Johnson Controls

Object Library

Access Control Object

Version 1.0, December 2008
24-10239-49 Revision B

These instructions are supplemental. Some times they are supplemental to other manufacturer's documentation. Never discard other manufacturer's documentation. Publications from Johnson Controls, Inc. are not intended to duplicate nor replace other manufacturer's documentation.

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.

ACCESS CONTROL OBJECT

INTRODUCTION

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 1 illustrates basic interactions.

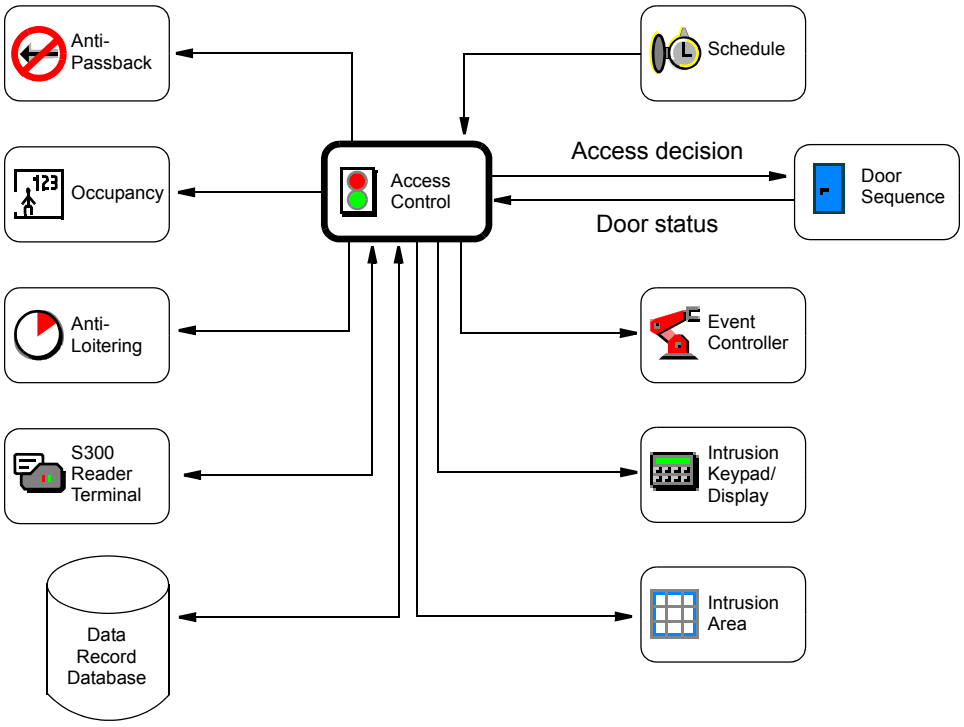
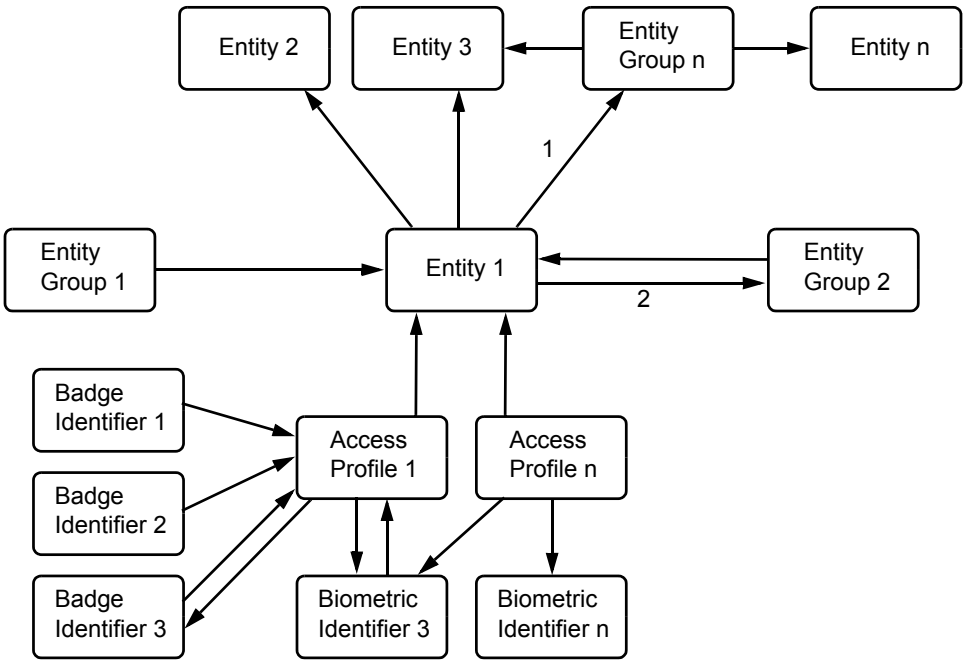


Figure 1: Access Control Object

DATA RECORD RELATIONSHIP OVERVIEW

The following diagram contains an example of relations between selected data records used by the Access Control object.

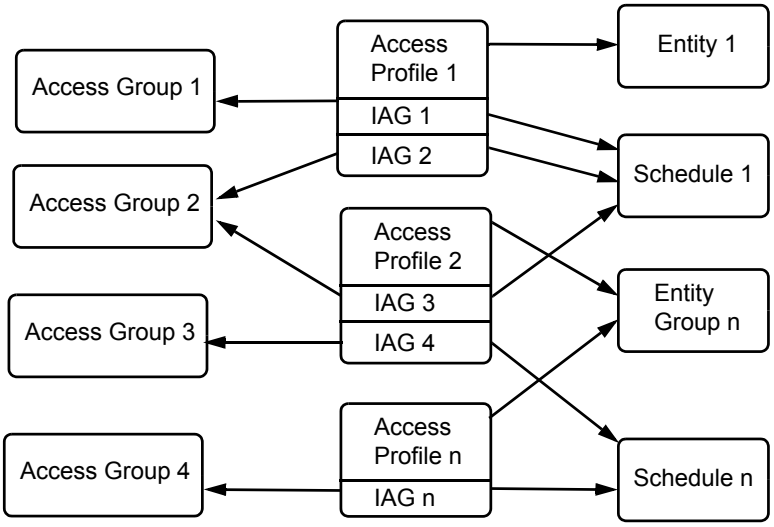


All relations are expressed as the ID of the item the arrow points to. The relations labeled with numbers are one of the following types:

1 = Must be accompanied by at least one member of the referenced Entity Group

2 = Entire Entity Group must be present to be granted access

The following diagram contains an example of data records of type Entity, Access Profile and Access Group.



IAG: Intrusion/Access Group and Timezone Pair (combines an intrusion group or an access group with a Schedule object on a per access profile basis)

ATTRIBUTES

This section describes visible attributes specific to the Access Control object. This object also contains:

- Attributes common to all objects in the P2000 Security Management System. For details, see the *General Object Information* document.
- Internal attributes, which are invisible to the user and cannot be modified directly, but may be referred to throughout this document.

