TIME/VALUE PAIRS (EVENTS)

A time/value pair describes the time when the scheduling feature writes user-defined values to the referenced item attributes (Scheduled Items). A defined time/value pair may also be referred to as an event. Weekly schedules and exception schedules use time/value pairs to define when events are scheduled to occur.

Each day of a weekly schedule and each exception day of an exception schedule has its own set of time/value pairs. When the time defined by the time/value pair arrives, the scheduling feature writes the defined value of the time/value pair to the referenced attribute of each item in the list of scheduled items.

SCHEDULED ITEMS

Scheduled items are the items referenced by a schedule in the list of scheduled items. The scheduling feature writes the defined value to the scheduled items at the time defined by the Time/Value Pairs. Schedules can reference any item that can be mapped into a CK722 controller. Adding items to a schedule involves selecting the item and then selecting the attribute of the item that you want the schedule to change. At times defined by time/value pairs, the schedule writes defined values to the attribute (scheduled attribute) associated with the referenced item (item name).

A single schedule can only apply values of one data type to all items in its list of scheduled items, therefore, the first item you add to the scheduled item list becomes the key item. The key item determines the data type of the values that the schedule writes to the attributes of all scheduled items. See the following **Example Scenario** for a description of how scheduled items and the key item work.

Example Scenario:

You can trigger a relay to turn on the lights. The relay is represented as a Binary Output in the P2000 SCT and when the *Present Value* attribute of the Security Binary Output object is Active, the relay is triggered and turns on the lights. To do this, add the Security Binary Output object to a schedule (item name) and then select *Present Value* as the scheduled attribute. See “Adding Scheduled Items” on page 8-18.

Table of Scheduled Items		
	Item Name	Scheduled Attribute
Add	Lobby Lights OUT01	Present Value
Remove	Lobby Door Seqnce	Portal Mode

← First item = Key item

Suppose the Security Binary Output object in this scenario is the key item because it is the first item in the list of scheduled items and the *Present Value* attribute is the scheduled attribute. Because the *Present Value* attribute of a binary output has a value of 0 or 1, the schedule can only write 0 or 1 to the scheduled attributes of the items appearing below it in the list.

Suppose you add a Door Sequence object below the Security Binary Output object and select the *Portal Mode* attribute of the DSO as the scheduled attribute. Because the *Portal Mode* attribute of a DSO has different values (Normal Access Mode, Override Open Mode, and Lockdown Secure Mode) than the Security Binary Output object (0 or 1), only values 0 or 1 (meaning Off) or 1 (meaning On) can be written to the DSO attribute because the schedule writes values to all items in the list of scheduled items using the data type of the Security Binary Output object, which is the key item. As long as the numeric value for the scheduled attribute fits your application, you can schedule it (e.g. 0 and 1 can be written to Booleans, Bytes, Unsigned 16, Unsigned 32, Signed 16, Signed 32, and Enumerations).

To change the key item, see “Editing Scheduled Items” on page 8-18.

DATES – CALENDAR ENTRY AND EXCEPTION SCHEDULE

You can use the following date formats for defining exceptions to weekly schedules using calendar entries or exception schedules:

- Date
- Date Range
- Week and Day

Date/Date Range

You can use a single date or a range of dates when defining exceptions to a weekly schedule. Table 8-3 describes the options available for defining a single date (January 1, 2006 for example) or range of dates (November 25, 2006 through November 26, 2006, for example) calendar or exception schedule entry.

Table 8-3: Date/Date Range Format

Menu	Options
Month	any, January through December
Day	any, 1 through 31
Year	any
Day of Week	any, Monday through Sunday

NOTE

For exception schedule entries, you also need to define the Priority at which the schedule executes the exception schedule.

Week and Day

You can use a week and day entry when defining exceptions to weekly schedule. Table 8-4 describes the options available for defining a week and day calendar or exception schedule entry (Saturday of the first week in May, for example). Week and day entries apply to all years.

Table 8-4: Week and Day Format

Menu	Options
Month	any, January through December
Week of Month	1 = Days 1 through 7 2 = Days 8 through 14 3 = Days 15 through 21 4 = Days 22 through 28 5 = Days 29 through 31 6 = Last seven days of the month
Day of Week	any, Monday through Sunday

NOTE

For exception schedule entries, you also need to define the Priority at which the schedule executes the exception schedule.

Wild Cards

You can use wild cards for defining specific calendar or schedule exceptions. Wild cards are the fields of an exception schedule or calendar entry date specified by the **any** selection. For example, an exception date of December 25, any year, and any day of week means the events defined in the exception schedule occur on December 25 of every year regardless of the events defined in the weekly schedule. To use wild cards, select any from the drop-down list of the Exception Detail or Calendar Entry dialog box.

For exceptions, the wild card part (field) of the date appears as **any** in the list of exceptions. For example, **December 25, any year Priority 8** appears in the list of exceptions after saving the selections described in Table 8-5 for a single Date entry.

Table 8-5: Schedule Wild Card Example – Single Date

Parameter	Value
Month	December
Day	25
Year	any
Day of Week	any
Priority	8

For calendars, the wild card part (field) of the date appears as any in the calendar entries list. For example, **February 2, any year** appears in the calendar entries list after saving the selections described in Table 8-6 for a single Date entry.

Table 8-6: Calendar Wild Card Example – Single Date

Parameter	Value
Month	February
Day	2
Year	any
Day of Week	any

You can use wild cards for the following types of exceptions and calendar entries:

- Date
- Date Range
- Week and Day

Wild Cards – Date

For a single date, a wild card opens a particular field to the full range of possible values for that field. Table 8-7 describes the meaning for each type of unspecified wild card date field.

Table 8-7: Unspecified Wild Card Date Field Meanings

Unspecified Field	Meaning
Month	Every month
Day of Month	Every day of the month, unless a particular day of the week is specified by the Day of Week field
Year	Every year
Day of Week	Every day of the week, unless a particular day of the month is specified by the Day of Month field

Table 8-8 shows some sample wild card dates and their meanings.

Table 8-8: Sample Wild Card Date Meanings Wild Cards – Date Range

Month	Day of Month	Year	Day of Week	Meaning
any	01	any	any	The first day of every month, every year
Mar	05	any	any	March 5 of every year
Mar	05	2006	any	March 5 of 2006
Mar	any	2006	any	Every day in March 2006

Table 8-8: Sample Wild Card Date Meanings Wild Cards – Date Range

Month	Day of Month	Year	Day of Week	Meaning
any	any	any	any	Every day
any	any	any	Friday	Every Friday
any	any	2006	Friday	Every Friday in 2006
Mar	any	2006	Friday	Every Friday in March of 2006
Mar	14	2006	Friday	Only on Friday, March 14, 2006
Apr	15	any	Monday	Every April 15 that occurs on a Monday

Wild Cards – Date Range

You can use wild cards for a range of dates. If the same field is a wild card in both the start date and end date, the range from start to end is infinite, except as limited by the other date fields. The Day of Week field is considered only when it is specified for both the start date and the end date. Table 8-9 shows some sample wild card date ranges and their meanings.

Table 8-9: Sample Wild Card Date Range Meanings Wild Cards – Week and Day

Start Date				End Date				Meaning
Month	Day of Month	Year	Day of Week	Month	Day of Month	Year	Day of Week	
any	any	any	Wednesday	any	any	any	Friday	Every Wednesday, Thursday, and Friday of every month, every year
any	any	2006	Wednesday	any	any	2006	Friday	Every Wednesday, Thursday, and Friday of every month in 2006
Jun	any	2006	Wednesday	Aug	any	2006	Friday	Every Wednesday, Thursday, and Friday in June, July, and August in 2006
Jun	any	2006	any	Aug	any	2006	any	Every day in June, July, and August in 2006
Aug	30	2006	any	any	any	any	any	Every day after August 30, 2006
any	10	any	any	any	12	any	any	The tenth, eleventh, and twelfth days of every month, every year

Table 8-9: Sample Wild Card Date Range Meanings Wild Cards – Week and Day

Start Date				End Date				Meaning
Month	Day of Month	Year	Day of Week	Month	Day of Month	Year	Day of Week	
Jul	04	any	Monday	Jul	04	any	Friday	Every July 4 that is a Monday, Tuesday, Wednesday, Thursday, or Friday, every year

Wild Cards – Week and Day

You can use wild cards for Week and Day entries. The Week and Day selection identifies a month, a week of the month, and a day of the week. The possible selections include **any** (every) as the week of the month, 1 (Days 1-7), 2 (Days 8-14), 3 (Days 15-21), 4 (Days 22-28), 5 (Days 29-31), and 6 (the Last seven days). Table 8-10 shows some sample wild card week and day exceptions.

Table 8-10: Sample Wild Card Week and Day Exceptions

Month	Week of Month	Day of Week	Meaning
any	any	any	Every day of the year
any	1 Days 1-7	Monday	Monday of the first week of every month
Jan	any	Tuesday	Every Tuesday in January
Feb	2 Days 8-14	any	Every day of the week in the second week of February
any	any	Wednesday	Wednesday of every week of every month
Mar	any	any	Every day of every week in March
any	3 Days 15-21	any	Every day of the third week of every month
Apr	4 Days 22-28	Thursday	Thursday in the fourth week of April
any	5 Days 29-31	any	Last three days if every month with 31 days
Jun	6 Last seven days	Saturday	Saturday in the last seven days of June

EFFECTIVE PERIOD

The effective period defines the active or inactive date range for a schedule (weekly and exception schedule). By default, a schedule is always active.

For example, you can create two schedules with the same items/objects and configure one for the normal school year and the other for the summer off season. Then you can define the effective period of the normal school year schedule from September through mid-June and the summer off season schedule from mid-June through August of each year.

FAST CLOCK

A device containing schedules activates the fast clock feature when the current date and time are changed in the device, or the device restarts following a power failure or reset. The fast clock feature ensures the schedule and its associated scheduled items are in the correct state based on day and time of the device.

The fast clock evaluates the time/value pairs scheduled for the current day from midnight to the current time, to find the last scheduled time and value for the current day. If the fast clock does not find any scheduled time/value pairs, then the value associated with the latest time for the current day is used. Then the fast clock writes this value to all scheduled attributes of all items references by the schedule's list of scheduled items.

DATA CONSISTENCY CHECKING

The exception schedule or calendar verifies the existence of exception and calendar dates and the start date of a date range occurs earlier than the end date for schedules and calendars when:

- Exception and calendar dates have all four fields defined and do not use wild cards (Month, Day of Month, Year, and Day of Week). For example, if you entered Monday, October 7, 2003, in the Exception Detail or Calendar Entry Detail dialog box, a warning message appears because October 7, 2003, is not a Monday.
- Exception and calendar dates have Month, Day of Month, and Year defined and use a wild card for the Day of Week field. For example, if you entered February 29, 2003, and **any** in the Day of Week field of the Exception Detail or Calendar Entry Detail dialog box, a warning message appears because there is no 29th day in February, 2003.
- Exception and calendar date ranges have all fields of both the start and end dates defined. For example if you entered a start date of Tuesday, July 3, 2003, and an end date of Monday, July 7, 2003, in the Exception Detail or Calendar Entry Detail dialog box, a warning message appears because July 3, 2003, is not a Tuesday.

- Exception and calendar date ranges have Month, Day of Month, and Year defined and use a wild card for the Day of Week field for both the start and end dates. For example, if you entered a start date of August 27, 2003, and **any** in the Day of Week field, and an end date of August 5, 2003, and **any** in the Day of Week field in the Exception Detail or Calendar Entry Detail dialog box, a warning message appears because the end date occurs before the start date.

Because the Schedule can reference Calendars residing in another device, the scheduling feature performs the following calendar reference verification:

- If a schedule references a calendar in another device and that device goes offline, the schedule assumes that while the device is offline, the calendar's *Present Value* does not change. For example, a schedule residing on CK722-1 refers to a calendar residing on CK722-2 and the Calendar object's *Present Value* attribute is selected. However, if CK722-2 is offline at midnight, the schedule assumes the calendar's *Present Value* changes to False at midnight. This means that the exception schedule is not active.
- If a schedule refers to a calendar and that calendar is deleted, the schedule assumes the Calendar object *Present Value* is False and the exception schedule is not active.
- When the date changes (at midnight), the Schedule object waits 20 seconds to receive Change of Value (COV) messages from any Calendar objects. After this 20-second period of time, the Schedule object determines if the current day is an exception day and applies the weekly schedule or exception schedule accordingly.

NOTE

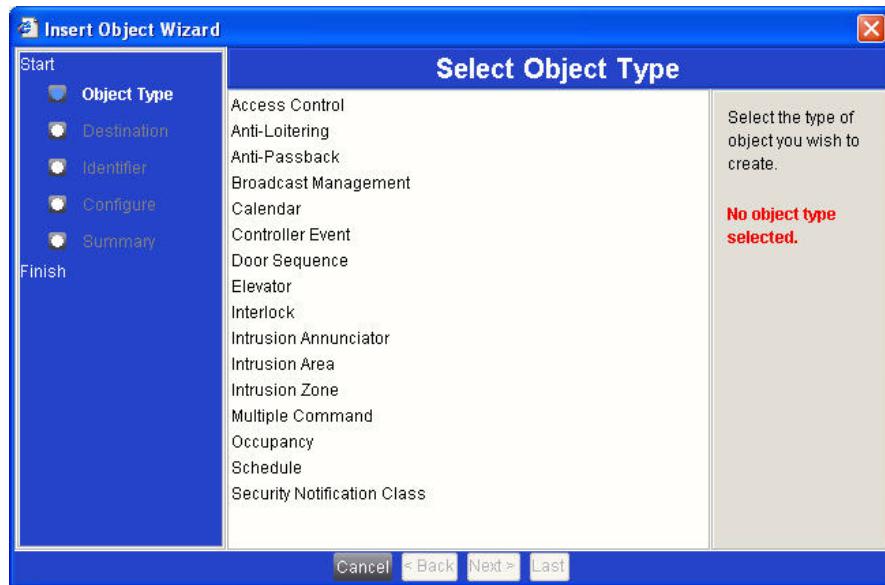
For events scheduled to execute at midnight, there may be a delay of up to 20 seconds after midnight due to Calendar object updates.

MANAGING SCHEDULES AND CALENDARS

Creating a Schedule or Calendar

To create a schedule or calendar:

1. From the **Insert** menu, select **Object**.
The Insert Object Wizard appears.



2. Select **Schedule** or **Calendar** and follow the Insert Object Wizard instructions.
If you skipped the **Configure** section of the Insert Object Wizard for a Schedule, you need to add at least one scheduled item before configuring the rest of the schedule. See “Adding Scheduled Items” on page 8-18.

Displaying an Existing Schedule or Calendar

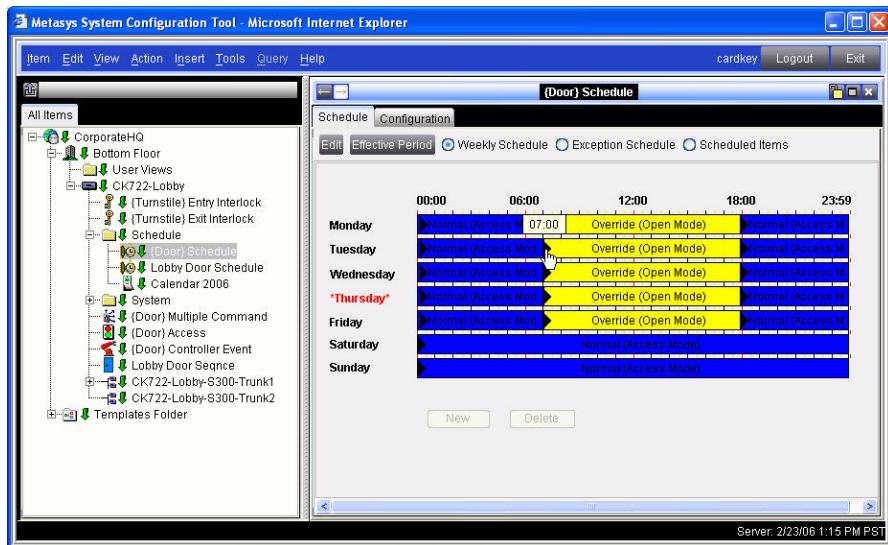
To display an existing schedule or calendar, drag and drop the item from the Navigation Tree into a display frame or double-click the item in the Navigation Tree.