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Access Denied On Open	3039	Boolean	WCA	-	-
Access Enabled	2989	Boolean	W	-	Redirected to *Access Enabled Default*
Access Enabled Default	4706	Boolean	WCA	-	-
Access Level	3003	Unsigned8	WCA	-	0 - 99

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Access Mask	3004	Bitstring	WCA	-	100 bits
Access Model	3002	Enumeration	WCA	1	0 = None 1 = Group 2 = Level 3 = Mask 4 = Group-or-Level 5 = Level-or-Mask 6 = Mask-or-Group 7 = Group-and-Level 8 = Level-and-Mask 9 = Mask-and-Group 10 = Group-or-(Level-and-Mask) 11 = Group-and-(Level-or-Mask) 12 = Level-or-(Mask-and-Group) 13 = Level-and-(Mask-or-Group) 14 = Mask-or-(Group-and-Level) 15 = Mask-and-(Group-or-Level) 16 = Group-or-Level-or-Mask 17 = Group-and-Level-and-Mask
Alternate Access	2916	Enumeration	WCA	1	0 = Never 1 = By Mask 2 = Always
Alternate Access Mask	3038	Bitstring	WCA	-	100 bits
Anti-Loitering Exemption Mask	4246	Bitstring	WCA	1st = 1	100 bits
Anti-Loitering In Object	3804	Object reference	WCAN	-	-
Anti-Loitering Out Object	3805	Object reference	WCAN	-	-
Anti-Loitering Transition	3802	Enumeration	WCA	-	0 = Disabled 1 = On Granted 2 = On Granted or Valid 3 = On Identification
Anti-Loitering Transition Needs Open	3803	Boolean	WCA	-	-
Anti-Passback Check	3789	Enumeration	WCA	-	0 = Disabled 1 = Monitor Only 2 = Enforce, Only When Operational 3 = Enforce, and Deny During Trouble
Anti-Passback Exemption Mask	3794	Bitstring	WCA	First bit = 1	100 bits
Anti-Passback In Object	3792	Object reference	WCAN	-	-
Anti-Passback Out Object	3793	Object reference	WCAN	-	-

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Anti-Passback Transition	3790	Enumeration	WCA	-	0 = Disabled 1 = On Granted 2 = On Granted or Valid 3 = On Identification
Anti-Passback Transition Needs Open	3791	Boolean	WCA	-	-
Asset Identification	3863	Enumeration	WCA	-	0 = Unaccompanied Asset Increase and Decrease 1 = Unaccompanied Asset Increase Only 2 = Unaccompanied Asset Decrease Only
Asset Notification	3019	Enumeration	WCA	-	0 = All 1 = Unaccompanied 2 = None
Asset Processing	2999	Enumeration	WCA	-	0 = Disabled 1 = Asset Tracking Only 2 = Deny All On Unaccompanied Asset
Asset Timeout	2965	Unsigned16	WCA	5	In seconds Min value = 5 Max value = 1000
Asset Trailing Time	3860	Unsigned16	WCA	-	In seconds Min value = 0, Max value = 1000
Central Data Check	3838	Boolean	WCA	1	-
Central Data Timeout	2963	Unsigned16	WCA	5	In seconds Min value = 1, Max value = 1000
Central Status Check	2996	Enumeration	WCA	-	0 = Disabled 1 = When Online 2 = Always
Central Status Exemption Mask	3032	Bitstring	WCA	First bit = 1	100 bits
Central Status Timeout	2964	Unsigned16	WCA	5	In seconds Min value = 1 Max value = 1000
Controller Event Object List	4136	List of object reference	WCAN	-	Max 100 entries
Decision Number	2934	Unsigned32	F	-	-
Denied Notification	3016	Enumeration	WCA	-	0 = All 1 = All but Host Unreachable 2 = None
Direction	746	Enumeration	WCA	-	0 = Undefined 1 = Ingress 2 = Egress
Door Sequence Object	3022	Object reference	WCAN	-	-

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Encrypted Wiegand Facility Codes	3034	Unsigned16	WCA	-	-
Entity Identification	3864	Enumeration	WCA	-	0 = Unaccompanied Entity Increase and Decrease 1 = Unaccompanied Entity Increase Only 2 = Unaccompanied Entity Decrease Only
Entity Notification	3020	Enumeration	WCA	-	0 = All 1 = Unaccompanied 2 = None
Entity Processing	3000	Enumeration	WCA	-	0 = Disabled 1 = Entity Tracking Only 2 = Deny Unaccompanied Entity 3 = Deny All On Unaccompanied Entity
Entity Timeout	2966	Unsigned16	WCA	5	In seconds Min value = 5, Max value = 1000
Entity Trailing Time	3861	Unsigned16	WCA	-	In seconds Min value = 0, Max value = 1000
Error Notification	3040	Boolean	WCA	1	-
Executive Privilege Mask	2939	Bitstring	WCA	First bit = 1	100 bits
Facility Code List	2988	List of Unsigned32	WCA	-	Max 100 entries
Fault Cause	2896	Enumeration	F	-	0 = None 1 = Out Of Memory 2 = Door Sequence 3 = Primary Reader 4 = Secondary Reader
Granted Notification	3015	Enumeration	WCA	-	0 = Always 1 = No Entry 2 = On Open Only 3 = Never
Group Identification	3865	Enumeration	WCA	-	0 = Incomplete Group Increase and Decrease 1 = Incomplete Group Increase Only 2 = Incomplete Group Decrease Only
Group Notification	3021	Boolean	WCA	1	-
Group Processing	3001	Enumeration	WCA	-	0 = Disabled 1 = Group Tracking Only 2 = Deny Incomplete Group 3 = Deny All On Incomplete Group

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Group Timeout	2967	Unsigned16	WCA	5	In seconds Min value = 5, Max value = 1000
Identifier Timeout	2962	Unsigned16	WCA	-	In seconds
Incomplete Groups	2955	Unsigned16	F	-	-
Intrusion Area Object	3837	Object reference	WCAN	-	-
Keypad Display Object	3013	Object reference	WCAN	-	-
Keypad Timeout	2961	Unsigned8	WCA	15	In seconds
Keypad Trigger	2960	Enumeration	WCA	1	0 = Pound Key Only 1 = Primary Channel 2 = Secondary Channel 3 = Any Channel 4 = All Channels
Lockdown Notification	3018	Boolean	WCA	1	-
Manual Reader	2998	Boolean	WCA	-	-
Notification Class	17	Unsigned32	WCA	1	-
Notify Priority	3644	Unsigned8	WCA	-	-
Number	4031	Unsigned16	A	-	-
Occupancy Check	3795	Enumeration	WCA	-	0 = Disabled 1 = Monitor Only 2 = Enforce, Only When Operational 3 = Enforce, and Deny During Trouble
Occupancy Exemption Mask	3800	Bitstring	WCA	First bit = 1	100 bits
Occupancy In Object	3798	Object reference	WCAN	-	-
Occupancy Out Object	3799	Object reference	WCAN	-	-
Occupancy Transition	3796	Enumeration	WCA	-	0 = Disabled 1 = On Granted 2 = On Granted or Valid 3 = On Identification
Occupancy Transition Needs Open	3797	Boolean	WCA	-	-
Override Mask	2940	Bitstring	WCA	Second bit = 1	100 bits
Override Notification	3017	Boolean	WCA	1	-
PIN Attempt Limit	2992	Unsigned8	WCA	3	0 - 7 attempts
PIN Attempt Sector	3868	Unsigned8	WCA	-	0 - 9
PIN Digits	2991	Unsigned8	WCA	4	1 - 9 digits

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
PIN Duress	2993	Enumeration	WCA	-	0 = None 1 = Plus One 2 = On Nine 3 = Minus One 4 = First Digit 5 = Last Digit
PIN Suppressed	2990	Boolean	W	-	-
PIN Use	3869	Enumeration	WCA	-	0 = For Access Only 1 = For Intrusion Only 2 = For Access and Intrusion
Present Value	85	Enumeration	F	-	0 = Not initialized 1 = Operational 2 = Unknown 3 = Fault
Primary Reader Object	2958	Object reference	WCAN	-	-
Reader Enabled	2956	Boolean	WCA	1	-
Secondary Reader Object	2959	Object reference	WCAN	-	-
Security Level	4036	Unsigned8	WCA	-	0 - 99
Security Mask	3006	Bitstring	WCA	-	100 bits
Security Mode	3005	Enumeration	WCA	-	0 = Restricted 1 = Exceptional
Security Mode Active Level	3008	Unsigned8	WCA	100	0 - 100
Set 1 Enabled	2968	Boolean	WV	-	Redirected to *Set 1 Enabled Default*
Set 1 Enabled Default	3781	Boolean	WCA	1	-
Set 1 First Identifier Format	2970	Unsigned32	WCA	-	-
Set 1 Mode	3782	Enumeration	WCA	-	0 = Regular Access 1 = Validation Only 2 = Tracking Only
Set 1 PIN Required	2969	Boolean	WCA	-	-
Set 1 Second Identifier Format	2971	Unsigned32	WCA	-	-
Set 1 Third Identifier Format	2972	Unsigned32	WCA	-	-
Set 2 Enabled	2973	Boolean	WV	-	Redirected to *Set 2 Enabled Default*
Set 2 Enabled Default	3783	Boolean	WCA	-	-
Set 2 First Identifier Format	2975	Unsigned32	WCA	-	-

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Set 2 Mode	3784	Enumeration	WCA	-	0 = Regular Access 1 = Validation Only 2 = Tracking Only
Set 2 PIN Required	2974	Boolean	WCA	-	-
Set 2 Second Identifier Format	2976	Unsigned32	WCA	-	-
Set 2 Third Identifier Format	2977	Unsigned32	WCA	-	-
Set 3 Enabled	2978	Boolean	WV	-	Redirected to *Set 3 Enabled Default*
Set 3 Enabled Default	3785	Boolean	WCA	-	-
Set 3 First Identifier Format	2980	Unsigned32	WCA	-	-
Set 3 Mode	3786	Enumeration	WCA	-	0 = Regular Access 1 = Validation Only 2 = Tracking Only
Set 3 PIN Required	2979	Boolean	WCA	-	-
Set 3 Second Identifier Format	2981	Unsigned32	WCA	-	-
Set 3 Third Identifier Format	2982	Unsigned32	WCA	-	-
Set 4 Enabled	2983	Boolean	WV	-	Redirected to *Set 4 Enabled Default*
Set 4 Enabled Default	3787	Boolean	WCA	-	-
Set 4 First Identifier Format	2985	Unsigned32	WCA	-	-
Set 4 Mode	3788	Enumeration	WCA	-	0 = Regular Access 1 = Validation Only 2 = Tracking Only
Set 4 PIN Required	2984	Boolean	WCA	-	-
Set 4 Second Identifier Format	2986	Unsigned32	WCA	-	-
Set 4 Third Identifier Format	2987	Unsigned32	WCA	-	-
Team Exemption Mask	3011	Bitstring	WCA	First bit = 1	100 bits
Team Mask	3123	Bitstring	WCA	-	100 bits
Team N-Man Rule	3010	Unsigned8	WCA	-	1 - 20 entities
Team Processing	4139	Boolean	WCA	-	-
Team Timeout	3051	Unsigned16	WCA	5	In seconds Min value = 1, max value = 1000
Time Zone Check	2994	Boolean	WCA	1	-

Table 1: Access Control Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options /Range
Tracking Object	3836	Object reference	WCAN	-	-
Unaccompanied Assets	2953	Unsigned16	F	-	-
Unaccompanied Entities	2954	Unsigned16	F	-	-
Validation Only Mask	3031	Bitstring	WCA	-	100 bits

A - Archive, C - Configurable, F - PMI (Person/Machine Interface) refreshing, N - Value not required, W - Writable, V - Initial value redirected

Access Denied On Open – Specifies whether the portal must be closed for access to be granted, unless the access profile has executive privilege, or a duress condition is signaled.

Access Enabled – Specifies whether the Access Control object may grant access to requests that do not have executive privilege. See “Access Decision Algorithm” on page 40 for details.

Access Enabled Default – Specifies the default value for the *Access Enabled* attribute. Changing this attribute automatically sets the *Access Enabled* attribute to the same value.

Access Level – Specifies the access level, that must be matched or surpassed by an identified access profile's access level to be granted access, provided access levels are used in the access decisions.

Access Mask – When access masks are used in the access decisions, to be granted access an identified access profile's normal mask must have at least one set bit in common with the access mask.

Access Model – Specifies which combination of permanent access models is used to determine access. An access request is denied whenever the combination of access requirements specified in this attribute is not met. The access decision is made by comparing the access group, access level, and access mask information in the access profile against the respective attributes in the Access Control object. See “Team Processing” on page 37 for details.

Combination	Access granted when...
None	(never)
Group	...access group allows access.
Level	...access level allows access.
Mask	...access mask allows access.
Group-or-Level	...access group or the access level allow access.

Combination	Access granted when...
Level-or-Mask	...access level or the access mask allow access.
Mask-or-Group	...access mask or the access group allow access.
Group-and-Level	...both access group and access level allow access.
Level-and-Mask	...both access level and access mask allow access.
Mask-and-Group	...both access mask and access group allow access.
Group-or-(Level-and-Mask)	...access group or both access level and access mask allow access.
Group-and-(Level-or-Mask)	...access group and either access level or access mask allow access.
Level-or-(Mask-and-Group)	...access level or both access mask and access group allow access.
Level-and-(Mask-or-Group)	...access level and either access mask or access group allow access.
Mask-or-(Group-and-Level)	...access mask or both access group and access level allow access.
Mask-and-(Group-or-Level)	...access mask and either access group or access level allow access.
Group-or-Level-or-Mask	...either access group or access level or access mask allow access.

Alternate Access – Specifies under which circumstances the alternate access feature at the Door Sequence object is invoked. The options are:

- Never - The alternate access feature is never invoked.
- By Mask - The alternate access feature is invoked when the identified access profile's normal mask, or security mask in the emergency mode, has at least one set bit in common with the Access Control object's *Alternate Access Mask* attribute.
- Always - The alternate access feature is invoked with every access grant.

Alternate Access Mask – Specifies the alternate access mask. To be eligible to invoke the alternate access feature at the Door Sequence object, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the alternate access mask.

Anti-Loitering Exemption Mask –Specifies the mask, that an identified access profile's normal mask, or security mask in emergency mode, must have at least one set bit in common with to be exempt from being monitored by the Anti-Loitering object.

Anti-Loitering In Object – Specifies the Anti-Loitering object that this Access Control object should inform upon an access transition, so that it can add the entity to its list of monitored entities. If the Access Control object does not handle any entities entering the anti-loitering area, this attribute should be blank. The referenced object must reside on the same controller.

Anti-Loitering Out Object – Specifies the Anti-Loitering object that this Access Control object should inform, so that it can remove the entity from its list of monitored entities. If the Access Control object does not handle any entities leaving the anti-loitering area, this attribute should be blank. The referenced object must reside on the same controller.

Anti-Loitering Transition – Specifies upon which event the Access Control object informs the Anti-Loitering object(s) that the entity's anti-loitering monitor status should be changed. The options are:

- Disabled - Disables the anti-loitering feature. The monitor status should not change on an access decision.
- On Granted - The monitor status should change on an access decision of type "Granted" and "Granted Alternate."
- On Granted Or Valid - The monitor status should change on an access decision of type "Granted," "Granted Alternate," "Valid," or "Silent."
- On Identification - The monitor status should change on any access decision.