Alternately, you can right-click on the item and select **View** from the menu that appears.

Displaying Scheduled Event (Time/Value Pairs)

To display scheduled events (time/value pairs):

1. Display a Schedule.
2. Select the **Weekly Schedule** radio button.
3. Place the cursor over a leading or trailing edge of a time bar segment. The time-of-day appears in 24-hour format in a small pop-up window.

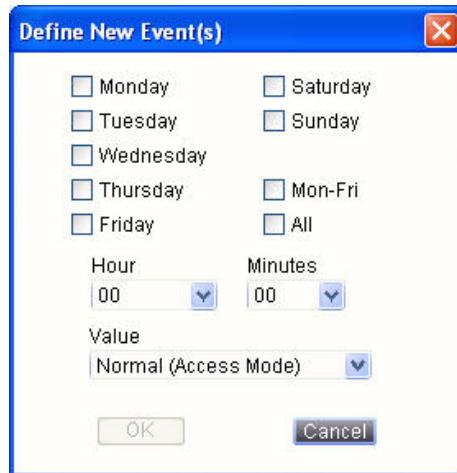


Adding Scheduled Events

You need to add at least one scheduled item before configuring the rest of the schedule. See “Adding Scheduled Items” on page 8-18.

To add scheduled events:

1. Display a Schedule.
2. Select the **Weekly Schedule** radio button.
3. Click **Edit**.
4. Click **New** or double click on a time bar segment and the Define New Event(s) dialog box appears.



Alternatively, you can right-click on a time bar segment and select **Add Event(s)**.

5. Select one or more days, the starting time (hour and minute), and a value for the event using the check boxes and drop-down lists and click **OK**.

6. Repeat Steps 4 and 5 as necessary.
7. Click **Save**.

Editing Scheduled Events

To edit scheduled events:

1. Display a Schedule.
2. Select the **Weekly Schedule** radio button.
3. Click **Edit**.
4. Double-click on the time bar segment. The Modify Event(s) dialog box appears.
Alternatively, you can select and right-click the event on the time bar and select **Modify Selected Event(s)** from the menu.
5. Adjust the starting time (hour and minute) and value as desired for the event using the drop-down lists and click **OK**.
6. Adjust the time of an event by dragging the beginning edge of the time bar segment to the right or left. When adjusting a time, a small pop-up window shows the digital time of day based on a 24-hour clock, and adjusts in 5-minute increments.
7. Click **Save**.

Deleting Scheduled Events

To delete scheduled events:

1. Display a Schedule.
2. Select the **Weekly Schedule** radio button.
3. Click **Edit**.
4. Select the event on the time bar and click **Delete**.
Alternately, you can select and right-click the event on the time bar and select **Delete Selected Event(s)** from the menu.
5. Click **Save**.

Copying and Pasting Events from One Day of the Week to Another

To copy and paste events from one day to another:

1. Display a Schedule.
2. Select the **Weekly Schedule** radio button.
3. Click **Edit**.

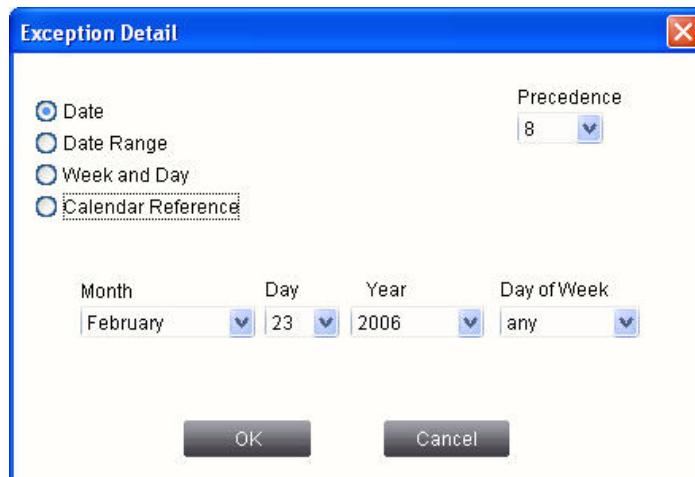
4. Select and right-click on the day (for example, Monday), and select **Copy** from the menu.
5. Select and right-click on the day to paste the event into and select **Paste** from the menu.

Adding Exception Schedules

To add exception schedules:

1. Display a Schedule.
2. Select the **Exception Schedule** radio button.
3. Click **Edit**.
4. Click the **New Exception** button to the right of the exception list to add exception schedules.

The Exception Detail dialog box appears.

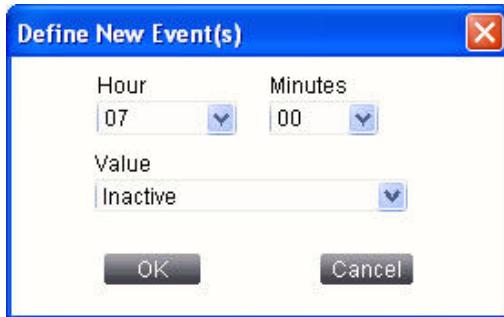


You can add exception schedules to the list using one of the following exception types:

- Date/Date Range
 - Week and Day
 - Calendar Reference
5. Select the radio button next to the type of exception.
 - Depending on the type of exception date selected, a different set of drop-down list options appears. The Calendar Reference exception type has a browse button that brings up a Navigation Tree for selecting a calendar.
 6. Make your selections from the drop-down lists or for the Calendar Reference, click the **browse** button and select a calendar from the Select Items Navigation Tree and click **OK**.
 7. Click **OK** on the Exception Detail dialog box.
 8. Select the exception schedule in the exception list.

9. Click the **New Event** button below the time bar or right-click on the time bar and select **Add Event(s)**.

The Define New Event(s) dialog box appears.



10. Select the starting time (hour and minute), and a value for the event using the check boxes and drop-down lists and click **OK**.
11. Repeat Steps 4 through 10 as necessary.
12. Click **Save**.

Editing Exception Schedules

To edit exception schedules:

1. Display a Schedule.
2. Select the **Exception Schedule** radio button.
3. Click **Edit**.
4. Double-click on the exception schedule.
The Exception Detail dialog box appears.
5. Make your modifications using the drop-down lists or for the Calendar Reference, click the **browse** button and select a calendar from the Select Items Navigation Tree. Click **OK**.
6. Click **OK** on the Exception Detail dialog box.
7. Select the exception schedule in the exception list.
8. Select and right-click on the event in the time bar and select **Modify Selected Event(s)** from the menu.
The Modify Event(s) box appears.
9. Select the starting time (hour and minute), and a value for the event using the check boxes and drop-down lists and click **OK**.
10. Repeat Steps 4 through 9 as necessary.
To change the value of an event, select and right-click an event and select a value from the menu.
To delete events, select and right-click the event and select delete selected event(s) from the menu.

11. Adjust the start and end times of events by dragging the edge of the time bar to the right or left.
12. Click **Save**.

Removing Exception Schedules

To remove exception schedules:

1. Display a Schedule.
2. Select the **Exception Schedule** radio button.
3. Click **Edit**.
4. Select the exception in the exception list.
5. Click **Delete**.
6. Click **Save**.

Adding Scheduled Items

To add scheduled items:

1. Display a Schedule.
2. Select the **Scheduled Items** radio button.
3. Click **Edit**.
4. Click **Add**.
5. Select one or more items from the Select Item Navigation Tree and click **OK**.
To select multiple items, use the <Ctrl> or <Shift> key.
6. Select the attribute that you want to schedule from the **Scheduled Attribute** drop-down list for each item name.
7. Click **Save**.

The first item in the list of scheduled items (Table of Scheduled Items) determines the data type used when the schedule writes values to the attributes of the scheduled item. See “Scheduled Items” on page 8-5.

Editing Scheduled Items

To edit scheduled items:

1. Display a Schedule.
2. Select the **Scheduled Items** radio button.
3. Click **Edit**.
4. Right-click on the item name and select an option from the pop-up menu.
For example, select make key item to promote the selected item to the key item of the schedule.

Important: If you promote an item to key item, all values in the time/value pairs of the scheduled events change to a default value. The default for numeric values is 0 or 0.0. The default for attributes that support multiple values is the first value available in the set of possible values. After changing the key item, we strongly recommend reviewing and adjusting the values for all time/value pairs in the schedule as necessary.

5. Select an option from the drop-down list of the scheduled attribute. For example, selecting *Present Value* means that the schedule writes a value to the *Present Value* attribute of that item.
6. Click **Save**.

Removing Scheduled Items

To remove scheduled items:

1. Display a Schedule.
2. Select the **Scheduled Items** radio button.
3. Click **Edit**.
4. Select the item name.
5. Click **Remove** or right-click and select remove object from the pop-up menu.

Important: If you remove the only remaining item, all values associated with the schedule disappear. Add a new scheduled item and make the necessary modifications to reinstate the schedule.

Editing the Effective Period of a Schedule

To edit the effective period of a schedule:

1. Display a Schedule.
2. Click **Edit**.
3. Click **Effective Period** in any tab of a schedule.

The Effective Period dialog box appears.



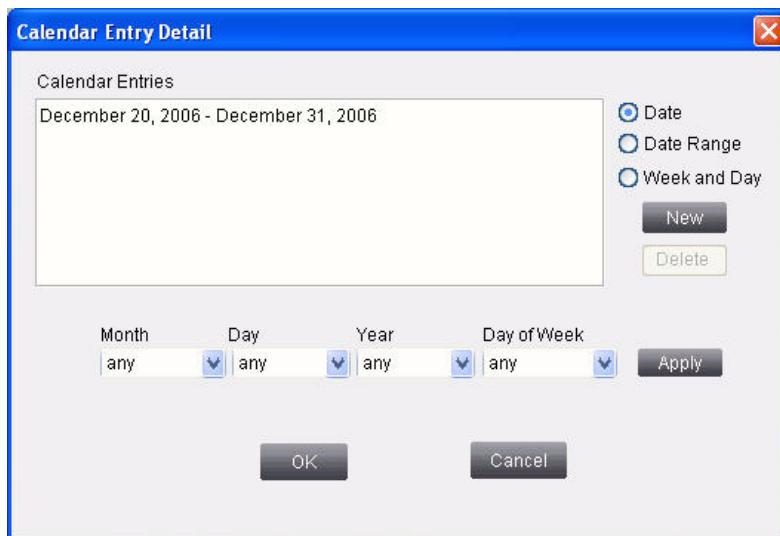
4. Select the start and end dates using the drop-down lists.
5. Click **Apply**.
6. Click **OK**.
7. Click **Save**.

Creating a New Calendar Entry

To create a new calendar entry:

1. Display a Calendar.
2. Click **Edit**.
3. Click **Entry Detail**.

The Calendar Entry Detail dialog box appears.



4. Click **New**.
 5. Select the **Date**, **Date Range**, or **Week and Day** radio button.
- Depending on the type of entry selected, a different set of drop-down list options appears.
6. Make your selections.
 7. Click **Apply**.
 8. Click **OK**.
 9. Click **Save**.

Alternately, create date and date range calendar entries visually by selecting an individual calendar date or by selecting and dragging the mouse over a range of dates.

Wild card date entries appear with a contrasting blue background on the calendar and have an asterisk next to the date. Once you create a wild card entry, visually selecting and deselecting it does not change the entry. To edit the entry, click **Entry Detail** on the calendar display frame.

Editing a Calendar Entry

To edit a calendar entry:

1. Display a Calendar.
2. Click **Edit**.
3. Click **Entry Detail**.

The Calendar Entry Detail dialog box appears.

4. Select the entry from the calendar entries list.
5. Use the drop-down lists to modify the entry.
6. Click **Apply**.
7. Click **OK**.
8. Click **Save**.

Alternately, edit Date and Date Range calendar entries visually by selecting an individual calendar date or by selecting and dragging the mouse over a range of dates.

Wild card date entries appear with a contrasting blue background on the calendar and have an asterisk next to the date. Once you create a wild card entry, visually selecting and deselecting it does not change the entry. To edit the entry, click **Entry Detail** on the calendar display frame.

Deleting a Calendar Entry

To delete a calendar entry:

1. Display a Calendar.
2. Click **Edit**.
3. Click **Entry Detail**.

The Calendar Entry Detail dialog box appears.

4. Select the entry from the **Calendar Entries** list.
5. Click **Delete**.
6. Click **OK**.
7. Click **Save**.

Alternatively, you can delete calendar entries visually by selecting the individual calendar date a second time or by selecting and dragging the mouse over the range of dates a second time. This method cancels the selection, thus deleting the entry.

Toggling Between Calendar Views

NOTE

The cursor turns into a hand when you place it over the header area of a calendar.

To toggle between calendar views:

1. Display a Calendar.
2. Place the cursor over the header of a calendar until it changes shape from an arrow to a hand and click.
The view switches between one-month and 12-month view.

Editing the Attributes of a Schedule or Calendar

To edit the attributes of a Schedule or Calendar:

1. Display a Schedule or Calendar.
2. Select the **Configuration** tab.
3. Click **Edit**.
4. Make your modifications using the drop-down lists and text boxes.
See the *Schedule Object Manual* or *Calendar Object Manual* for details about the attributes of these items.

NOTE

*Clearing the **Enabled** check box of the Schedule disables (turns off) the schedule.*

5. Click **Save**.

Copying and Pasting a Schedule or Calendar (Offline Mode Only)

To copy and paste a schedule or calendar:

1. Select the schedule or calendar from the Navigation Tree.
2. Select **Copy** from the **Edit** menu.
3. Select the desired destination for the copied schedule or calendar in the Navigation Tree.
4. Select **Paste** from the **Edit** menu.

The Paste Item Wizard appears.



5. Modify the name of the schedule or calendar as desired and click **Paste**.

Deleting a Schedule or Calendar

To delete a schedule or Calendar:

1. Select the schedule or calendar you want to delete in the Navigation Tree.
2. Select **Delete Items** from the **Edit** menu.

SECURED PREMISES NOTIFICATION SETTINGS

The steps described in this appendix are necessary to ensure UL 1076 compliance when a controller event is used to unsuppress (arm) life safety alarm signals.

Per UL 1076, if a user can unsuppress life safety alarms at the protected premises, e.g. through a controller event, then when this event is invoked the user must receive an audible or visible indication that the P2000 server received the message generated by the controller after the event was processed. If the user does not receive the expected indication, then either the controller is offline from the server or the controller did not process the controller event request.

Verify that your P2000 SMS's settings are configured according to the information provided in this appendix. Any parameters not specified may be programmed at the end-user's discretion.

SEQUENCE OF EVENTS

The following information describes a typical sequence of events given the configurations described in this chapter.

1. Applicable life safety alarm inputs are in a secure state and are not suppressed (i.e. they are armed).
2. An authorized person initiates (activates) a controller event, which suppresses (disarms) one or more life safety alarm input points.
3. All life safety alarm signals associated with the controller event are now suppressed and will not report to the host.
4. An authorized person deactivates the previously activated controller event.
5. All life safety alarm signals associated with the controller event are now unsuppressed (armed) and will report to the host (if the panel is online).
6. The P2000 SMS, having received the controller event deactivate message, initiates its event and sets the appropriate output point.
7. The output point activation causes an audible or visible indicator to be annunciated at the location where the controller event was deactivated.