Anti-Loitering Transition Needs Open – Specifies whether the portal must be open during the access time for the anti-loitering timer to start.

Anti-Passback Check – Specifies how the anti-passback rules should be applied. See "Anti-Passback" on page 50 for details. The options are:

- Disabled - The anti-passback rules don't not have an impact on the access decision.
- Monitor Only - Anti-passback rule violations are reported, but have no impact on the access decision.
- Enforce, Only When Operational - Anti-passback rule violations are reported and cause access requests to be denied when the anti-passback decision is of the denied category or unknown entity. Anti-passback decisions of not operational or offline do not cause access to be denied, but are only reported.
- Enforce, and Deny During Trouble - Anti-passback rule violations are reported and cause access requests to be denied when the anti-passback decision is of the denied category, unknown entity, not operational, or offline.

Anti-Passback Exemption Mask – Specifies the anti-passback exemption mask. To be exempt from the anti-passback rules, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the anti-passback exemption mask.

Anti-Passback In Object – Specifies the Anti-Passback object that this Access Control object should consult in making the access decision and inform upon an access transition, so that it can account for the entity having entered the area. If the Access Control object does not handle entry requests, this attribute should be blank. The referenced object must reside on the same controller.

Anti-Passback Out Object – Specifies the Anti-Passback object that this Access Control object should consult in making the access decision and inform upon an access transition, so that it can account for the entity having left the area. If the Access Control object does not handle exit requests, this attribute should be blank. The referenced object must reside on the same controller.

Anti-Passback Transition – Specifies upon which event the Access Control object informs the Anti-Passback object(s) that the anti-passback status should be updated. The options are:

- Disabled - The anti-passback status should not be updated on an access decision.
- On Granted - The anti-passback status should be updated on an access decision of type "Granted" and "Granted Alternate."
- On Granted Or Valid - The anti-passback status should be updated on an access decision of type "Granted," "Granted Alternate," "Valid," or "Silent."
- On Identification - The anti-passback status should be updated on any access decision in which the entity is identified.

Anti-Passback Transition Needs Open – Specifies whether the portal must be open during the access time for the anti-passback status be updated.

Asset Identification – Specifies what happens to the number of unaccompanied assets after being identified by the Access Control object. This attribute allows for an optimization of the asset processing feature, in which Access Control objects dedicated to long range tracking readers would only be able to increase the number of unaccompanied assets, and Access Control objects dedicated to reading owner's badges would only be able to decrease the number of unaccompanied assets.

Asset Notification – Specifies which asset related notifications are generated. See "Asset Processing" on page 33 for details. The options are:

- All - Both Asset Tracked and Unaccompanied Assets notifications are generated.
- Unaccompanied - Only Unaccompanied Assets notifications are generated.
- None - Neither Asset Tracked nor Unaccompanied Assets notifications are generated.

Asset Processing – Specifies whether access will be denied when unaccompanied assets are detected. See "Asset Processing" on page 33 for details. The options are:

- Disabled - No asset processing.
- Asset Tracking Only - An asset is tracked upon identification, and a notification is generated when the asset is not accompanied by an owner within a defined time.
- Deny All on Unaccompanied Asset - As above, but access to the portal is denied for all entities except those with executive privilege when unaccompanied assets are present.

Asset Timeout – Specifies the time an asset must remain in the list of unaccompanied assets to be reported as unaccompanied. The Access Control object may extend this value by 9% to handle the expiration of multiple assets in a single operation. See "Asset Processing" on page 33 for details.

Asset Trailing Time – Specifies the time an asset may trail its owner without being reported as unaccompanied. See "Asset Processing" on page 33 for details.

Central Data Check – Specifies whether the Access Control object operates in shared or local mode. See "Deferred Decisions" on page 45 for details.

Central Data Timeout – Specifies the maximum time allowed to wait before resuming the algorithm after a decision is deferred to the host with the request for central data. While the decision is deferred to the host, the Access Control object ignores any data received through any of its channels.

Central Status Check – Specifies whether central status information is used in the access decision. See "Deferred Decisions" on page 45 for details. The options are:

- Disabled - The access decision is not submitted to the host.
- When Online - The access decision is eventually submitted to the host only if it is known to be online.
- Always - The access decision is eventually submitted to the host.

NOTE

When access decision is submitted to the host for Central Status Check, the host always informs panel to deny access.

Controller Event Object List – Specifies which controller events can be activated or deactivated at this portal.

Central Status Exemption Mask – Specifies the central status exemption mask. To be exempt from the central status check, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the central status exemption mask. See "Access Decision Algorithm" on page 40 for details.

Central Status Timeout – Specifies the maximum time allowed to wait before resuming the algorithm after a decision is deferred to the host with the request for central status information. While the decision is deferred to the host, the Access Control object ignores any data received through any of its channels.

Decision Number – Indicates the number of access decisions made by the Access Control object since the controller was last started.

Denied Notification – Specifies which access denied related notifications are generated, provided their generation is not suppressed by the settings in the *Override Notification* and *Lockdown Notification* attributes. When this attribute is in the "All but Host Unreachable" state, all access denied related notifications except those caused by the host being unreachable are generated.

Direction – Specifies how the Access Control object informs its reader objects about its own access decisions as well as access decisions that are reported from its Door Sequence object. This attribute is instrumental in card-in-card-out configurations in which the access decision must only be indicated to the side from which access was requested.

The options are:

- Undefined - The Access Control object forwards all access decisions to its reader objects.
- Ingress - The Access Control object only forwards access decisions that are categorized as “Undefined” or “Ingress” to its reader objects, but not those categorized as “Egress.”
- Egress - The Access Control object only forwards access decisions that are categorized as “Undefined” or “Egress” to its reader objects, but not those categorized as “Ingress.”

Door Sequence Object – Specifies the Door Sequence object this Access Control object should use. If blank, the Access Control object cannot sign up to Door Sequence object’s *Summary* attribute to acquire the values for its *Portal Status* and *Portal Mode* attributes.

Encrypted Wiegand Facility Codes – Specifies how many of the facility codes in the *Facility Code List* attribute are exclusively to be used for Encrypted Wiegand identifiers. All these facility codes must be at the beginning of the list. This avoids using an Encrypted Wiegand facility code for a non-Encrypted Wiegand identifier, or vice versa, which could lead to an ambiguous identification. As for identifier formats other than Encrypted Wiegand specifying a facility code in the *Facility Code List* attribute is not required, the *Encrypted Wiegand Facility Codes* attribute may be set to the number of entries in the *Facility Code List* attribute.

Entity Identification – Specifies what may happen to the number of unaccompanied entities after being identified by the Access Control object. This attribute allows for an optimization of the entity processing feature.

Entity Notification – Specifies which entity related notifications are generated. See “Entity Processing” on page 35 for details. The options are:

- All - Both Entity Tracked and Unaccompanied Entity notifications are generated.
- Unaccompanied - Only Unaccompanied Entities notifications are generated.
- None - Neither Entity Tracked nor Unaccompanied Entities notifications are generated.

Entity Processing – Specifies whether an entity must have an escort at this portal, and whether access will be denied when unaccompanied entities are detected. See “Entity Processing” on page 35 for details. The options are:

- Disabled - No entity processing.
- Entity Tracking Only - An entity is tracked upon identification, and a notification is generated when the entity is not accompanied by an associated entity within a defined time.
- Deny Unaccompanied Entity - As above, but access to the portal is denied for unaccompanied entities.
- Deny All on Unaccompanied Entity - As above, but access to the portal is denied for all entities except for those with executive privilege when an unaccompanied entity is present.

Entity Timeout – Specifies the time an entity must remain in the list of unaccompanied entities to be reported as unaccompanied. The Access Control object may extend this value by 9% to handle the expiration of multiple entities in a single operation. See "Entity Processing" on page 35 for details.

Entity Trailing Time – Specifies the time an entity may trail its accompanying escort without being reported as unaccompanied. See "Entity Processing" on page 35 for details.

Error Notification – Specifies whether the Access Control object sends out extra notifications every time an error is encountered, such as missing data records or objects. These error notifications are in addition to all other notifications the Access Control object may generate.

Executive Privilege Mask – Specifies the executive privilege mask. To be exempt from most parts of the access decision algorithm an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the executive privilege mask.

Executive privilege does not overcome a validation-only situation, but *exempts* the bearer from being checked for:

- Access enabled
- Access level
- Access mask
- Access group
- Security level
- Security mask
- Portal open violation
- Unaccompanied asset and entity rules
- Incomplete group rule

See "Access Decision Algorithm" on page 40 for details.

Facility Code List – Specifies a list of facility codes that is used for pre-screening each presented badge. The identifier format interpreting the data from a presented badge must produce one of the facility codes in the list. If this list is empty, no facility code based pre-screening will be performed. As the facility code is part of the data that uniquely identifies a badge, an empty facility code list provides just as much security as listing each allowed facility code. It is therefore recommended to leave this attribute list empty, except when an Encrypted Wiegand format is used.

Fault Cause – Indicates the highest priority reason why the Access Control object is not operational. In case multiple reasons apply, the value with the lower enumeration value is given. The options are:

- None - The Access Control object works as defined.

- Out Of Memory - The controller does not have enough memory to store the most recent access decision and time stamp for each entity. When out of memory, no access decisions are made.
- General - This state is not used by the Access Control object.

Granted Notification – Specifies which access granted related notifications are generated, provided their generation is not suppressed by the settings in the *Override Notification* and *Lockdown Notification* attributes. When this attribute is set to "On Open Only," access granted related notifications are only generated when the door contact is open at any time during the access time. When set to "No Entry," an additional notification of type "No Entry" is generated if access was granted, but the door never opened during the access time or before the next entity is identified.

Group Identification – Specifies what may happen to the number of incomplete groups after being identified by the Access Control object. This attribute allows for an optimization of the group processing feature.

Group Notification – Specifies whether Incomplete Group notifications should be generated. See "Group Processing" on page 36 for details.

Group Processing – Specifies whether a portal restricts access to members of an entity group unless the entire group is registered at the portal. See "Group Processing" on page 36 for details. The options are:

- Disabled - No group processing.
- Group Tracking Only - An incomplete group is tracked when a group member is identified, and a notification is generated when the group is not complete within a defined time.
- Deny Incomplete Group - As above, but access to the portal is denied for all members of an incomplete group.
- Deny All on Incomplete Group - As above, but access to the portal is denied for all entities except for those with executive privilege when incomplete groups are present.

Group Timeout – Specifies the time in which each member of an entity group must have been identified in order for the access to be granted for the entire group. The Access Control object may extend this value by 9% to handle the expiration of multiple groups in a single operation. See "Group Processing" on page 36 for details.

Identifier Timeout – Specifies the maximum time allowed in seconds for all required identifiers to be received at an Access Control object that are to be used in a single access request. If 0 seconds are specified, there is no timeout for the gathering of identifiers.

Incomplete Groups – Indicates the number of incomplete entity groups of which their members were recently identified by the Access Control object. See "Group Processing" on page 36 for details.

Intrusion Area Object – Specifies the Intrusion Area object that represents the intrusion area that this Access Control object can grant access to. The Access Control object will deny access to the area if it is armed, and the entity requesting access does not have the right to disarm that area. The referenced object must reside on the same controller.

Keypad Display Object – Specifies the Intrusion Keypad/Display object that the Access Control object shall inform when it makes a decision of granted or granted alternate for an access profile that has intrusion rights. The referenced object must reside on the same controller.

Keypad Timeout – Specifies the maximum time allowed in seconds between two keys being pressed on a keypad. If the keypad timeout is exceeded, the entire keypad buffer is emptied. If 0 seconds are specified, there is no timeout for the keypad.

Keypad Trigger – Specifies the event that concludes the keypad entry, provided it is not filtered out by the *Primary Reader Object* and *Secondary Reader Object* attributes. The # key is always a keypad trigger, concluding the keypad entry. The options are:

- Pound Key Only - The keypad entry is considered complete only when the # key is pressed.
- Primary Channel - The keypad entry is considered complete when the # key is pressed or an identifier is received on the primary channel.
- Secondary Channel - The keypad entry is considered complete when the # key is pressed or an identifier is received on the secondary channel.
- Any Channel - The keypad entry is considered complete when the # key is pressed or an identifier is received on any channel.
- All Channels - The keypad entry is considered complete when the # key is pressed or identifiers were received on all channels.

Lockdown Notification – Specifies whether access granted, alternate access granted, and access denied related notifications are generated when the *Portal Mode* attribute is in the "Lockdown" state.

Manual Reader – Specifies whether this Access Control object is used at a manual reader. When set, access grants of this Access Control object are of category Valid, with subtype Manual Reader.

Notification Class – Specifies which Security Notification Class object should be used by the Access Control object to send notifications.

Notify Priority – Specifies the Priority parameter of all notifications generated by the Access Control object.

Number – Indicates the instance number of the Access Control object's *Object Identifier* attribute. The instance number equals the 1 based bit position in an access group's access control object mask. This attribute is provided for technical reasons only.

Occupancy Check – Specifies how the occupancy rule should be applied. See “Occupancy” on page 51 for details. The options are:

- Disabled - The occupancy rule does not have an impact on the access decision.
- Monitor Only - Occupancy rule violations are reported, but have no impact on the access decision.
- Enforce, Only When Operational - Occupancy rule violations are reported and cause access request to be denied when the occupancy decision is of type “Denied.” Occupancy decisions of type “Not Operational” or “Offline” do not cause access to be denied, but they are reported.
- Enforce, and Deny During Trouble - Occupancy rule violations are reported and cause access request to be denied when the occupancy decision is of type “Denied,” “Not Operational,” or “Offline.”

Occupancy Exemption Mask – Specifies the occupancy exemption mask. To be exempt from the access decision being subject to any occupancy rule, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the occupancy exemption mask. This mask does not prevent an entity from being counted as an occupant.

Occupancy In Object – Specifies the Occupancy object that this Access Control object should consult in making the entry access decision and inform upon an access transition, so that it can account for the entity being in the occupancy space. If the Access Control object does not handle any entities entering the occupancy space, this attribute should be blank. The referenced object must reside on the same controller.

Occupancy Out Object – Specifies the Occupancy object that this Access Control object should consult in making the exit access decision and inform upon an access transition, so that it can account for the entity having left the occupancy space. If the Access Control object does not handle any entities leaving the occupancy space, this attribute should be blank. The referenced object must reside on the same controller.

Occupancy Transition – Specifies upon which event the Access Control object informs the Occupancy object(s) that the number of entities in the area should be changed. The options are:

- Disabled - The number of entities in the area should not change on an access decision.
- On Granted - The number of entities in the area should change on an access decision of type “Granted” and “Granted Alternate.”
- On Granted Or Valid - The number of entities in the area should change on an access decision of type “Granted,” “Granted Alternate,” “Valid,” or “Silent.”
- On Identification - The number of entities in the area should change on any access decision.