P2000 SYSTEM CONFIGURATION TOOL (SCT) PROCEDURES

The following minimum objects must be inserted in the P2000 SCT:

- Site object
- CK722 Device object
- S300 Hardware Module object that supports at least one reader (e.g. RDR2S, RDR2)
- S300 Hardware Module object that supports input/output control (e.g. SIO8, IO8, RDR2S with available general purpose inputs/outputs, etc.)
- S300 Reader Terminal object (the device associated with this object triggers the controller event)
- Security Binary Output object (the output device that audibly or visibly annunciates to the entity that the life safety alarms are currently unsuppressed whenever the entity deactivates the controller event)
- Security Supervised Input object (alarm input point that is suppressed/unsuppressed by the activation/deactivation of the controller event)
- Access Control object (links with the Controller Event object)
- Intrusion Zone object (suppresses/unsuppresses its associated input points)
- Controller Event object (modifies the Intrusion Zone object accordingly)

The information in this section does not cover how to insert the objects listed. For information on inserting objects, refer to the *P2000AE System Configuration Tool (SCT) Manual*.

Once all objects are inserted, follow the procedures in this section to configure the objects for UL 1076 compliance. For additional attribute information, refer to the associated object manual.

Access Control Object Configuration

1. Open the Access Control object and click **Edit**.
2. Select the Controller Event object in the **Controller Event Object List** attribute (**Configuration** tab).
3. Click **Save**.

Intrusion Zone Object Configuration

1. Open the Intrusion Zone object and click **Edit**.
2. Select one or more alarm inputs in the **Alarm Input Attribute List** attribute. These alarm inputs will be suppressed/unsuppressed according to the controller event activation/deactivation.
3. Enter a value of 5 or more seconds in the **Exit Time** attribute. This is the length of time, in seconds, that the output will audibly or visibly annunciate when the controller event is deactivated and the alarm inputs are unsuppressed (armed).
4. In the **Delay Annunciator Output Attribute**, select the Security Binary Output object that will annunciate when alarm inputs are unsuppressed due to the deactivation of a controller event.
5. Click **Save**.

Controller Event Object Configuration

1. Open the Controller Event object and click **Edit**.
2. Select the Intrusion Zone object in the **Target Attribute** field. Verify that **Present Value** is selected as the attribute.
3. Verify that the **Target Value** attribute is set to **1**.
4. According to how you wish the controller event to be invoked from the keypad, follow the instructions in “Invoking Controller Events from a Keypad” on page B-5.
5. Click **Save**.

P2000 HOST PROCEDURES

In order to suppress/unsuppress life safety alarms, an entity’s Access Profile settings must have an **Event Privilege** equal to or greater than the Controller Event object’s *Event Privilege* attribute level.

Refer to the *P2000AE Software User Manual* for detailed instructions on modifying an entity’s Access Profile.

USING KEYPAD READERS

The following sections describe how to invoke access requests, Common PIN requests (previously called Air Crew requests), Card ID requests, Timed Overrides, and Controller Events using a keypad reader.

There is a 15 second (default) time-out on all keypads. Whenever the keypad is idle for more than 15 seconds, all keys entered so far will be ignored, and the entire key sequence needs to be re-entered. The number of seconds before the keypad times out can be modified using the Access Control object's *Keypad Timeout* attribute.

NOTE

For information on assigning identifiers to entities, refer to the P2000AE Software User Manual.

INVOKING ACCESS REQUESTS FROM A KEYPAD

► **To Invoke Access with an Access Badge Identifier:**

1. Set the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to the corresponding card format.
2. To be able to invoke access using an access badge identifier at any time, verify that the Access Control object's **Set n PIN Required** check boxes are cleared (default).
3. At the keypad reader, present the access badge identifier.

► **To Invoke Access with Card ID:**

1. Set the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Card ID**.
2. Verify that the Access Control object's **Set n PIN Required** check boxes are cleared (default).
3. At the keypad reader, enter the Card ID number and press the # key.

► **To Invoke Access with PIN and Card ID:**

1. Set the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Card ID**.

2. Select the Access Control object's **Set *n* PIN Required** check box.
3. At the keypad reader, enter PIN, then enter the Card ID number, and press the # key.

➤ **To Invoke Access Using PIN and Access Badge Identifier:**

1. Select the applicable access badge identifier format from the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set).
2. Select the Access Control object's **Set *n* PIN Required** check box.
3. At the keypad reader, enter PIN and then present the access badge identifier.

➤ **To Invoke Access with PIN and Access Badge Identifier, Allowing PIN After Access Badge Identifier:**

1. Select **Pound Key Only** from the Access Control object's **Keypad Trigger** attribute (Configuration tab).
2. Select the applicable access badge identifier format from the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set).
3. Select the Access Control object's **Set *n* PIN Required** check box.
4. At the keypad reader, present the badge, enter PIN and press the # key.

NOTE

You may present the access badge identifier at any time before pressing the # key.

INVOKING COMMON PIN ACCESS REQUESTS FROM A KEYPAD

➤ **To Invoke Common PIN Access:**

1. Set that the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Common PIN**.
2. To request Common PIN access, press the star (*) key, then press number 2, followed by the Common PIN number and the # key.

Invoking Timed Overrides from a Keypad

➤ **To Invoke Timed Override with Access Badge Identifier:**

1. The Door Sequence object's **Timed Override Mode** attribute must be set to a value *other than Not Allowed*.

2. The Access Control object must have at least one **Override Mask** check box (Privileges tab) enabled.
3. The entity's Access Profile (P2000 host software) must be granted the security flag privilege defined in the ACO. For example, if the ACO's **Override** check box is enabled (Privileges tab – Override Mask), the entity's Access Profile must have the Override security flag listed with a status of **Grant**.
4. To **start** Timed Override, press the star (*) key followed by number 0, enter the number of minutes, and present the access badge identifier.
5. To **stop** Timed Override, press the star (*) key followed by number 0 and present the access badge identifier.

➤ **To Invoke Timed Override with Card ID:**

1. The Door Sequence object's **Timed Override Mode** attribute must be set to **Timed Override** or **Timed Shunt**.
2. The Access Control object must have at least one **Override Mask** check box (Privileges tab) enabled.
3. The entity's Access Profile (P2000 host software) must be granted the security flag privilege defined in the ACO. For example, if the ACO's **Override** check box is enabled (Privileges tab – Override Mask), the entity's Access Profile must have the Override security flag listed with a status of **Grant**.
4. Set that the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Card ID**.
5. Verify that the Access Control object's **Set n PIN Required** check boxes are cleared (default).
6. To **start** Timed Override, enter the Card ID number, press the star (*) key followed by number 0, enter the number of minutes, and press the # key.
7. To **stop** Timed Override, enter the Card ID number, press the star (*) key followed by number 0, and press the # key.

➤ **To Invoke Timed Override with PIN and Card ID:**

1. The Door Sequence object's **Timed Override Mode** attribute must be set to a value *other than Not Allowed*.
2. The Access Control object must have at least one **Override Mask** check box (Privileges tab) enabled.
3. The entity's Access Profile (P2000 host software) must be granted the security flag privilege defined in the ACO. For example, if the ACO's **Override** check box is enabled (Privileges tab – Override Mask), the entity's Access Profile must have the Override security flag listed with a status of **Grant**.
4. Set the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Card ID**.
5. Select the Access Control object's **Set n PIN Required** check box.

6. To **start** Timed Override, enter PIN, enter the Card ID number, press the star (*) key followed by number 0, enter the number of minutes, and press the # key.
7. To **stop** Timed Override, enter the PIN, number, enter the Card ID number, press the star (*) key followed by number 0, and press the # key.

➤ **To Invoke Timed Override with PIN and Access Badge Identifier:**

1. The Door Sequence object's **Timed Override Mode** attribute must be set to a value *other than Not Allowed*.
2. The Access Control object must have at least one **Override Mask** check box (Privileges tab) enabled.
3. The entity's Access Profile (P2000 host software) must be granted the security flag privilege defined in the ACO. For example, if the ACO's **Override** check box is enabled (Privileges tab – Override Mask), the entity's Access Profile must have the Override security flag listed with a status of **Grant**.
4. Select the applicable access badge identifier format from the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set).
5. Select the Access Control object's **Set n PIN Required** check box.
6. To **start** Timed Override, enter PIN, press the star (*) key followed by number 0, enter the number of minutes, and present the access badge identifier.
7. To **stop** Timed Override, enter PIN, press the star (*) key followed by number 0, and present the access badge identifier.

➤ **To Invoke Timed Override with PIN and Access Badge Identifier, Allowing PIN After Access Badge Identifier:**

1. The Door Sequence object's **Timed Override Mode** attribute must be set to a value *other than Not Allowed*.
2. The Access Control object must have at least one **Override Mask** check box (Privileges tab) enabled.
3. The entity's Access Profile (P2000 host software) must be granted the security flag privilege defined in the ACO. For example, if the ACO's **Override** check box is enabled (Privileges tab – Override Mask), the entity's Access Profile must have the Override security flag listed with a status of **Grant**.
4. Select **Pound Key Only** from the Access Control object's **Keypad Trigger** attribute (Configuration tab).
5. Select the applicable access badge identifier format from the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set).
6. Select the Access Control object's **Set n PIN Required** check box.

7. To **start** Timed Override, enter PIN, press the star (*) key followed by number 0, enter number of minutes, present the badge¹, and press the # key.
8. To **stop** Timed Override, enter PIN, press the star (*) key followed by number 0, present the badge, and press the # key.

NOTE

You may present the access badge identifier at any time before pressing the # key.

INVOKING CONTROLLER EVENTS FROM A KEYPAD**► To Invoke Controller Events with an Access Badge Identifier:**

1. Insert a Controller Event object.
2. In the Controller Event object's **Event Code** attribute, enter the keypad code that will be used to trigger the controller event.
3. Verify the Controller Event object's **Trigger Type** is set to **Always On Positive Decision**. For information on other Trigger Type options, refer to the *Controller Event Object Manual*.
4. In the Access Control object, select the defined Controller Event object in the **Controller Event Object List** attribute.
5. To activate event, press the star (*) key followed by number 1, enter the keypad code, and present the access badge identifier.
6. To deactivate event, press the star (*) key followed by number 4, enter the keypad code, and present the access badge identifier.

► To Invoke Controller Events with Card ID:

1. Insert a Controller Event object.
2. In the Controller Event object's **Event Code** attribute, enter the keypad code that will be used to trigger the controller event.
3. Verify the Controller Event object's **Trigger Type** is set to **Always On Positive Decision**. For information on other Trigger Type options, refer to the *Controller Event Object Manual*.
4. In the Access Control object, select the defined Controller Event object in the **Controller Event Object List** attribute.
5. Set that the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Card ID**.
6. Verify that the Access Control object's **Set n PIN Required** check boxes are cleared (default).
7. To activate event, enter the Card ID number, press the star (*) key followed by number 1, enter the keypad code, and press the # key.

8. To deactivate event, enter the Card ID number, press the star (*) key followed by number 4, enter the keypad code, and press the # key.

➤ **To Invoke Controller Events with PIN and Card ID:**

1. Insert a Controller Event object.
2. In the Controller Event object's **Event Code** attribute, enter the keypad code that will be used to trigger the controller event.
3. Verify the Controller Event object's **Trigger Type** is set to **Always On Positive Decision**. For information on other Trigger Type options, refer to the *Controller Event Object Manual*.
4. In the Access Control object, select the defined Controller Event object in the **Controller Event Object List** attribute.
5. Set the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set) to **Card ID**.
6. Select the Access Control object's **Set *n* PIN Required** check box.
7. To activate event, enter PIN, enter the Card ID number, press the star (*) key followed by number 1, enter the keypad code, and press the # key.
8. To deactivate event, enter PIN, enter the Card ID number, press the star (*) key followed by number 4, enter the keypad code, and press the # key.

➤ **To Invoke Controller Events with PIN and Access Badge Identifier:**

1. Insert a Controller Event object.
2. In the Controller Event object's **Event Code** attribute, enter the keypad code that will be used to trigger the controller event.
3. Verify the Controller Event object's **Trigger Type** is set to **Always On Positive Decision**. For information on other Trigger Type options, refer to the *Controller Event Object Manual*.
4. In the Access Control object, select the defined Controller Event object in the **Controller Event Object List** attribute.
5. Select the applicable access badge identifier format from the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set).
6. Select the Access Control object's **Set *n* PIN Required** check box.
7. To activate event, enter PIN, press the star (*) key followed by number 1, enter the keypad code, and present the access badge identifier.
8. To deactivate event, enter PIN, press the star (*) key followed by number 4, enter the keypad code, and present the access badge identifier.

➤ **To Invoke Controller Events with PIN and Access Badge Identifier, Allowing PIN After Access Badge Identifier:**

1. Insert a Controller Event object.
2. In the Controller Event object's **Event Code** attribute, enter the keypad code that will be used to trigger the controller event.

3. Verify the Controller Event object's **Trigger Type** is set to **Always On Positive Decision**. For information on other Trigger Type options, refer to the *Controller Event Object Manual*.
4. In the Access Control object, select the defined Controller Event object in the **Controller Event Object List** attribute.
5. Select **Pound Key Only** from the Access Control object's **Keypad Trigger** attribute (Configuration tab).
6. Select the applicable access badge identifier format from the Access Control object's **First, Second, or Third Identifier Format** attribute (under any enabled Identification Set).
7. Select the Access Control object's **Set *n* PIN Required** check box.
8. To activate event, enter PIN, press the star (*) key followed by number 1, enter the keypad code, present the badge, and press the # key.
9. To deactivate event, enter PIN, press the star (*) key followed by number 4, enter the keypad code, present the badge, and press the # key.

NOTE

You may present the access badge identifier at any time before pressing the # key.

QUICK GUIDE TO USING KEYPAD READERS

Use the following quick guide to determine the key sequence at a keypad reader required for a particular action. This section assumes all P2000 host and SCT settings have already been configured for this action.

Legend

Keypad Code	Enter the Keypad Code.	badge	Present the access badge identifier.
PIN	Enter the PIN number.	* 0	
Card ID	Enter the Card ID number.	# 1	Press the specified key.
Minutes	Enter the number of minutes.		

Invoking Access Requests from a Keypad

With Access Badge Identifier

To request access: badge

With Card ID

To request access: Card ID #

With PIN and Card ID

To request access: PIN Card ID #

With PIN and Access Badge Identifier

To request access: PIN badge

With PIN and Access Badge Identifier, allowing PIN after Badge

To request access: PIN badge¹ #

¹⁾ The badge can be presented at any time before the # key is pressed (before, during or after the PIN is entered).

Invoking Common PIN Access Requests from a Keypad

* 2 Common PIN #

Invoking Timed Overrides from a Keypad

With Access Badge Identifier

To start override: badge

To stop override: badge

With Card ID

To start override: #

To stop override: #

With PIN and Card ID

To start override: #

To stop override: #

With PIN and Access Badge Identifier

To start override: badge

To stop override: badge

With PIN and Access Identifier, Allowing PIN After Badge

To start override: #

To stop override: #

¹⁾ The badge can be presented at any time before the # key is pressed (before, during or after the PIN and the Timed Override sequence are entered).

Invoking Controller Events from a Keypad

With Access Badge Identifier

To activate event: badge

To deactivate event: badge

With Card ID

To activate event: #

To deactivate event: #

With PIN and Card ID

To activate event: #

To deactivate event: #

With PIN and Access Badge Identifier

To activate event: badge

To deactivate event: badge

With PIN and Access Badge Identifier, Allowing PIN After Badge

To activate event: #

To deactivate event: #

¹⁾ The badge can be presented at any time before the # key is pressed (before, during or after the PIN and the Controller Event sequence are entered).