Occupancy Transition Needs Open – Specifies whether the portal must be open during the access time for the number of entities in the area to change.

Override Mask – Specifies the override mask. To be eligible to issue timed overrides, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the override mask. See “Access Decision Algorithm” on page 40 for details.

Override Notification – Specifies whether access granted, alternate access granted, and access denied related notifications are generated when the *Portal Mode* attribute is in the “Override” state.

PIN Attempt Limit – Specifies after how many consecutive invalid PIN attempts a PIN Retry Alarm should be generated. To allow unlimited number of attempts, set this attribute to 0.

PIN Attempt Sector – Specifies the sector the Access Control object belongs to as far as counting invalid PIN attempts is concerned. All Access Control objects within a controller may use a common counter, or may be grouped into sectors, e.g. in case invalid PIN attempts for Intrusion Keypad/Display authentication should not count towards the invalid PIN attempts for access control.

PIN Digits – Specifies the number of PIN digits expected by the Access Control object. When an entity's PIN has fewer digits than expected by this attribute, it must be entered with the appropriate number of leading zeros. An entity's PIN that has more digits than expected by this attribute cannot be used.

PIN Duress – Specifies how a PIN can be used to generate duress alarms. The options are:

- None - A PIN cannot be used to generate a duress alarm.
- Plus One - The last digit of the PIN is increased by one to generate a duress alarm. When the last digit is a 9, it needs to be replaced with a 0.
- On Nine - Exactly one digit of the PIN needs to be replaced by a 9 to generate a duress alarm. This setting precludes any 9s in regular PINs.
- Minus One - The last digit of the PIN is decreased by one to generate a duress alarm. When the last digit is a 0, it needs to be replaced with a 9.
- First Digit - The first digit of the PIN needs to be incorrect to generate a duress alarm.
- Last Digit - The last digit of the PIN needs to be incorrect to generate a duress alarm.

PIN Suppressed – Specifies whether a PIN can be omitted in conjunction with access requests.

PIN Use – Specifies how entered PIN numbers are to be considered in the identification process. This attribute allows a single reader to provide input to two Access Control objects (one exclusively for determining intrusion rights and the other one exclusively determining access rights) while maintaining a separation of PINs used for access request and for intrusion requests. The options are:

- For Access Only - The entered PIN must match a PIN marked either for access only or for access and intrusion in the access profile.
- For Intrusion Only - The entered PIN must match a PIN marked either for intrusion only or for access and intrusion in the access profile.

- For Access and Intrusion - The entered PIN must match any PIN in the access profile.

Present Value – Indicates the current status of the Access Control object:

- Not Initialized - The Access Control object is not yet initialized. This is only an initial state and changes shortly after the Access Control object starts up.
- Operational - The Access Control object has enough memory, and all of the objects referenced by the *Door Sequence Object* attribute, the *Primary Reader Object* attribute, and the *Secondary Reader Object* attribute have *Present Value* attributes that are neither "Unknown" nor "Fault."
- Unknown - Indicates that either the object referenced by the *Door Sequence Object* attribute, the *Primary Reader Object* attribute, or the *Secondary Reader Object* attribute has a *Present Value* attribute of "Unknown," while the *Fault Cause* attribute is "None."
- Fault - Indicates that the Access Control object is not operational. The highest priority reason why the Access Control object is not operational is contained in the *Fault Cause* attribute.

Primary Reader Object – Specifies the primary reader data input channel this Access Control object works on. In the most common application, each instance of an Access Control object in a controller uses its primary channel. For some special applications that require input from two different readers, an Access Control object would use two channels. All applications requiring more than two channels need to be assembled using more than one Access Control object. The referenced object must reside on the same controller.

Reader Enabled - Specifies whether this Access Control object should make access decisions at all. If not, no incoming data is interpreted, no access decision is made, and no access decision related notifications are generated.

Secondary Reader Object – Specifies the secondary reader data input channel this Access Control object works on. For more details see the description of the *Primary Reader Object* attribute. The referenced object must reside on the same controller.

Security Level – Specifies the security level, that must be matched or surpassed by an identified access profile's security level to be granted access.

Security Mask – Specifies the security mask. To be granted access, provided security masks are used in the access decisions, an identified access profile's security mask must have at least one set bit in common with the security mask. See "Security Models" on page 39 for details.

Security Mode – Specifies whether the regular access model contributes to the access decision in the emergency mode. See "Security Models" on page 39 for details. The options are:

- Restricted - The regular access model still applies, but so does the security model. With the exception of the normal mask used in the regular access model, the access profile's security mask is used for all other mask comparisons.

- Exceptional - The regular access model does not apply, but only the security model. The access profile's security mask is used for all mask comparisons.

Security Mode Active Level – Specifies the Access Control object's security level, that must be matched or surpassed for the emergency mode to be declared. Once declared, the selected security mode becomes active, and the access profiles' security masks are used instead of the normal masks.

Set 1 Enabled – Specifies whether *Set 1...* attributes can currently be used to trigger access requests.

Set 1 Enabled Default – Specifies the default value for *Set 1 Enabled*. attribute. Changing this attribute automatically sets the *Set 1 Enabled* attribute to the same value.

Set 1 First Identifier Format – Specifies the identifier format of the primary identifier for Set 1.

Set 1 Mode – Specifies the way the *Set 1...* identification set is used. The options are:

- Regular Access - The *Set 1...* identification set is used for regular access requests.
- Validation Only - The *Set 1...* identification set is used to register people at a portal for the purpose of entity or group processing.
- Tracking Only - The *Set 1...* identification set is used to track assets or entities with readers that pick up the identifier at a long range, so the assets or entities do not actively have to present their credentials.

Set 1 PIN Required – Specifies whether a PIN is required in conjunction with Set 1 access requests.

Set 1 Second Identifier Format – Specifies the identifier format of the secondary identifier for Set 1.

Set 1 Third Identifier Format – Specifies the identifier format of the third identifier for Set 1.

Set 2 Enabled – Specifies whether *Set 2...* attributes can currently be used to trigger access requests.

Set 2 Enabled Default – Specifies the default value for the *Set 2 Enabled* attribute. Changing this attribute automatically sets the *Set 2 Enabled* attribute to the same value.

Set 2 First Identifier Format – Specifies the identifier format of the primary identifier for Set 2.

Set 2 Mode – Specifies the way the *Set 2...* identification set is used. See "Set 1 Mode" on page 22 for details.

Set 2 PIN Required – Specifies whether a PIN is required in conjunction with Set 2 access requests.

Set 2 Second Identifier Format – Specifies the identifier format of the secondary identifier for Set 2.

Set 2 Third Identifier Format – Specifies the identifier format of the third identifier for Set 2.

Set 3 Enabled – Specifies whether *Set 3...* attributes can currently be used to trigger access requests.

Set 3 Enabled Default – Specifies the default value for the *Set 3 Enabled* attribute. Changing this attribute automatically sets the *Set 3 Enabled* attribute to the same value.

Set 3 First Identifier Format – Specifies the identifier format of the primary identifier for Set 3.

Set 3 Mode – Specifies the way the *Set 3...* identification set is used. See “Set 1 Mode” on page 22 for details.

Set 3 PIN Required – Specifies whether a PIN is required in conjunction with Set 3 access requests.

Set 3 Second Identifier Format – Specifies the identifier format of the secondary identifier for Set 3.

Set 3 Third Identifier Format – Specifies the identifier format of the third identifier for Set 3.

Set 4 Enabled – Specifies whether *Set 4...* attributes can currently be used to trigger access requests.

Set 4 Enabled Default – Specifies the default value for the *Set 4 Enabled* attribute. Changing this attribute automatically sets the *Set 4 Enabled* attribute to the same value.

Set 4 First Identifier Format – Specifies the identifier format of the primary identifier for Set 4.

Set 4 Mode – Specifies the way the *Set 4...* identification set is used. See “Set 1 Mode” on page 22 for details.

Set 4 PIN Required – Specifies whether a PIN is required in conjunction with Set 4 access requests.

Set 4 Second Identifier Format – Specifies the identifier format of the secondary identifier for Set 4.

Set 4 Third Identifier Format – Specifies the identifier format of the third identifier for Set 4.

Team Exemption Mask – Specifies the team exemption mask. To be exempt from team processing, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the team exemption mask. An exempt entity still counts toward meeting the team requirements. See “Team Processing” on page 37 for details.

Team Mask – Specifies all required roles that must be qualified to pass the team processing rule. The roles are taken from the identified access profile's normal mask, or security mask in the emergency mode. See “Team Processing” on page 37 for details.

Team N-Man Rule – Specifies the minimum number of entities that must be qualified to pass the team processing rule. See “Team Processing” on page 37 for details.

Team Processing – Specifies whether the team processing feature is enabled. See section 3.5 for details.

Team Timeout – Specifies the maximum time allowed in seconds for all team members to be qualified to pass the team processing rule. See “Team Processing” on page 37 for details.

Time Zone Check – Specifies whether the Schedule objects referenced by an access profile or its associated data records are ignored in access decisions. If this flag is set, all such referenced Schedule objects are treated as if their *Present Value* was not 0. This attribute does not impact the Schedule objects themselves, or any attributes that the Schedule object writes to.

Tracking Object – Specifies the Access Control object that needs to be consulted for its number of unaccompanied assets, unaccompanied entities, and incomplete groups, and that needs to informed about entities being identified. This attribute typically references an Access Control object that handles the input from a long range tracking reader. The referenced object must reside on the same controller. See “Asset Processing” on page 33 and “Entity Processing” on page 35 for details.

Unaccompanied Assets – Indicates the number of unaccompanied assets recently identified by the Access Control object. See “Asset Processing” on page 33 for details.

Unaccompanied Entities – Indicates the number of unaccompanied entities recently identified by the Access Control object. See “Entity Processing” on page 35 for details.

Validation Only Mask – Specifies the validation only mask. To be exempt from the access decision algorithm and to be given a valid access, an identified access profile's normal mask or security mask in the emergency mode must have at least one set bit in common with the validation only mask. This setting is most useful for group members that badge only to be validated as group members. Bypassing the access decision algorithm leaves the applications of anti-passback, anti-loitering and occupancy unaffected. See “Access Decision Algorithm” on page 40 for details.

COMMANDS

This section describes commands that can be issued to this object from SCT.

Table 2: Access Control Object Commands

Command Name	Description
`PIN Suppressed Start`	Writes the *PIN Suppressed* attribute to “True.”
`PIN Suppressed Stop`	Writes the *PIN Suppressed* attribute to “False.”
`Access Enabled Start`	Writes the *Access Enabled* attribute to “True.”
`Access Enabled Stop`	Writes the *Access Enabled* attribute to “False.”
`Set 1 Enabled Start`	Writes the *Set 1 Enabled* attribute to “True.”
`Set 1 Enabled Stop`	Writes the *Set 1 Enabled* attribute to “False.”
`Set 2 Enabled Start`	Writes the *Set 2 Enabled* attribute to “True.”
`Set 2 Enabled Stop`	Writes the *Set 2 Enabled* attribute to “False.”
`Set 3 Enabled Start`	Writes the *Set 3 Enabled* attribute to “True.”
`Set 3 Enabled Stop`	Writes the *Set 3 Enabled* attribute to “False.”
`Set 4 Enabled Start`	Writes the *Set 4 Enabled* attribute to “True.”
`Set 4 Enabled Stop`	Writes the *Set 4 Enabled* attribute to “False.”
`Change Attribute`	See the description below.

The `Change Attribute` is a generic command available for writing the attributes of an object. It is mainly used to change an attribute value from those features which work only with commands. For the sole purpose of giving a generic example, there is no command defined to change the *Notify Priority* attribute of an object. `Change Attribute` could, therefore, be used to change the *Notify Priority* attribute through an interlock or multiple command, both features which require commands to be entered. The `Change Attribute` command requires two parameters:

- Attribute - This parameter specifies which attribute of the object is to be written. Only writable attributes may be changed by this command.
- New value - This parameter specifies new value to be written and must be the same data type as the attribute. The only data types allowed in this command are those allowed as command parameters. A command priority can be specified if the attribute to be changed is a prioritized attribute.

VIEWS

This section illustrates how the System Configuration Tool displays properties of the Access Control object. These screens also allow you to set the values of configurable attributes. For more information refer to the *System Configuration Tool (SCT)* manual.