CONFIGURING OFFLINE MODE OPTIONS

When a hardware module (e.g. RDR2, RDR2S, or RDR2S-A) becomes disconnected from its supervisory controller (e.g. CK722), the P2000 SCT can be configured to perform one of the following actions when entities attempt to access a controlled area:

- Deny access to all entities who present an access badge identifier at a selected card reader. See “No Access in Offline Mode” on page 1.
- Grant access to entities who present an access badge identifier at a selected card reader, as long as the badge’s card type and facility code match the *Offline Card Type* and *Offline Facility Code* attributes of the S300 Reader Terminal object. See “Card Access in Offline Mode” on page 1.
- Grant access to entities who present an access badge identifier at a selected card reader *and* enter a valid Offline PIN at the keypad (the order of these actions can be reversed by modifying the S300 Reader Terminal object’s *Offline PIN After Card* attribute). See “Card and PIN Access in Offline Mode” on page 3.

NO ACCESS IN OFFLINE MODE

Configure this option to deny access to all entities who attempt to access a controlled area at a selected reader when the hardware module is offline with its supervisory device. This offline option is typically used in high security applications.

To deny access to all entities at a particular reader during offline mode, enter a value of **0** (zero) in the S300 Reader Terminal object’s *Offline Facility Code* attribute (0 is the default value).

CARD ACCESS IN OFFLINE MODE

Configure this option to grant access to entities who present an access badge identifier at a selected card reader in offline mode, as long as the badge’s card type and facility code match the *Offline Card Type* and *Offline Facility Code* attributes of the S300 Reader Terminal object. This offline option is typically used in low-to-medium security applications.

► To Configure Card Access in Offline Mode:

1. From the P2000 SCT, open the S300 Reader Terminal object that represents the reader you wish to configure for offline mode.

2. Click **Edit**.
3. From the *Offline Card Type* attribute, select the card type that will be accepted at the reader when in offline mode.

When attempting to access an area controlled by this reader, the entity's card type must match the card type selected in this field.

Table C-1: Selecting the Offline Card Type

If you select ...	Then ...
None	No entities will be able to access the area controlled by the reader.
Standard Wiegand	<p>The <i>Offline Facility Code</i> attribute you enter covers a total of four consecutive facility codes that are granted access in offline mode. These facility codes can be computed by dividing the entered facility code by 4, and multiplying the resulting integer number (i.e. ignoring the rest) by 4.</p> <p>This number and its immediate 3 higher integers are the facility codes that are granted access in offline mode.</p> <p>Example:</p> <ul style="list-style-type: none"> ■ Offline Facility Code: 85 ■ Divide by 4: $85 \div 4 = 21.25$ ■ Resulting Integer: 21 ■ Multiply Integer by 4: $21 \times 4 = 84$ ■ Acceptable Facility Codes: 84, 85, 86, 87
Encrypted Wiegand	Enter any numeric value for the <i>Offline Facility Code</i> attribute.
Binary BaFe	Enter a value of 4 for the <i>Offline Facility Code</i> attribute.
Magnetic Stripe	After saving the S300 Reader Terminal object, open the source Access Control object (i.e. the ACO linked to the S300 Reader Terminal object), select the Identification tab, and select Decoded Card as an <i>Identifier Format</i> attribute.
Eyecam Prox	Enter any numeric value for the <i>Offline Facility Code</i> attribute.
26 bit Sensor Forward	Enter any numeric value for the <i>Offline Facility Code</i> attribute.
26 bit Sensor Reverse	Enter any numeric value for the <i>Offline Facility Code</i> attribute.
HID Corporate 1000	Enter any numeric value for the <i>Offline Facility Code</i> attribute.

4. Enter a numeric value for the *Offline Card Type* attribute according to the Offline Card Type selected. See Table C-1.
5. Click **Save**.

CARD AND PIN ACCESS IN OFFLINE MODE

Configure this option to grant access to entities who present an access badge identifier at a selected card reader *and* enter a valid Offline PIN at the keypad. This offline option is typically used in medium-to-high security applications.

► To Configure Card and PIN Access in Offline Mode:

1. Follow the instructions on configuring card access in offline mode starting on page C-1.
2. Select the *Offline PIN Required* attribute's check box.
3. Select **4 Digits** (default) or **5 Digits** for the *Offline PIN Digits* attribute.
4. Enter a numeric value for the *Offline PIN Scramble Mode* attribute.

NOTE

Offline PINs are based on the Offline PIN Scramble Mode attribute and must be requested from Technical Support.

5. Select or clear the *Offline PIN After Card* attribute's check box.
 - If you select the check box, the Offline PIN can be entered *after* the badge is presented to the reader.
 - If you clear the check box (default), the Offline PIN must be entered *before* the badge is presented to the reader.

IDENTIFIER FORMATS

This appendix lists and describes all of the official card formats included as part of the P2000 installation.

NOTE

The identifier format descriptions reference the P2000 software's Edit Terminal screen and Card Type tab. For more information, refer to the P2000 AE Software User Manual.

10000 – Default – This format serves as the default format that all Access Control objects in the JCI Standard Templates for doors use. It turns the first 128 bits received from the card reader into a 128-bit card number, and produces no facility code and no issue level. The first received bit becomes the least significant bit of the card number; the 128th received bit becomes the most significant bit of the card number.

12000 – BaFe BCD – This format requires exactly 34 bits to be received, and produces the same card information as the card type *BCD BaFe* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

12010 – BaFe Binary – This format requires exactly 34 bits to be received, and produces the same card information as the card type *Binary BaFe* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

14000 – Cardkey Standard – This format requires exactly 34 bits to be received, and produces the same card information as the card type *Standard Wiegand* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

14001 – Cardkey Standard Reverse – This format requires exactly 34 bits to be received, but in reverse order compared to the *14000 – Cardkey Standard* format. It produces the same card information as the card type *Standard Wiegand* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

14002 – Cardkey Std Rev Duress – This format requires exactly 34 bits to be received, but in reverse order compared to the *14000 – Cardkey Standard* format. It produces the same card information as the card type *Standard Wiegand* (on the Edit Terminal screen's Card Type tab

of the P2000 system configuration), as well as a duress condition. This card format is only suitable for use in Access Control objects.

14010 – Cardkey Encrypted – This format requires exactly 34 bits to be received, and produces the same card information as the card type *Encrypted Wiegand* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

14011 – Cardkey Encrypted Reverse – This format requires exactly 34 bits to be received, but in reverse order compared to the *14010 – Cardkey Encrypted* format. It produces the same card information as the card type *Encrypted Wiegand* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

14012 – Cardkey Enc Rev Duress – This format requires exactly 34 bits to be received, but in reverse order compared to the *14010 – Cardkey Encrypted* format. It produces the same card information as the card type *Encrypted Wiegand* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration), as well as a duress condition. This card format is only suitable for use in Access Control objects.

14020 – Cardkey L46 Magstripe – This format requires exactly 18 characters to be received, and produces the same card information as the card type *Magstripe* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration) when used with a Cardkey L46 reader. This card format is only suitable for use in Access Control objects.

14021 – Cardkey L46 Mag Rev Dur – This format requires exactly 18 characters to be received, but in reverse order compared to the *14020 – Cardkey L46 Magstripe* format. It produces the same card information as the card type *Magstripe* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration) when used with a Cardkey L46 reader, and also produces a duress condition. This card format is only suitable for use in Access Control objects.

14030 – Cardkey Magstripe – This format requires exactly 18 characters to be received, and produces the same card information as the card type *Magstripe* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

14031 – Cardkey Mag Rev Duress – This format requires exactly 18 characters to be received, but in reverse order compared to the *14030 – Cardkey Magstripe* format. It produces the same card information as the card type *Magstripe* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration), and also produces a duress condition. This card format is only suitable for use in Access Control objects.

21000 – Eyecam Prox Indala – This format requires exactly 34 bits to be received, and produces the same card information as the card type *Eyecam, Prox, Indala* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

22000 – FIPS 201 Wiegand 200 – This format produces a 15-digit ID number and a 1-digit Issue Level out of the 200-bit Wiegand signal received by the FIPS 201 Wiegand 200

standard by HID. The Facility Code and HMAC fields are not produced and are therefore 0. The ID Number field is constructed as AAAASSSSXCCCCCC.

AAAA = Agency Code

SSSS = System Code

X = Credential Series

CCCCCC = Credential Number

This 15-digit ID number is compatible with the P2000 Badge Edit Style: FASC-N.

22010 – FIPS 201 Wiegand 75 (with expiration date checking) – This format produces a 15-digit ID number and a 1-digit Issue Level out of the 75-bit Wiegand signal received by the FIPS 201 Wiegand 75 standard by HID. The Facility Code and HMAC fields are not produced and are therefore 0. The ID Number field is constructed as AAAASSSS0CCCCCC.

AAAA = Agency Code

SSSS = System Code

0 = Credential Series (always 0, because it is not contained in reader data)

CCCCCC = Credential Number

This 15-digit ID number is compatible with the P2000 Badge Edit Style: FASC-N.

This format also decodes the expiration date from the received data, and causes access to be denied for expired credentials. This card format is only suitable for use in Access Control objects.

26000 – HID Corporate 1000 – This format requires exactly 37 bits to be received, and produces the same card information as the card type *HID Corporate 1000* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

30000 – JCI Raw 128 Bit – This format turns the first 128 bits received from the card reader into a 128-bit card number, and produces no facility code and no issue level. The first received bit becomes the least significant bit of the card number; the 128th received bit becomes the most significant bit of the card number.

37000 – Motorola 32 Bit – This format requires exactly 32 bits to be received, and produces the same card information as the card type *32 bit Motorola* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

48710 – Sagem Morpho MA520 (with biometric mismatch indication) – This format produces a 20-bit card number and a 12-bit facility code out of a 37-bit signal. The 1st received bit, when equal to 1, indicates a biometric mismatch condition. The 2nd received bit becomes the most significant bit of the card number; the 21st received bit becomes the least significant bit of the card number. The 22nd through the 25th received bits are not used. The 26th received

bit becomes the most significant bit of the facility code; the 37th received bit becomes the least significant bit of the facility code.

49000 – Sensor 26 Bit – This format requires exactly 26 bits to be received, and produces the same card information as if both card types *26 bit Sensor Forward* and *26 bit Sensor Reverse* were selected on the Edit Terminal screen's Card Type tab of the P2000 system configuration. This card format is only suitable for use in Access Control objects.

49001 – Sensor 26 Bit Forward – This format requires exactly 26 bits to be received, and produces the same card information as the card type *26 bit Sensor Forward* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

49010 – Sensor 26 Bit Inverted – This format requires exactly 26 bits to be received, and produces the same card information as if the types *26 bit Sensor Forward*, *26 bit Sensor Reverse* and *Invert Data* were selected on the Edit Terminal screen's Card Type tab of the P2000 system configuration. This card format is only suitable for use in Access Control objects.

49011 – Sensor 26 Bit Inv Forw – This format requires exactly 26 bits to be received, and produces the same card information as if the types *26 bit Sensor Forward* and *Invert Data* were selected on the Edit Terminal screen's Card Type tab of the P2000 system configuration. This card format is only suitable for use in Access Control objects.

49011 – Sensor 26 Bit Inv Rev – This format requires exactly 26 bits to be received, and produces the same card information as if the types *26 bit Sensor Reverse* and *Invert Data* were selected on the Edit Terminal screen's Card Type tab of the P2000 system configuration. This card format is only suitable for use in Access Control objects.

49012 – Sensor 26 Bit Inv Rev Du – This format requires exactly 26 bits to be received, and produces the same card information as if the types *26 bit Sensor Reverse* and *Invert Data* were selected on the Edit Terminal screen's Card Type tab of the P2000 system configuration, as well as a duress condition. This card format is only suitable for use in Access Control objects.

49020 – Sensor 26 Bit Reverse – This format requires exactly 26 bits to be received, and produces the same card information as the card type *26 bit Sensor Reverse* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration). This card format is only suitable for use in Access Control objects.

49021 – Sensor 26 Bit Rev Duress – This format requires exactly 26 bits to be received, and produces the same card information as the card type *26 bit Sensor Reverse* (on the Edit Terminal screen's Card Type tab of the P2000 system configuration), as well as a duress condition. This card format is only suitable for use in Access Control objects.

INDEX

Symbols

? Command 5-31

Numerics

- 10/100Base-T 2-9
- 10000 – Default Format D-1
- 12000 – BaFe BCD Format D-1
- 12010 – BaFe Binary Format D-1
- 14000 – Cardkey Standard Format D-1
- 14001 – Cardkey Standard Reverse Format D-1
- 14002 – Cardkey Std Rev Duress Format D-1
- 14010 – Cardkey Encrypted Format D-2
- 14011 – Cardkey Encrypted Reverse Format D-2
- 14012 – Cardkey Enc Rev Duress Format D-2
- 14020 – Cardkey L46 Magstripe Format D-2
- 14021 – Cardkey L46 Mag Rev Dur Format D-2
- 14030 – Cardkey Magstripe Format D-2
- 14031 – Cardkey Mag Rev Duress Format D-2
- 21000 – Eyecam Prox Indala Format D-2
- 22000 – FIPS 201 Wiegand 200 Format D-2
- 22010 – FIPS 201 Wiegand 75 Format D-3
- 26 bit Sensor Forward selecting for offline mode C-2
- 26 bit Sensor Reverse selecting for offline mode C-2
- 26000 – HID Corporate 1000 Format D-3
- 30000 – JCI Raw 128 Bit Format D-3
- 37000 – Motorola 32 Bit Format D-3
- 48710 – Sagem Morpho MA520 Format D-3
- 49000 – Sensor 26 Bit Format D-4
- 49001 – Sensor 26 Bit Forward Format D-4
- 49010 – Sensor 26 Bit Inverted Format D-4
- 49011 – Sensor 26 Bit Inv Forw Format D-4

49011 – Sensor 26 Bit Inv Rev Format D-4

- 49012 – Sensor 26 Bit Inv Rev Du Format D-4
- 49020 – Sensor 26 Bit Reverse Format D-4
- 49021 – Sensor 26 Bit Rev Duress Format D-4

4x5 Rule 2-9

diagram 2-10

A

Abort Signals 4-30

About

- access badge identifiers 4-10
- access control 4-11
- Access Control objects 3-14
- Anti-Loitering objects 3-18
- Anti-Passback objects 3-16
- Broadcast Management objects 3-43
- Calendar objects 3-42
- central stations 4-30
- CK722 controllers 2-6
- CK722 Device objects 3-8
- Controller Event objects 3-26
- data verification 8-11–8-12
- Door Sequence objects 3-15
- Elevator objects 3-38
- exception schedules 3-41, 8-3
- fast clock feature 8-11
- Folder objects 3-40

identifiers 4-8

Interlock objects 3-24

Intrusion Annunciator objects 3-21

Intrusion Area objects 3-19

intrusion detection 4-24

intrusion detection

systems 3-23

Intrusion Keypad/Display objects

Intrusion Zone objects 3-20

KDM display 2-16

KONE Controller objects 3-31

KONE Elevator objects 3-34

KONE Integration objects 3-27

KONE IP COP objects 3-36

KONE IP DOP objects 3-37

KONE-IP Controller

objects 3-33

KONE-IP Integration

objects 3-29

Multiple Command objects 3-25

objects 3-5–3-6

Occupancy objects 3-17

Otis Controller objects 3-32

Otis Elevator objects 3-35

Otis Integration objects 3-28

P2000 host 2-3

P2000 SCT 2-5

P2000 SMS components 2-1

P2000 workstations 2-3

PIN code identifiers 4-11

RFID tags 4-11

S300 Hardware Module objects 3-10

S300 Reader Terminal objects 3-11

S300 Trunk objects 3-9

Schedule objects 3-41

scheduled items 8-5

scheduling 8-1

Security Binary Output objects 3-13

Security Notification Class objects 3-44

Security Supervised Input objects 3-12

Site objects 3-39

templates 6-1

time/value pairs 8-5

weekly schedules 8-3

wild cards 8-7

Access

alternate 7-47

denying in offline mode C-1

Access Badge Identifiers 4-5, 4-12

described 4-10

invoking access with keypad B-1–B-2

invoking controller events with keypad B-5–B-6

invoking timed overrides with keypad B-2, B-4

using with PIN codes 4-11

Access Control

alarm monitoring 4-18

anti-loitering feature 4-19

anti-passback feature 4-19

application examples 7-4

controller events 4-23

database partitioning 4-24

description 4-11

- factory templates 6-1
 features and applications 4-15
 guard tour 4-23
 host computer 4-12
 identifiers 4-8
 input devices 4-22
 input/output devices 4-22
 integration with intrusion 7-57
 man-traps feature 4-21
 mustering feature 4-21
 occupancy feature 4-21
 output devices 4-22
 panels 4-13
 scheduling feature 4-18
 supervisory controllers 4-13
 system events 4-23
 video imaging feature 4-17
 visitor management
 feature 4-18
- Access Control Objects** 3-3
 configuring for UL 1076
 compliance A-2
 described 3-14
 functions with Occupancy
 objects 3-17
 identifier format for 6-2
 linking objects to 3-2
- Access Decisions** 3-14
- Access Points** 1-4, 4-11, 4-13
 in man-traps 4-21
- Access Privileges** 4-18
- Access Profiles** 4-18
- Access Requests**
 invoking with keypad B-1
- Accessing**
 command mode with
 KDM 2-19-2-20
 list of annunciators with
 KDM 2-15
 list of areas with KDM 2-15
 list of zones with KDM 2-15
- ACK Command** 2-15
- Acknowledging**
 intrusion alarms 2-15, 2-24–
 2-25
- ACO**
 See *Access Control Objects*
- Activate All Allowed Floors** 7-56
- Activate Only Selected Floor** 7-56
- Activating**
 intrusion zones 2-15, 2-23–
 2-24
- Active Alarms/Bypass**
 Screen 2-17–2-18
- Active Date Range** 8-11
- ACTV Command** 2-15
- ADA**
 configuring a door opening
 device 5-17
 door application 7-46
 relay 7-47
- Adapting Templates** 7-3
- Adding**
 exception schedules 8-16
- scheduled events 8-14
 scheduled items 8-5, 8-18
- Adding Objects for Application**
 anti-loitering 7-25
 anti-passback entry/exit 7-22
 asset protection 7-44
 emergency exit portal 7-9
 intrusion detection 7-58
 intrusion with supervised
 inputs 7-65
 keylock arming/disarming 7-83
 low level elevator 7-52
 multiple intrusion areas 7-73
 occupancy with turnstiles 7-38
 parking lot occupancy 7-27
 reader with tamper switch 7-15
 single portal with two
 readers 7-6
 timed anti-passback 7-20
- Address Conflicts**
 avoiding 5-29
- Air Crew Requests** B-1
- Alarm Inputs** 7-62
- Alarm Messages**
 receiving 5-30
- Alarm Monitoring** 4-18
- Alarm Signals** 4-30
 unsupressing life safety A-1
- Alarms**
 acknowledging 2-15, 2-24–
 2-25
- All Alarm and Status Mode** 2-17
- All Alarm and Status**
 Screens 2-17
- All Alarms Only Mode** 2-18
- Alternate Access Timing** 7-47
- Americans with Disabilities Act**
 See *ADA*
- ANNU Command** 2-15
- Annunciation** 4-24, 7-62, 7-72
- Annunciators**
 accessing list of 2-15
 silencing 2-15, 2-27–2-28, 3-21
- Annunciators Screen** 2-17–2-18
- Anonymous Mode**
 occupancy 3-17, 4-21
- Anti-Loitering** 1-1, 4-19
 application 7-24
- Anti-Loitering Objects** 3-3
 described 3-18
- Anti-Passback** 1-1, 4-19
 entry/exit rule 4-20
 hard entry/exit rule 7-22
 soft entry/exit rule 7-22
 time rule 4-19–4-20
- Anti-Passback Objects** 3-3
 described 3-16
- Application (Priority)** 5-26
- Application Descriptions** 7-26
 asset protection 7-43
 emergency exit portal 7-8
 occupancy with turnstiles 7-36
 parking lot occupancy 7-26
 portal entry
- reader/keypad 7-7
 single reader 7-4
 two readers 7-5
- portal entry/exit 7-8
 reader with tamper switch 7-14
- Application Examples**
 access control 7-4
 intrusion detection 7-56
- Application Notes**
 asset protection 7-46
 emergency exit portal 7-13
 low level elevator 7-56
 multiple zones with one
 area 7-62
 occupancy with turnstiles 7-42
 parking lot occupancy 7-36
 portal entry
 two readers 7-7
 portal entry with reader/
 keypad 7-7
 reader with tamper switch 7-19
- Application Templates**
 anti-loitering 7-24
 anti-passback entry/exit 7-22
 asset protection 7-44
 assisted access 7-48
 emergency exit portal 7-9
 intrusion with supervised
 inputs 7-65
 keylock arming/disarming 7-82
 low level elevator 7-51
 multiple intrusion areas 7-72
 multiple zones with one
 area 7-57
 occupancy with turnstiles 7-37
 parking lot occupancy 7-27
 portal entry
 single reader 7-4, 7-7
 two readers 7-5
 portal entry/exit 7-8
 reader with tamper switch 7-14
 timed anti-passback 7-20
- Applications**
 access control 4-15
 anti-loitering 4-19, 7-24
 anti-passback 4-19
 assisted access 7-46
 CK722 4-1
 controller events 4-23
 database partitioning 4-24
 developing 3-5
 emergency exit portal 7-8
 entry/exit anti-passback 7-21
 guard tour 4-23
 guard tour examples 4-24
 intrusion with supervised
 inputs 7-62
 low level elevator
 interface 7-49
 man-traps 4-21
 motion detection 3-4
 multiple zones with one
 area 7-57

- mustering 4-21
 occupancy 4-21
 occupancy with turnstiles 7-36
 P2000 SMS 4-1
 parking lot occupancy 7-26
 portal entry 4-15
 reader/keypad 7-7
 single reader 7-4
 portal entry/exit 4-16, 7-8
 reader with tamper switch 7-14
 security C-1
 timed anti-passback 7-19
 using system events 4-23
 visitor management 4-18
Archive Database
 defined 1-3
 downloading 5-8–5-10
Archiveable
 attribute defined as 5-19
AREA Command 2-15
Areas
 accessing list of 2-15
 anti-loitering 3-18, 4-19
 anti-passback 4-19
 arming 2-20–2-21
 disarming 2-21–2-22
 intrusion 4-28
 special handling when
 arming 2-21
Areas Screen 2-17
ARM Command 2-15
ARM Key 2-15
Armed State 4-31
Arming
 intrusion areas 2-15, 2-20–2-21
 intrusion zones 4-28
 special handling 2-21
 with keylock application 7-82
ASHRAE 1-3
Assets 1-3, 4-1
 assigning ownership to 4-3–4-4
 categories of 4-2
 described 4-2
 groups of 4-3
 tracking 7-43
 tracking with RFID tags 4-11
 using identifiers with 4-8
Assigning
 CK722 static IP address 5-7,
 5-33
 entity escorts 4-3, 4-5
 entity ownership 4-3–4-4
 entity sponsors 4-3
 schedule exception priority
 levels 8-4
 unused field points 5-19
Assisted Access 7-46
Assumptions
 JCI_RDR2S_Card-In
 template 6-27
 JCI_RDR2S_Card-In-Card-Out
 template 6-33
Asynchronous Operation 5-20–
 5-21
- Attributes** 1-4
 common device 3-5
 common to all objects 3-5
 configurable 3-5
 editing calendar 8-22
 editing schedule 8-22
 examples of prioritizing 5-25
 for job-specific templates 7-3
 prioritizing 5-24
 read only 3-5
 selecting package 7-3
 time zone 3-39, 3-41
Auto Life Safety (Priority) 5-26
Avoiding Address Conflicts 5-29
- B**
- B.M.S. Protocol** 3-28
BACnet 1-1, 1-4, 3-5
 defined 1-3
BACnet Broadcast Management Devices
 See *BBMD*
Badge Design 4-17
Badge Readers 4-13
Badges
 access 4-10
 identification 4-9–4-10
BaFe BCD Format D-1
BaFe Binary Format D-1
Barcodes 4-9
Barium Ferrite Cards C-2
Barrier Gate Operators 7-26
Basic Components
 intrusion detection
 systems 4-25
Basic H/W Module
 Templates 6-21–6-23
Battery Low Input 5-11
Baud Rate
 detection with RDR2S 2-12
 detection with RDR2S-A 2-13
BBMD 3-43
Best Practices 5-15–5-20
Bi-directional_Man_Trap
 Template 6-63
Binary BaFe
 selecting for offline mode C-2
Biometrics 4-12
Bond Sensors
 best use of 5-18
Boolean Data Type 5-19
Broadcast Management Objects
 described 3-43
Browsing
 All Alarm and Status
 screens 2-17
Burglar Alarm Systems
 See *Intrusion Detection Systems*
Buses
 RS-485 3-45
Buttons
 Calendar object 8-3
Bypassed State 4-31
Bypassing
- intrusion zones 2-15, 2-22–
 2-23
BYPS Command 2-15
- C**
- Cables**
 RS-232 null modem 5-32
Cabling
 network 2-9
Calendar Entries 8-6
 deleting 8-21
 wild cards 8-7
Calendar Objects
 button descriptions 8-3
 described 3-42, 8-2
 interaction with Schedule
 objects 3-41
 uses of 8-2
Calendars
 copying 8-22
 creating 8-12
 creating entries 8-20
 deleting 8-23
 displaying 8-13
 editing attributes of 8-22
 editing entries 8-21
 managing 8-12
 pasting 8-22
 toggling between views 8-22
Calibrating Inputs 5-14
Cancel Signals 4-30
Card Formats D-1
 using catch-all type 5-18
Card ID Identifiers
 invoking access with
 keypad B-1
 invoking controller events with
 keypad B-5–B-6
 invoking timed overrides with
 keypad B-3
Card ID Requests B-1
Card Readers 4-13
Card Types
 selecting for offline mode C-2
Card-Activated Events 4-23
Cardholders
 See *Entities*
Card-In Door Application
 with one reader and REX 7-4
Card-In-Card-Out (CICO) 6-3
Card-In-Card-Out Doors
 setting up 5-17
Cardkey Enc Rev Duress
 Format D-2
Cardkey Encrypted Format D-2
Cardkey Encrypted Reverse
 Format D-2
Cardkey L46 Mag Rev Dur
 Format D-2
Cardkey L46 Magstripe
 Format D-2
Cardkey Mag Rev Duress
 Format D-2
Cardkey Magstripe Format D-2

- Cardkey Standard Format D-1
 Cardkey Standard Reverse Format D-1
 Cardkey Std Rev Duress Format D-1
 Cards
 See *Identifiers*
 Catch-All Card Formats 5-18
 CCTV 4-23
 Central Stations
 about 4-30
 signaling 4-30
 Change of Value Messages 8-12
 Changing
 Metasys password 5-31
 Chapter Descriptions 1-2
 CICO
 See *Card-In-Card-Out Doors*
 CK722 Controllers 1-4-2-1
 applications 4-1
 architecture 3-1
 assigning static IP address to 5-7, 5-33
 commissioning 5-1
 communication settings 2-11
 defined 1-3
 description of 2-6
 determining IP address of 5-31
 determining version of 5-31
 disconnected from S300 module C-1
 downloading archive database to 5-8-5-10
 hardware components 2-7
 installation verification 5-7
 installing 5-1
 large enclosure for 2-35
 major components of 2-7
 manually setting IP address of 5-31
 mounting 2-6
 network communication 2-9
 pinging on network 5-31
 receiving SNMP traps 5-30
 restoring functionality 5-29
 small enclosure for 2-36
 system configuration 2-8
 upgrading operating system of 5-29
 verifying online status 5-10
 writing flash memory 5-34
 CK722 Device Objects 1-4
 described 3-8
 CK722 Requirements 5-5
 CLI
 See *Command Line Interface (CLI)*
 Closed Circuit Television
 See *CCTV*
 Cluster 4-12
 Codes
 PIN 4-11
 Command Line Interface (CLI)
 using 5-31
- Command Mode 2-16
 accessing with KDM 2-19-2-20
 Commands
 keypad/display module 2-15, 2-19
 Commissioning
 CK722 Controllers 5-1-5-30
 Common Attributes 3-5
 Common Device Attributes 3-5
 Common PIN Identifiers
 invoking access with keypad B-2
 Common PIN Requests B-1
 Common Security Applications 7-4
 Communication Settings
 CK722 controllers 2-11
 Communication Systems
 intrusion detection 4-30
 Compliance for UL 1076 A-1
 Components
 of ID badge identifiers 4-9
 Configurable Attributes 3-5
 Configuration
 P2000 SMS diagram 2-34
 typical video imaging 4-17
 verification (CK722) 5-7
 Configuring
 card access in offline mode C-1
 offline mode options C-1
 offline PINs C-3
 Connectors
 CK722 controller 2-7
 Contacts
 door 4-13
 Controller Event Objects
 configuring for UL 1076 compliance A-3
 creating apps with 5-20, 5-23
 described 3-26
 Controller Events
 described 4-23
 invoking with keypad B-5-B-6
 unsupressing life safety alarms A-1
 using keypad readers B-1
 Controllers
 description of intrusion 4-27
 legacy 1-1
 peer-to-peer communication 5-3
 Controlling
 doors 3-15
 objects with Interlock 3-24
 objects with Multiple Command 3-25
 Copying
 calendars 8-22
 existing templates 7-2
 scheduled events 8-15
 schedules 8-22
 templates 6-3, 7-1
 COV Messages
 See *Change of Value Messages*
 Crash Bars 4-15
 use with emergency exits 7-8
 See also *Panic Bars*
 Creating
 badge identifiers 4-10
 calendar entries 8-20
 calendars 8-12
 custom logic application 5-20-5-23
 naming convention 5-15
 P2000 user accounts 4-8
 package for door 5-12-5-13
 schedules 8-12
 Credentials
 See *Identifiers*
 Critical Equipment (Priority) 5-26
 Cursor Test 5-14
 Custom Card Formats 5-3
 Custom Logic Application
 creating 5-20-5-23
 Customizing
 templates 6-3
- D**
- Data Types 5-19
 applying to scheduled items 8-5
 available with UDFs 4-8
 Data Verification 8-11-8-12
 Database
 writing flash memory to 5-34
 Databases
 archive 1-3
 downloading 5-8-5-10
 P2000 2-3
 partitioning 4-24
 Date Formats 8-6
 scheduling 8-6
 Date Range
 exceptions 8-4
 format 8-6
 using with wild cards 8-9
 Date Type Exceptions 8-4
 Dates
 formats with scheduling 8-6
 wild cards 8-8
 Default Identifier Format D-1
 Defining
 schedule exceptions 8-2
 weekly schedule exceptions 8-6
 Definitions 1-3
 Delay Announcer Output Attribute 7-62
 Deleting
 calendar entries 8-21
 calendars 8-23
 scheduled events 8-15
 schedules 8-23
 Demand Limiting (Priority) 5-27
 Detectors 4-24
 dual technology 4-26
 glass break 4-26
 infrared beam interruption 4-26
 intrusion detection 4-25

- microwave 4-26
 motion 4-14
 movement 4-26
 paired 4-27
 passive infrared 4-26
 pressure mats 4-26
 resetting with KDM 2-15
 stopping test of 2-15
 testing 2-15
 ultrasonic 4-26
 vibration 4-25
- D**
Devices
 input/output 4-22
DIN Enclosures 2-8, 2-34
 large 2-35
 small 2-36
Disabling
 schedules 8-22
Disarmed State 4-31
Disarming
 intrusion areas 2-15, 2-21–2-22
 intrusion zones 4-28
 with keylock application 7-82
Disaster Recovery 4-12
Displaying
 calendars 8-13
 next KDM item 2-15
 previous KDM item 2-15
 scheduled events 8-13
 schedules 8-13
Door Contacts 4-13, 7-46
 using with inactive doors 5-17
Door Control 3-15
Door H/W Module
 Templates 6-24–6-34
Door Hardware 4-13
Door Locks
 magnetic 4-14
Door Opening Devices
 best use of 5-17
Door Requirements 5-12
Door Sensors
 intrusion detection 4-25
Door Sequence Objects
 described 3-15
Door x-Templates 6-34
Doors
 CICO 5-17, 6-3
 creating package for 5-12–5-13
 emergency 4-15
 fail safe/secure 4-18
 without contact or REX 5-16
Double Knock
 application 7-70
 detection 4-27
Downloading
 archive database 5-8–5-10
 rules 5-8
DSO
See Door Sequence Objects
DSRM Command 2-15
Dual Technology Detectors 4-26
Dynamic Host Configuration Protocol (DHCP) 5-3
- E**
E.M.S Protocol 3-28
EAC
See Electronic Access Control
Editing
 calendar attributes 8-22
 calendar entries 8-21
 exception schedules 8-17
 schedule attributes 8-22
 schedule effective period 8-19
 scheduled events 8-15
 scheduled items 8-18
Effective Period 8-11
Egress Devices 4-14
 REX application 7-4
Eight_Door_Occupancy
 Template 6-63
Eight_Zone_Area_Keypad_Annun
 Template 6-63
Eight_Zone_Area_No_Keypad
 Template 6-64
Electric Locks 4-14
Electric Strikes 4-14
Electronic Access Control 4-11
 systems 4-12
Elevator Objects
 described 3-38
 KONE Controller 3-31
 KONE Integration 3-27
 KONE-IP Controller 3-33
 KONE-IP Integration 3-29
 Otis 3-35
 Otis Controller 3-32
 Otis Integration 3-28
Elevators
 low level interface
 application 7-49
Emergency
 mustering 4-21
Emergency Exits 4-15
 configuring application for 7-8
 described 7-8
Enclosures
 S300-DIN-L 2-35
 S300-DIN-S 2-36
 S300-XL 2-37
 S300-XS 2-38
 S300-XXS 2-39
 types of 2-34
Encrypted Wiegand
 selecting for offline mode C-2
Entities
 assigning ownership to 4-3–4-4
 categories of 4-2
 defined 1-3
 described 4-1
 escorting 4-3, 4-5
 groups of 4-3
 journals 4-8
 sponsoring 4-3–4-4
 types of 4-1
 validation of 4-7
 viewing status of 4-7
Entity Groups 4-3
- assigning as sponsors/
 owners 4-4
 escorting 4-5–4-6
Entity Track Mode
 occupancy 4-21
Entry Time Attribute 7-62
Entry/Exit
 anti-passback application 7-21
 anti-passback rule 3-16, 4-20
 portal application 4-16
Enumeration Data Type 5-19
Escorting
 entities 4-3
 entity groups 4-5–4-6
Escorts 4-5
Ethernet 2-9
Evacuation
 mustering 4-21
Event Notification Messages 3-44
Events
 displaying scheduled 8-13
 system 4-23
 time/value pairs 8-5
 triggering controller 3-26
 unsupressing life safety
 alarms A-1
Examples
 adding scheduled items 8-5
 adding two RTOs 3-2
 linking objects 3-1
 motion detection 3-4
Exception Schedules 3-42, 4-18,
 8-1
 adding 8-16
 date formats 8-6
 defining 8-2, 8-6
 described 3-41, 8-3
 editing 8-17
 removing 8-18
 week and day format 8-7
 wild cards 8-7
 with time/value pairs 8-5
Exception Types 8-4
Executive Privileges 7-44
EXIT Command 2-15
EXIT Key 2-17
Exiting KDM Interface 2-15
Expansion Enclosures 2-8, 2-34
 S300-XL 2-37
 S300-XS 2-38
 S300-XXS 2-39
Eyecam Prox
 selecting for offline mode C-2
Eyecam Prox Indala Format D-2
- F**
F1 Key
 on KDM 2-15
Facility Codes 3-5, C-1
Fail Safe/Unlocked 4-14
 described 4-18
Fail Secure/Locked 4-14
 described 4-18
Fast Clock

- described 8-11
F
 Field Device Level
 P2000 SMS 2-1
 Field Devices 2-1, 4-13
 communication with 2-9
 defined 1-3
 description of 2-11
 I16 modules 2-30
 IO8 modules 2-31
 RDR2 modules 2-29
 SI8 modules 2-33
 SIO8 modules 2-32
 supported on S300 bus 2-11
 Field Points
 assigning unused 5-19
 defined 1-3
 stealing 5-19
 Fingerprints 4-9, 4-17
 FIPS 201 Wiegand 200
 Format D-2
 FIPS 201 Wiegand 75 Format D-3
 Firmware Version
 viewing CK722 5-31
 Flash Memory
 writing to 5-34
 Float Data Type 5-19
 Floor List Attribute
 configuring 7-54
 Floors
 elevator application 7-50
 Folder Objects
 described 3-40
 Formats
 card/identifier D-1
 catch-all card type 5-18
 date 8-6
 Free Exit 4-14–4-15
 Full-IO H/W Module
 Templates 6-4–6-20
 Functional Keys
 on KDM 2-15
G
 Gate Operators
 types of 7-26
 Glass Break Detectors 4-26
 Global Events 4-23
 Graphical Representations
JCI_I16_Full-IO template 6-6
JCI_IO8_Full-IO template 6-8
JCI_KDM_with-ACO
 template 6-62
JCI_RDR2S_Card-In
 template 6-25
JCI_RDR2S_Card-In-Card-Out
 template 6-29
JCI_RDR2SA_Basic
 template 6-22
JCI_RDR2SA_Full-IO
 template 6-16
JCI_RDR8S_Basic
 template 6-23
JCI_RDR8S_Full-IO
 template 6-18
J
JCI_SIO8_Full-IO
 template 6-13
JCI_x_Card-In template 6-40
JCI_x_Card-In_IAD_TAMP
 template 6-44
JCI_x_CICO template 6-48
JCI_x_CICO_IAD_TAMP
 template 6-53
JCI_x_Contact template 6-35
JCI_x_Contact-w-Alarm
 template 6-37
JCI_x_Elevator template 6-59
S300_SI8 template 6-10
G
 Ground Wiring 3-46
 Guard Tour
 described 4-23
 examples 4-24
 Guidelines
 numbering h/w modules 5-29
H
 Hard Entry/Exit Rule 7-22
 Hardware
 door 4-13
 intrusion detection
 systems 4-31
 Hardware Module Objects
 See *S300 Hardware Module Objects*
 Hardware Module
 Requirements 5-11
 Hardware Module Templates 6-1
 Hardware Modules
 creating manually 5-12
 creating with template 5-11
 numbering guidelines 5-29
 Heavy Equipment Delay
 (Priority) 5-26
help Command 5-31
 HID Corporate 1000 Format D-3
 selecting for offline mode C-2
 High Availability 4-12
 Holiday Schedules
 See *Exception Schedules*
 Host Computers
 access control 4-12
 P2000 2-3
 Host Level
 P2000 SMS 2-1
 Host Priority 5-28
I
 I16 Modules 2-30
 Identification Badges 4-9–4-10
 Identifier Activity 4-7
 Identifier Formats 5-3, D-1
 ACO 6-2
 Identifiers 4-5, 4-11–4-12
 access badge 4-10–4-11
 defined 1-3
 described 4-8
 identification badges 4-9–4-10
 PIN 4-11
 types of 4-8–4-9
I
 Importing
 templates 7-1
 Inactive Date Range 8-11
 Inactive Doors
 independent reporting of 5-17
 using contacts with 5-17
 Inactive Mode 2-16
 Infrared Beam Interruption
 Detectors 4-26
 Input Devices 4-22
 intrusion 4-24
 Input Objects
 See *Security Supervised Input Objects*
 Input/Output 1-3, 4-22
 Inputs
 battery low 5-11
 calibrating 5-14
 panel tamper 5-11
 power fail 5-11
 supervised 7-62
 Installation
 verification (CK722) 5-7
 Installing
 CK722 Controllers 5-1
 Instantiating
 intrusion application
 template 7-76–7-78
 Integer Data Type 5-19
 Integration Objects
 KONE 3-27
 KONE-IP 3-29
 Otis 3-28
 S300 Trunk 3-9
 Interlock Objects
 adding to emergency exit
 application 7-9, 7-11–7-12
 adding to keylock
 application 7-83–7-86
 adding to occupancy with
 turnstiles
 application 7-38–7-41
 adding to parking lot occupancy
 application 7-28–7-35
 adding to reader with tamper
 switch application 7-15,
 7-17
 creating apps with 5-20, 5-23
 described 3-24
 Internet Protocol (IP) 5-3
 Introduction
 to manual 1-1
 Intrusion Alarms
 acknowledging 2-15, 2-24–
 2-25
 Intrusion Announcer Objects
 described 3-21
 interaction with KDOs 3-22
 Intrusion Announciators
 accessing list of 2-15
 silencing 2-15, 2-27–2-28
 Intrusion Applications
 double knock 7-70
 keylock arming/disarming 7-82

- lighted display to signal arming/
disarming 7-81
- multiple areas with
unsupervised inputs 7-71
- multiple zones with one
area 7-57
- paired sensors 7-70
- supervised inputs 7-62
- Intrusion Area Objects 3-4, 3-19
interaction with Intrusion Zone
objects 3-20
- interaction with KDOs 3-22
- Intrusion Areas 4-28
accessing list of 2-15
adding to intrusion
application 7-78-7-80
- application with multiple 7-71
- arming 2-15, 2-20-2-21
- controlling across multiple
CK722 controllers 7-72
- disarming 2-15, 2-21-2-22
- special handling 2-21
- Intrusion Detection
application examples 7-56
- areas 4-28
- communication systems 4-30
- double knock 4-27
- factory templates 6-1
- integration with access
control 7-57
- introduction to 4-24
- paired sensors 4-27
- perimeters 4-25
- reporting 4-27
- role of system 4-24
- sensors/detectors 4-25
- signals 4-30
- zones 4-28
- Intrusion Detection Systems
basic components 4-25
- controllers 4-27
- described 3-23
- hardware 4-31
- keypad description 4-27
- non-supervised 4-31
- supervised 4-31
- three states 4-31
- Intrusion Keypad/Display Objects
described 3-22
- Intrusion Sensors
resetting 2-15
- Intrusion Signals 4-30
- Intrusion Zone Objects 3-4, 3-20
configuring for UL 1076
compliance A-3
- interaction with Intrusion Area
objects 3-19
- interaction with KDOs 3-22
- Intrusion Zones 4-28
accessing list of 2-15
activating 2-15, 2-23-2-24
- bypassing 2-15, 2-22-2-23
- resetting sensors for 2-25-2-26
- testing sensors for 2-26-2-27
- IO8 Modules 2-31
- IP Address
assigning static 5-7, 5-33
- manually setting 5-31
- viewing CK722 5-31
- Issuing Prioritized Writes 5-24
- Items
adding to schedules 8-5
- referencing with Schedule
objects 8-5
- J**
- JCI Legacy Templates 6-63-6-65
- JCI Raw 128 Bit Format D-3
- JCI Standard Templates 6-1
categories of 6-4
- naming convention of 6-2
- JCI_I16_Full-IO Template 6-5
graphical representation 6-6
- object diagram 6-5
- object list and description 6-6
- JCI_IO8_Full-IO Template 6-7
graphical representation 6-8
- non-default attributes 6-9
- object diagram 6-7
- object list and description 6-8
- JCI_KDM_with-ACO
Template 6-61
- graphical representation 6-62
- non-default attributes 6-63
- object diagram 6-61
- object list and description 6-62
- JCI_RDR2S_Card-In
Template 6-24
- assumptions 6-27
- graphical representation 6-25
- non-default attributes 6-26
- object diagram 6-24
- object list and description 6-25
- JCI_RDR2S_Card-In-Card-Out
Template 6-27
- assumptions 6-33
- graphical representation 6-29
- non-default attributes 6-32
- object diagram 6-29
- object list and description 6-30
- JCI_RDR2SA_Basic
Template 6-21
- graphical representation 6-22
- non-default attributes 6-23
- object diagram 6-21
- object list and description 6-22
- JCI_RDR2SA_Card-In
Template 6-33
- JCI_RDR2SA_Card-In-Card-Out
Template 6-33
- JCI_RDR2SA_Full-IO
Template 6-14
- graphical representation 6-16
- non-default attributes 6-17
- object diagram 6-15
- object list and description 6-17
- JCI_RDR8S_Basic Template 6-23
- graphical representation 6-23
- non-default attributes 6-23
- object diagram 6-23
- object list and description 6-23
- JCI_RDR8S_Full-IO
Template 6-18
- graphical representation 6-18
- non-default attributes 6-20
- object diagram 6-18
- object list and description 6-18
- JCI_SI8_Full-IO Template 6-9
non-default attributes 6-11
- object diagram 6-10
- object list and description 6-11
- JCI_SIO8_Full-IO Template 6-12
graphical representation 6-13
- object diagram 6-12
- object list and description 6-13
- JCI_SIO8_Full-IO template
non-default attributes 6-14
- JCI_x_Card-In Template 6-40
graphical representation 6-40
- non-default attributes 6-42
- object diagram 6-40
- object list and description 6-41
- JCI_x_Card-In_IAD
Template 6-42
- JCI_x_Card-In_IAD_TAMP
Template 6-43
- graphical representation 6-44
- non-default attributes 6-46
- object diagram 6-43
- object list and description 6-44
- JCI_x_Card-In_TAMP
Template 6-42
- JCI_x_CICO Template 6-47
graphical representation 6-48
- non-default attributes 6-51
- object diagram 6-48
- object list and description 6-49
- JCI_x_CICO_IAD Template 6-52
- JCI_x_CICO_IAD_TAMP
Template 6-52
- graphical representation 6-53
- non-default attributes 6-57
- object diagram 6-53
- object list and description 6-54
- JCI_x_CICO_TAMP
Template 6-51
- JCI_x_Contact Template 6-34
graphical representation 6-35
- non-default attributes 6-36
- object diagram 6-34
- object list and description 6-35
- JCI_x_Contact-w-Alarm
Template 6-36
- graphical representation 6-37
- non-default attributes 6-39
- object diagram 6-37
- object list and description 6-38
- JCI_x_Elevator Template 6-58
graphical representation 6-59
- non-default attributes 6-60
- object diagram 6-58
- object list and description 6-59

- Job-Specific Templates 5-15, 7-1
adapting for 7-3
attributes for 7-3
creating 7-1–7-3
defined 7-1
testing 5-16
Journals 4-8
- K**
- KDM
See Keypad/Display Module
- KDO
See Intrusion Keypad/Display Objects
- Key Items 8-5
- Key Override Switches 4-15
- Key Terms 1-3
- Keylock Arming/Disarming 7-82
- Keypad Readers
invoking access requests with B-1
invoking common PIN requests with B-2
invoking controller events with B-5
invoking timed overrides with B-2
quick guide B-8–B-9
using B-1
- Keypad Timeout B-1
- Keypad/Display Module 4-27
about the display 2-16
accessing command mode 2-19–2-20
acknowledging intrusion alarms 2-24–2-25
activating zones 2-23–2-24
arming areas 2-20–2-21
bypassing intrusion zones 2-22–2-23
commands 2-15, 2-19
description 4-27
disarming areas 2-21–2-22
keys 2-15
modes 2-16
resetting zone sensors with 2-25–2-26
silencing annunciators with 2-27–2-28
testing sensors with 2-26–2-27
- Keypad/Display Objects
See Intrusion Keypad/Display Objects
- Keys
on KDM 2-15
- KONE Controller Objects described 3-31
- KONE Elevator Objects described 3-34
- KONE Integration Objects as parent object 3-31 described 3-27
- KONE IP COP Objects described 3-36
- KONE IP DOP Objects described 3-37
- KONE-IP Controller Objects described 3-33
- L**
- LAN
See Local Area Network
- Large Enclosures S300-DIN-L 2-35
S300-XL 2-37
- LED Indicators CK722 controllers 2-7
SI8 module 2-33
SIO8 module 2-32
- Legacy Controllers 1-1, 2-3
defined 1-3
- Legacy Templates 6-63–6-65
- Life Safety
unsupressing alarm signals A-1
- Linking
objects 3-1–3-2
- Load Rolling (Priority) 5-27
- Load Wizard 5-8
- Loading
intrusion application template 7-76–7-78
- Local Area Network 1-4, 2-3, 2-5, 4-30
- P2000 SMS
recommendations 2-9
- Local Events 4-23
- Local Feedback 6-3
- Lockdown Mode 7-43, 8-6
- Locks
electric 4-14
magnetic 4-14
- Long-Range Readers 7-43
- Loops 7-27–7-28, 7-36
- Low Level Elevator Interface 7-49
- M**
- Magnetic Locks 4-14
- Magnetic Stripe Cards 4-10
selecting for offline mode C-2
- Managing
calendars 8-12
schedules 8-12
visitors 4-18
- Man-traps 4-21
- Manual Conventions 1-2
- Manual Life Safety (Priority) 5-26
- Manually-operated Devices 4-27
- Mapping Points 5-16
- MCE
See Metasys Control Engine
- Messages
COV 8-12
event notification 3-44
- Metasys Control Engine 1-3, 3-5, 3-31–3-32
defined 1-4
- Metasys Password changing 5-31
- Microwave Detectors 4-26
- Minimum On Off (Priority) 5-26
- Miscellaneous Templates 6-58–6-63
- Modes
All Alarm and Status 2-17
All Alarms Only 2-18
KDM 2-16
Select Alarm and Status 2-18
Select Alarms Only 2-18
- Modes of Operation RDR2S-A and RDR2S 6-24
- Modifying
an original template 7-1
calendar attributes 8-22
calendar entries 8-21
exception schedules 8-17
objects for assisted access application 7-48
objects in single portal application 7-6
schedule attributes 8-22
schedule effective period 8-19
scheduled items 8-18
templates 6-3, 7-1
- Modifying Objects for Application
anti-loitering 7-25
anti-passback entry/exit 7-23
asset protection 7-45
intrusion with supervised inputs 7-67
low level elevator 7-54
multiple intrusion areas 7-74
parking lot occupancy 7-35
timed anti-passback 7-21
- Momentary Alarms 7-70
- Monitoring
alarms 4-18
guard activities 4-23
number of occupants 4-21
- Monitoring Facilities 4-30
- MORE Command 2-15
- Motion Detection Application 3-4
- Motion Detectors 4-14
- Motorola 32 Bit Format D-3
- Mounting
CK722 controllers 2-6
RDR2S modules 2-12
RDR2S-A modules 2-13
- Movement Detectors 4-26
- Multiple Command Objects
creating apps with 5-20, 5-23
described 3-25
- Multiple Priorities
releasing 5-28
- Muster Readers 4-7
- Mustering 4-7
application 4-21
- N**
- Naming Convention 5-2, 5-15
JCI standard templates 6-2
- Naming Objects 5-15

- Network
 cabling 2-9
 communication 2-9
 configuration 2-8
 Network Controllers 2-3, 2-6
 CK722 1-3
 Network Devices
 max distance between 2-9
 Network Utility Tool 5-7
 described 5-29
 Networking Guidelines
 10/100Base-T 2-9
 NEXT Command 2-15
 Non-Default Attributes
 JCI_IO8_Full-IO template 6-9
 JCI_KDM_with-ACO
 template 6-63
 JCI_RDR2S_Card-In
 template 6-26
 JCI_RDR2S_Card-In-Card-Out
 template 6-32
 JCI_RDR2SA_Basic
 template 6-23
 JCI_RDR2SA_Full-IO
 template 6-17
 JCI_RDR8S_Basic
 template 6-23
 JCI_RDR8S_Full-IO
 template 6-20
 JCI_SI8_Full-IO template 6-11
 JCI_SIO8_Full-IO
 template 6-14
 JCI_x_Card-In template 6-42
 JCI_x_Card-In_IAD_TAMP
 template 6-46
 JCI_x_CICO template 6-51
 JCI_x_CICO_IAD_TAMP
 template 6-57
 JCI_x_Contact template 6-36
 JCI_x_Contact-w-Alarm
 template 6-39
 JCI_x_Elevator template 6-60
 Non-supervised Intrusion
 Detection Systems 4-31
 Normal Access Mode 8-6
 Normally Closed Switches 2-12,
 2-29, 4-13
 Normally Open Switches 2-12,
 2-29, 4-13
 Notes
 asset protection
 application 7-46
 emergency exit
 application 7-13
 entity 4-8
 intrusion application 7-62
 low level elevator
 application 7-56
 occupancy with turnstiles
 application 7-42
 parking lot occupancy
 application 7-36
 reader with tamper switch
 application 7-19
 single portal with reader/keypad
 application 7-7
 single portal with two readers
 application 7-7
 Numerical Keys
 on KDM 2-15
 NUT
 See *Network Utility Tool*
- O**
- Object Attributes 1-4
 Object Diagrams
 JCI_I16_Full-IO template 6-5
 JCI_IO8_Full-IO template 6-7
 JCI_KDM_with-ACO
 template 6-61
 JCI_RDR2S_Card-In
 template 6-24
 JCI_RDR2S_Card-In-Card-Out
 template 6-29
 JCI_RDR2SA_Basic
 template 6-21
 JCI_RDR2SA_Full-IO
 template 6-15
 JCI_RDR8S_Basic
 template 6-23
 JCI_RDR8S_Full-IO
 template 6-18
 JCI_SI8_Full-IO template 6-10
 JCI_SIO8_Full-IO
 template 6-12
 JCI_x_Card-In template 6-40
 JCI_x_Card-In_IAD_TAMP
 template 6-43
 JCI_x_CICO template 6-48
 JCI_x_CICO_IAD_TAMP
 template 6-53
 JCI_x_Contact template 6-34
 JCI_x_Contact-w-Alarm
 template 6-37
 JCI_x_Elevator template 6-58
 Object Engine 3-5
 Object Hierarchy
 anti-loitering application 7-26
 anti-passback entry/exit
 application 7-24
 asset protection
 application 7-46
 assisted access application 7-49
 emergency exit
 application 7-13
 intrusion application 7-61
 intrusion with supervised inputs
 application 7-69
 keylock arming/disarming
 application 7-87
 low level elevator
 application 7-55
 multiple intrusion areas
 application 7-76
 occupancy with turnstiles
 application 7-42
 parking lot occupancy
 application 7-36
 reader with tamper switch
 application 7-19
 single portal with two readers
 application 7-6
 timed anti-passback
 application 7-21
 Object Interactions
 intrusion detection system 3-23
 Object List 3-7
 Object Naming 5-15
 Object Types 1-4
 Objects
 Access Control 3-3, 3-14
 adding for anti-loitering
 application 7-25
 adding for anti-passback entry/
 exit application 7-22
 adding for asset protection
 application 7-44
 adding for emergency exit
 application 7-9
 adding for intrusion
 application 7-58
 adding for intrusion with 4-state
 inputs application 7-65
 adding for keylock arming/
 disarming application 7-83
 adding for low level elevator
 application 7-52
 adding for multiple intrusion
 areas application 7-73
 adding for occupancy with
 turnstiles application 7-38
 adding for parking lot
 application 7-27
 adding for portal
 application 7-6
 adding for reader with tamper
 switch application 7-15
 adding for timed anti-passback
 application 7-20
 Anti-Loitering 3-3, 3-18
 Anti-Passback 3-3, 3-16
 Broadcast Management 3-43
 Calendar 3-42, 8-2
 CK722 Device 3-8
 common attributes 3-5
 common device attributes 3-5
 Controller Event 3-26
 controlling with Interlock
 objects 3-24
 controlling with Multiple
 Command objects 3-25
 defined 1-4
 described 3-5
 Door Sequence 3-15
 Elevator 3-38
 examples 3-1
 Folder 3-40
 Interlock 3-24
 Intrusion Announcer 3-21
 Intrusion Area 3-4, 3-19
 Intrusion Keypad/Display 3-22
 Intrusion Zone 3-4, 3-20

- JCI_I16_Full-IO template 6-6
 JCI_IO8_Full-IO template 6-8
 JCI_KDM_with-ACO template 6-62
 JCI_RDR2S_Card-In template 6-25
 JCI_RDR2S_Card-In-Card-Out template 6-30
 JCI_RDR2SA_Basic template 6-22
 JCI_RDR2SA_Full-IO template 6-17
 JCI_RDR8S_Basic template 6-23
 JCI_RDR8S_Full-IO template 6-18
 JCI_SI8_Full-IO template 6-11
 JCI_SIO8_Full-IO template 6-13
 JCI_x_Card-In template 6-41
 JCI_x_Card-In_IAD_TAMP template 6-44
 JCI_x_CICO template 6-49
 JCI_x_CICO_IAD_TAMP template 6-54
 JCI_x_Contact template 6-35
 JCI_x_Contact-w-Alarm template 6-38
 JCI_x_Elevator template 6-59
 KÖNÉ Controller 3-31
 KONE Elevator 3-34
 KONE Integration 3-27
 KONE IP COP 3-36
 KONE IP DOP 3-37
 KONE-IP Controller 3-33
 linking 3-1-3-2
 list of 3-7
 modifying for anti-loitering application 7-25
 modifying for anti-passback entry/exit application 7-23
 modifying for asset protection application 7-45
 modifying for assisted access application 7-48
 modifying for intrusion with 4-state inputs application 7-67
 modifying for low level elevator application 7-54
 modifying for multiple intrusion areas application 7-74
 modifying for parking lot application 7-35
 modifying for timed anti-passback application 7-21
 modifying single portal application 7-6
 Multiple Command 3-25
 Occupancy 3-3, 3-17
 Otis Controller 3-32
 Otis Elevator 3-35
 Otis Integration 3-28-3-29
 overview 3-6
 required for UL 1076 compliance A-2
 S300 Hardware Module 3-10
 S300 Reader Terminal 3-11
 S300 Trunk 3-9
 Schedule 3-2, 3-41, 8-1
 Security Binary Output 3-13
 Security Notification Class 3-44
 Security Supervised Input 3-12
 Site 3-39
 slaves 3-25
 template 1-4
 types of 3-6-3-7
 Occupancy 1-1
 Occupancy Applications 4-21
 parking lot 7-26
 turnstile 7-36
 Occupancy Objects 3-3
 described 3-17
 Offline Card Type C-2
 Offline Facility Code C-1
 Offline Mode 2-16, 8-22, C-1
 card access C-1
 deny access C-1
 wiring 6-27
 Offline PINs
 configuring C-3
 Online Status
 verifying 5-10
 Open Collector 2-12, 2-29
 Operating System
 upgrading for CK722 controllers 5-29
 Operator Override (Priority) 5-26
 Opto-isolation 3-46
 Organization 4-7
 Otis Controller Objects
 described 3-32
 Otis Elevator Objects
 described 3-35
 Otis Integration Objects
 as parent objects 3-32
 described 3-28-3-29
 Output Devices 4-22
 intrusion 4-24
 Output Objects
 See *Security Binary Output Objects*
 Output Points
 silencing 3-21
 Override Open Mode 8-6
 Overview
 P2000 SMS 2-2
 Overview Modes 2-17
 Overview Screens 2-17
- P**
 P2000 Host 1-4, 4-12
 communicating with CK722 controllers 2-9
 description of 2-3
 integration with P2000 SCT 2-3
 UL 1076 compliance procedures A-3
 P2000 Host Priority 5-28
 P2000 SCT 1-1, 2-3, 3-5
 defined 1-4
 description of 2-5
 integration with P2000 host 2-3
 starting a new project 5-1-5-14
 UL 1076 compliance procedures A-2
 P2000 Security Management System
 See *P2000 SMS*
 P2000 Site Parameters 5-3
 P2000 SMS 1-1, 1-3, 4-15, 5-1
 applications 4-1
 configuration 2-34
 defined 1-4
 introduction to components 2-1
 LAN recommendations 2-9
 organization 4-7
 system levels 2-2
 video imaging 4-10, 4-17
 P2000 System Configuration Tool
 See *P2000 SCT*
 P2000 Time Zones 5-18
 using 8-2
 P2000 Users 4-8
 P2000 Workstations 2-3
 Package Attributes
 selecting 7-3
 Package Tags
 defined 1-4
 Packages
 creating for door 5-12-5-13
 defined 1-4
 Paired Alarms 7-70
 Paired Sensors 4-27
 application 7-70
 Panel Tamper Input 5-11
 Panels 4-13
 See *Network Controllers*
 See also *Legacy Controllers*
 Panic Bars 4-15
 use with emergency exits 7-8
 See also *Crash Bars*
 Parking Lot Application 7-26
 Partitioning 4-24
 Partitions 5-2
 Passive Infrared Detectors 4-26
 Pasting
 calendars 8-22
 scheduled events 8-15
 schedules 8-22
 Peer-to-Peer Communication 5-3
 Peer-to-Peer Technology 3-5
 Perimeters
 defined 4-25
 Peripheral Devices 1-3
 P2000 SMS 2-1
 Person Entities 1-3, 4-1
 using identifiers with 4-8
 Personal Identification Number
 See *PIN Identifiers*

- PIN After Card
 in offline mode C-3
- PIN Digits
 configuring for offline mode C-3
- PIN Identifiers 4-9, 4-12–4-13
 described 4-11
 invoking access with B-2
 invoking access with keypad B-1
 invoking controller events with keypad B-6
 invoking timed overrides with keypad B-3–B-4
- Pinging CK722 devices 5-31
- PIR
 See *Passive Infrared Detectors*
- Portal Control 3-15
- Portal Entry Application 4-15
 reader/keypad combination 7-7
- Portal Entry/Exit
 Application 4-16, 7-8
- Portal Mode 8-6
- Portals
 defined 1-4
 fail safe/secure 4-18
 role of 7-4
- Portraits 4-9, 4-17
- Power Fail Input 5-11
- Pressure Mats/Sensors 4-26
- PREV Command 2-15
- Primary Servers 4-12
- Priorities
 P2000 host 5-28
 releasing multiple 5-28
- Prioritized Attributes
 releasing 5-27
 writing 5-27
- Prioritizing Attributes 5-24
 examples 5-25
- Priority
 assigning with exception schedule entries 8-6–8-7
- Priority Levels
 assigning for exceptions scheduling 8-4
- Privileges
 entity 4-7
 executive 7-44
- Project Requirements 5-2
- Proprietary Monitoring Facilities 4-30
- Protecting Assets 7-43
- Protocols
 B.M.S. 3-28
 BACnet 1-3
 communication 2-9
 E.M.S. 3-28
 RS-485 2-9
 SNMP 5-30
- Proximity Cards 4-10
- Pull Stations 4-27
- Push Button Switches 4-15
- Q**
- Quick Reference
 using keypad readers B-8–B-9
- R**
- Radio Frequency Identification Tags
 See *RFID Tags*
- RAID 4-12
- RDR2 Modules
 description of 2-29
- RDR2_Two_Door Template 6-64
- RDR2S 5-13
- RDR2S Modules
 described 2-12
 mounting 2-12
 wiring red/green lights 6-33
- RDR2S_Entry_Exit
 Template 6-24, 6-64
- RDR2S_One_Door
 Template 6-64
- RDR2S_One_Door_App
 Template 6-64
- RDR2S_One_Door_IO
 Template 6-64
- RDR2S_SI_O_ONLY
 Template 6-64
- RDR2S_Two_Door
 Template 6-64
- RDR2S_Two_Door_App
 Template 6-64
- RDR2S_Two_Door_IO
 Template 6-64
- RDR2S-A 5-13
- RDR2S-A and RDR2S
 modes of operation 6-24
- RDR2S-A Modules
 described 2-13
 mounting 2-13
 two reader support 7-7
- RDR8S 5-11
- Read Only Attributes 3-5
- Reader Terminal Objects 3-2
 See *S300 Reader Terminal Objects*
- Readers 1-3, 4-13
 denying access in offline mode C-1
 RFID 4-11, 7-43
 using keypad B-1
 with tamper switches 7-14
- Real-time Applications 2-1
- reboot Command 5-31
- Receiving
 alarm messages 5-30
 system event messages 5-30
- Recommended Priorities 5-26
- Redundancy 4-12
- Redundant Systems 4-12
- Referencing
 items with Schedule objects 8-5
- Relays 1-3
 ADA 7-47
 SPDT 2-12, 2-29, 2-31–2-32
- Releasing
 multiple priorities 5-28
 priorities 5-24
 prioritized attributes 5-27
- Remote Redundancy 4-12
- Removing
 exception schedules 8-18
 schedule items 8-19
- Reporting
 intrusion alarms 4-27
 of inactive doors 5-17
- Request to Exit Devices 4-14, 7-4
- Requirements
 per CK722 5-5
 per door 5-12
 per hardware module 5-11
 per project 5-2
 per site 5-3
- Resetting
 zone sensors 2-15, 2-25–2-26
- Restoring CK722
 Functionality 5-29
- REX Devices 7-46
 See *Request to Exit Devices*
- RFID Readers 7-43
- RFID Tags 7-43
 described 4-11
- Role Name 5-15
- RS-232 Null Modem Cable 5-32
- RS-485
 bus 3-1, 3-45
 communication protocol 2-9
 ports 2-11
- RST Command 2-15
- RTO
 See *Reader Terminal Objects*
- Rules
 when downloading 5-8
- S**
- S300 Bus 2-11–2-13
 ground wiring on 3-46
- S300 Expansion Enclosures 2-34
- S300 Hardware Module Objects
 described 3-10
- S300 Hardware Modules
 disconnected from controller C-1
- S300 I/O Modules
 description of 2-11
- S300 Keypad/Display Modules
 description of
- S300 Reader Terminal Objects
 described 3-11
- S300 Trunk Objects
 described 3-9
- S300_I16 Template 6-65
- S300_IO8 Template 6-65
- S300_SI8 Template 6-65
 graphical representation 6-10
- S300_SIO8 Template 6-65
- S300-DIN Enclosures 2-34
- S300-DIN-L Enclosures 2-34–2-35

- S300-DIN-RDR2S Modules
See *RDR2S Modules*
- S300-DIN-RDR2S-A Modules
See *RDR2S-A Modules*
- S300-DIN-S Enclosures 2-34, 2-36
- S300-I16 Modules
See *I16 Modules*
- S300-IO8
See *IO8 Modules*
- S300-KDM
See *S300 Keypad/Display Modules*
- S300-RDR2 Modules
See *RDR2 Modules*
- S300-SI8 Modules
See *SI8 Modules*
- S300-SIO8 Modules
See *SIO8 Modules*
- S300-XL Enclosures 2-34, 2-37
- S300-XS Enclosures 2-34, 2-38
- S300-XXS Enclosures 2-34, 2-39
- Sagem Morpho MA520
Format D-3
- Schedule Objects
described 3-41
factory installed 7-53
linking to objects 3-2
linking to time zones 5-5, 5-15
using 8-1
using P2000 time zones with 8-2
- Scheduled Events
adding 8-14
copying 8-15
deleting 8-15
displaying 8-13
editing 8-15
pasting 8-15
- Scheduled Items 8-5
adding 8-18
described 8-5
editing 8-18
removing 8-19
- Schedules 8-1
active/inactive date range of 8-11
adding exception 8-16
adding items to 8-5
copying 8-22
creating 8-12
deleting 8-23
disabling 8-22
displaying 8-13
editing attributes of 8-22
editing effective period of 8-19
editing exception 8-17
exception 3-41–3-42, 8-3
managing 8-12
pasting 8-22
removing exception 8-18
weekly 8-3
- Scheduling 4-18, 5-18
data verification 8-11–8-12
- defining exceptions 8-2
described 8-1
effective period 8-11
fast clock feature 8-11
uses 8-1
week and day format 8-7
wild cards 8-7
- Scheduling (Priority) 5-27
- Scramble Mode C-3
- Scrolling
KDM menu 2-15
- Secure Signals
intrusion detection 4-30
- Secured Areas 4-11
- Secured Premises Notification Settings A-1
- Security Applications
access control 7-4
- Security Binary Output Objects
described 3-13
interaction with Intrusion Zone objects 3-20
- Security Flags 5-2–5-3
- Security Management Systems 4-12
- Security Notification Class Objects
described 3-44
- Security Supervised Input Objects
described 3-12
interaction with Intrusion Zone objects 3-20
- Segment Length
max distance 2-9
- Select Alarm and Status Mode 2-18
- Select Alarms Only Mode 2-18
- Selecting
offline card type C-2
package attributes 7-3
- Sensor 26 Bit Format D-4
- Sensor 26 Bit Forward Format D-4
- Sensor 26 Bit Inv Forw Format D-4
- Sensor 26 Bit Inv Rev Du Format D-4
- Sensor 26 Bit Inv Rev Format D-4
- Sensor 26 Bit Inverted Format D-4
- Sensor 26 Bit Rev Duress Format D-4
- Sensor 26 Bit Reverse Format D-4
- Sensors 1-3, 4-24
door 4-25
dual technology 4-26
glass break 4-26
infrared beam interruption 4-26
intrusion detection 4-25
microwave 4-26
movement 4-26
paired 4-27
passive infrared 4-26
pressure mats 4-26
- resetting 2-15, 2-25–2-26
stopping test of 2-15
testing 2-15, 2-26–2-27
ultrasonic 4-26
vibration 4-25
- Server Cluster 4-12
- Servers 4-12
P2000 2-3
- Setting
IP address manually 5-31
- Shunt 4-22
- SI8 Modules 2-33
- Signaling
arming/disarming 7-81
central stations 4-30
- Signals
abort 4-30
alarm 4-30
cancel 4-30
intrusion detection 4-30
secure 4-30
- Signatures 4-9, 4-17
- Silencing
annunciators 2-15, 2-27–2-28
output points 3-21
- Simple Network Management Protocol Traps
See *SNMP Traps*
- Single Pole Double Throw
See *SPDT Relays*
- SIO8 Modules 2-32
- SIO8_Eight_Zone_Outputs Template 6-65
- Site Objects
described 3-39
- Site Requirements 5-3
- Sites
defined 1-4
- Slave Objects 3-25
- Slide Gate Operators 7-26
- SLNC Command 2-15
- Small Enclosures
S300-DIN-S 2-36
S300-XS 2-38
S300-XXS 2-39
- Smart Cards 4-10
- SNMP Traps
receiving 5-30
- Soft Entry/Exit Rule 7-22
- SPDT Relays 2-12, 2-29, 2-31–2-32
- Special Access 7-47
- Sponsoring Entities 4-3–4-4
- Standard Wiegand
selecting for offline mode C-2
- Starting
new SCT project 5-1–5-14
- STAT Command 2-15
- State Machines
defined 5-23
- States
in Multiple Command objects 3-25

- intrusion detection systems 4-31
- Static IP Address assigning 5-7, 5-33
- Status viewing entity 4-7
- Status Menu accessing with KDM 2-15
- Status Output Attribute 7-81
- Stealing Unused Field Points 5-19
- STOP Command 2-15
- Stopping sensor test 2-15
- Strikes electric 4-14
- Subordinate Objects 3-25
- Supervised Input Modules S18 2-33
- Supervised Input/Output Modules SIO8 2-32
- Supervised Inputs 7-62
- Supervised Intrusion Detection Systems 4-31
- Supervisory Controllers 4-13
- Supervisory Devices 1-3
- Supervisory Level P2000 SMS 2-1
- Swing Gate Operators 7-26
- Switches 4-13 key override 4-15 normally closed 2-12, 2-29 normally open 2-12, 2-29 push button 4-15
- Synchronizing archive database 5-8–5-10 rules 5-8
- Synchronous Operation 5-20–5-21 advantages of 5-21 constraints of 5-21
- Synchronous vs. Asynchronous Operation 5-20 example 5-21
- System Accounts 1-3, 4-1 categories of 4-2
- System Components CK722 controllers 2-6 P2000 host 2-4 P2000 SCT 2-5 RDR2S module 2-12 S300 I/O modules 2-11 S300 Keypad/Display Module 2-14
- System Configuration CK722 controllers 2-8
- System Event Messages receiving 5-30
- System Events described 4-23
- System Levels 2-2
- System Status verifying 5-14
- System Test 5-14
- T**
- Tags defined 1-4 RFID 4-11
- Tamper Inputs 7-63
- Tamper Switches 7-14
- Template Instances See *Packages*
- Template Names max number of characters 6-2
- Template Objects 1-4
- Templates 3-5 adapting job-specific 7-3 basic h/w module 6-21–6-23 Bi-directional_Man_Trap 6-63 categories of 6-4 copying 6-3, 7-1–7-2 creating h/w modules with 5-11 creating job-specific 7-1–7-3 creating packages from 5-12–5-13 customizing 6-3 defined 1-4 described 6-1 door h/w module 6-24–6-34 Eight_Door_Occupancy 6-63 Eight_Zone_Area_Keypad_Ann un 6-63 Eight_Zone_Area_No_Keypad 6-64 full-io h/w module 6-4–6-20 importing 7-1 information framework 6-3 JCI standard type 6-1 JCI_I16_Full-IO 6-5 JCI_IO8_Full-IO 6-7 JCI_KDM_with-ACO 6-61 JCI_RDR2S_Card-In 6-24 JCI_RDR2S_Card-In-Card-Out 6-27 JCI_RDR2SA_Basic 6-21 JCI_RDR2SA_Card-In 6-33 JCI_RDR2SA_Card-In-Card-O ut 6-33 JCI_RDR2SA_Full-IO 6-14 JCI_RDR8S_Basic 6-23 JCI_RDR8S_Full-IO 6-18 JCI_SI8_Full-IO 6-9 JCI_SIO8_Full-IO 6-12 JCI_x_Card-In 6-40 JCI_x_Card-In_IAD 6-42 JCI_x_Card-In_IAD_TAMP 6-43 JCI_x_Card-In_TAMP 6-42 JCI_x_CICO 6-47 JCI_x_CICO_IAD 6-52 JCI_x_CICO_IAD_TAMP 6-5 2 JCI_x_CICO_TAMP 6-51 JCI_x_Contact 6-34 JCI_x_Contact-w-Alarm 6-36 JCI_x_Elevator 6-58 job-specific 5-15, 7-1 legacy 6-63–6-65
- miscellaneous 6-58–6-63 modifying 6-3, 7-1 naming convention of 6-2 RDR2_Two_Door 6-64 RDR2S_Entry_Exit 6-24, 6-64 RDR2S_One_Door 6-64 RDR2S_One_Door_App 6-64 RDR2S_One_Door_IO 6-64 RDR2S_SI_O_ONLY 6-64 RDR2S_Two_Door 6-64 RDR2S_Two_Door_App 6-64 RDR2S_Two_Door_IO 6-64 S300_I16 6-65 S300_IO8 6-65 S300_SI8 6-65 S300_SIO8 6-65 SIO8_Eight_Zone_Outputs 6-6 5
- Thirty_Two_Zone_Area_Annun 6-65 types of 6-1
- Tera Term Pro 5-32
- Terminal Emulation Software 5-32
- TEST Command 2-15
- Testing intrusion zone sensors 2-15, 2-26–2-27 job-specific templates 5-16
- Testing the System 5-14
- Thirty_Two_Zone_Area_Annun Template 6-65
- Time anti-loitering 3-18
- Time Rule anti-passback 3-16, 4-19–4-20, 7-19
- Time Zone Attribute 3-39 Schedule object 3-41
- Time Zones 5-3, 5-18 elevator application 7-51 linking to schedule objects 5-5, 5-15 using 8-2
- Time/Value Pairs 8-1, 8-13 described 8-5 with exceptions 8-4
- Timed Anti-Passback Application 7-19
- Timed Overrides B-1 invoking with keypad B-2–B-4
- Time-out Setting keypads B-1
- Toggling Between Calendar Views 8-22
- tracert Command 5-31
- Tracking assets 4-2, 7-43 entity movement 4-21 number of vehicles in parking lot 7-26
- Transactions viewing identifier 4-7
- Transitional Areas

- man-traps 4-21
Traps
 receiving SNMP 5-30
Triggering
 controller events 3-26
Trouble Inputs 7-63
Troubleshooting
 CK722 controllers 5-29
Trunk Objects
 See *S300 Trunk Objects*
Turning Off Schedules 8-22
Turnstile Application 7-36
Types
 of entities 4-1
 of objects 1-4, 3-6
- U**
UDFs
 See *User-defined Fields*
UL A-1
UL 1076 Compliance A-1
 configuring ACO A-2
 configuring Controller Event
 object A-3
 configuring Intrusion Zone
 object A-3
P2000 host procedures A-3
P2000 SCT procedures A-2
Ultrasonic Detectors 4-26
Underwriters Laboratories
 See *UL*
Unsecured Areas 4-11
Unsupervised Input Modules
 I16 2-30
Unsupervised Input/Output
 Modules
 IO8 2-31
Unsupervised Inputs
 intrusion application 7-71
Unsuppressing
 life safety alarm signals A-1
Upgrading
 CK722 operating system 5-29
User Accounts 4-8
User-defined Fields 4-8
Using
 catch-all card format 5-18
 command line interface 5-31
 keypad readers B-1, B-8–B-9
 P2000 time zones 8-2
 wild cards 8-7
- V**
Validation 4-7
Vehicle Loops 7-27–7-28, 7-36
Verifying
 CK722 online status 5-10
 schedule dates 8-11–8-12
 system status 5-14
Vibration Detectors 4-25
Video Imaging 4-10, 4-17
 typical configuration 4-17
Viewing
 CK722 firmware version 5-31
- CK722 IP info 5-31
Visitor Management 4-18
- W**
WAN
 See *Wide Area Network*
Week and Day Entries
 using with wild cards 8-10
Week and Day Format 8-7
Weekly Schedules 4-18, 8-1, 8-3
 defining exceptions to 8-6
 week and day format 8-7
 with time/value pairs 8-5
Wide Area Network 1-4, 2-5, 4-30
Wiegand 2-29
 cards 4-10, C-2
 interface 2-12
Wild Cards
 described 8-7
 specifying dates 8-8
 using with date range 8-9
 using with week and day
 entries 8-10
Wiring
 ADA relay 7-47
 network 2-9
 offline mode 6-27
 S300 bus ground 3-46
Wizards
 load 5-8
Workstations
 P2000 2-3
Writing
 prioritized attributes 5-27
- X**
x-Templates 6-1
 correctly mapping points
 from 5-16
 door type 6-34
- Z**
ZONE Command 2-15
Zones 4-28
 accessing list of 2-15
 activating 2-15, 2-23–2-24
 bypassing 2-15, 2-22–2-23
 resetting sensors for 2-25–2-26
 testing sensors for 2-26–2-27
Zones Screen 2-17