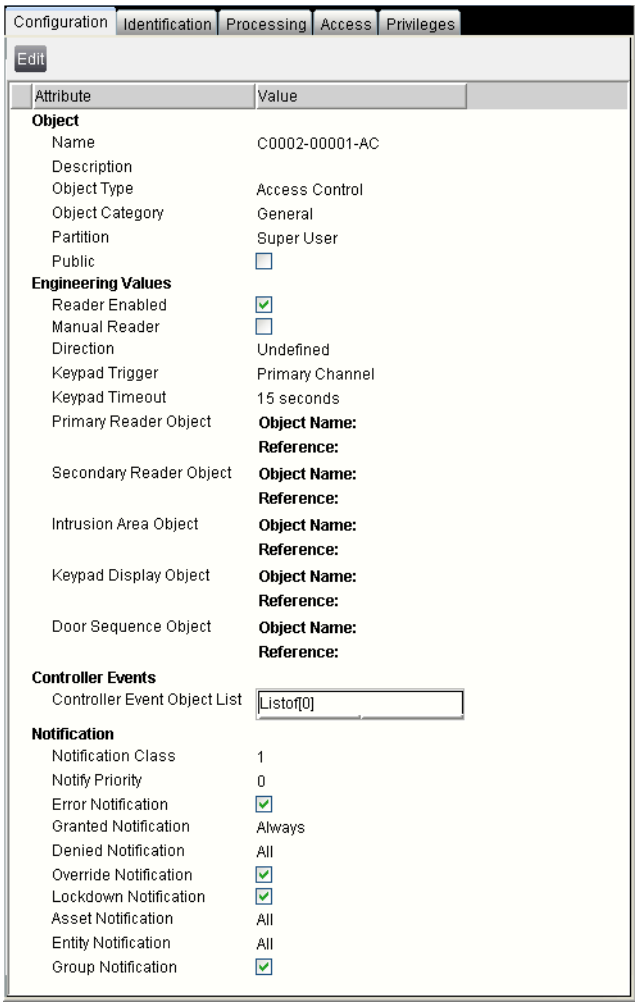


Figure 2: Configuration View

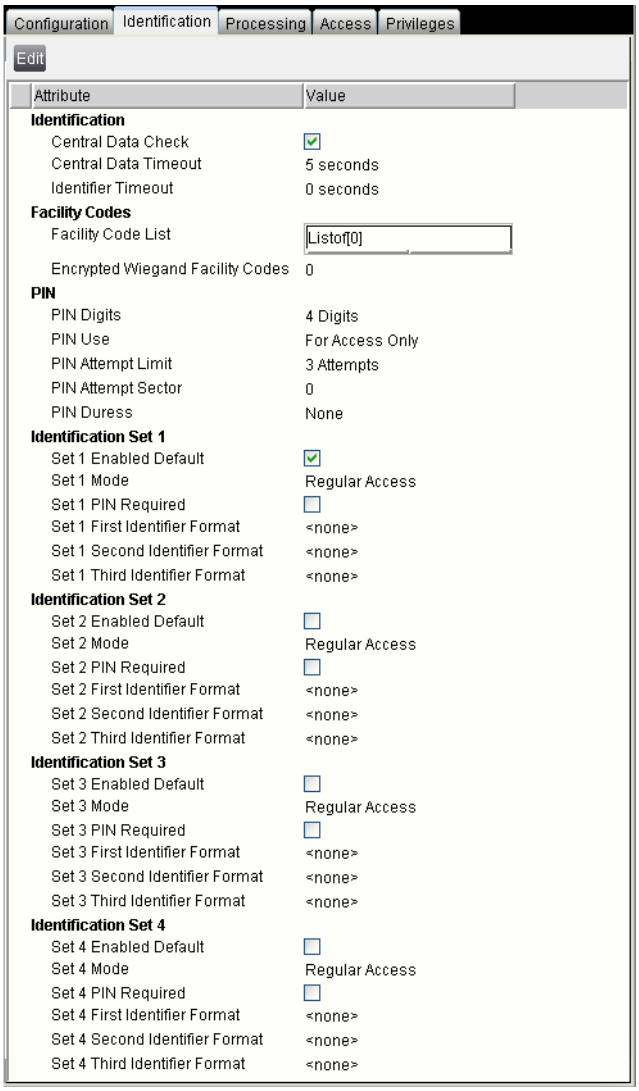


Figure 3: Identification View

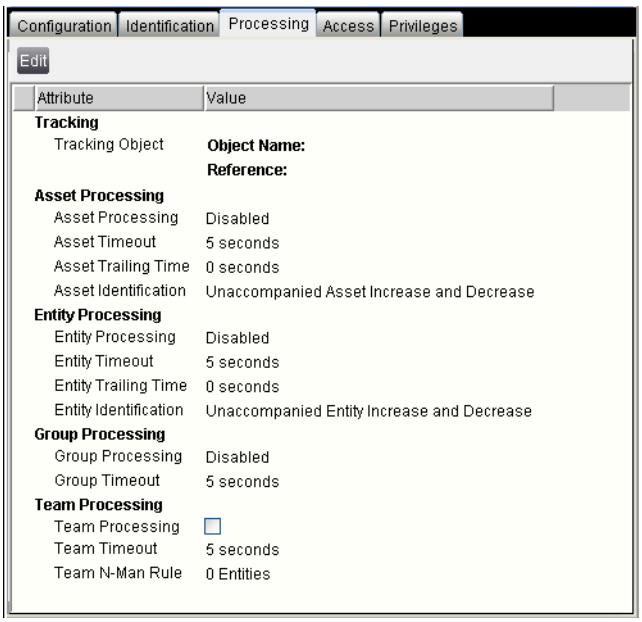


Figure 4: Processing View

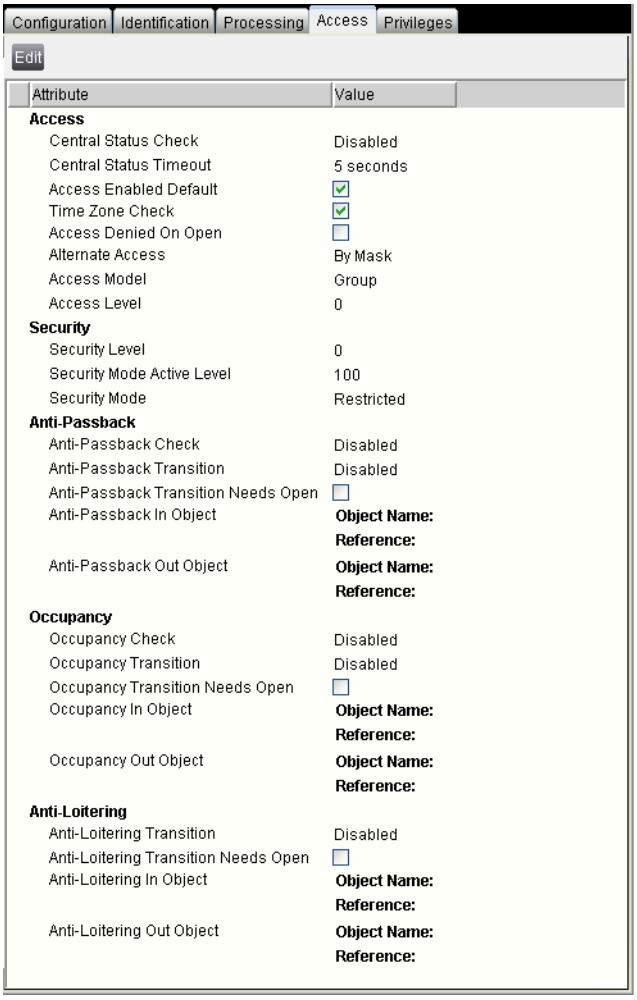


Figure 5: Access View

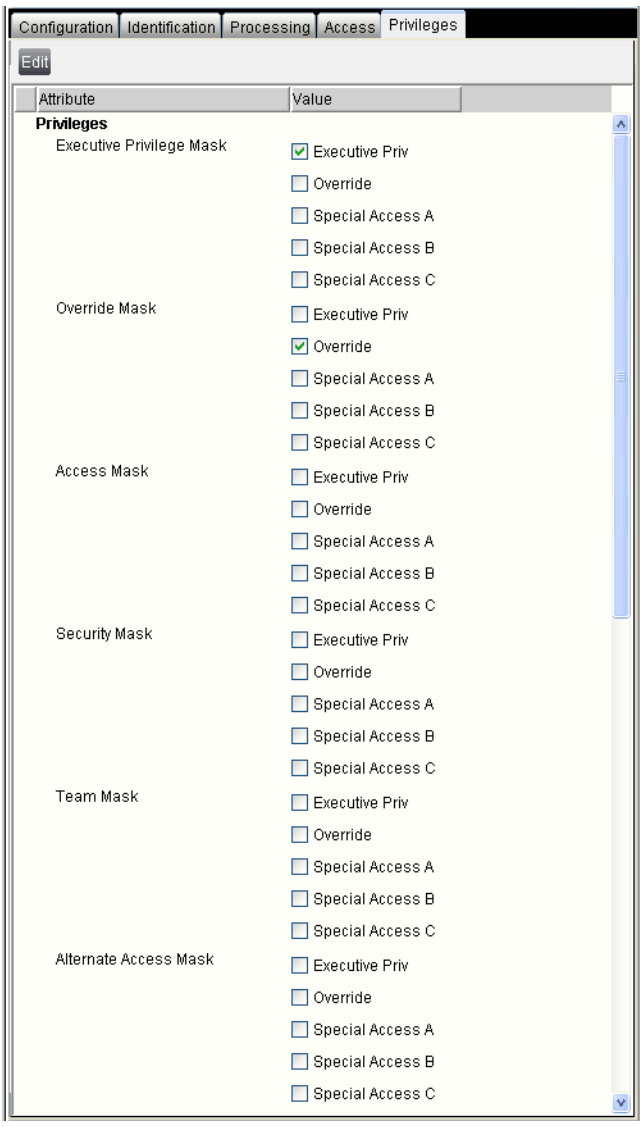


Figure 6: Privileges View - Part I

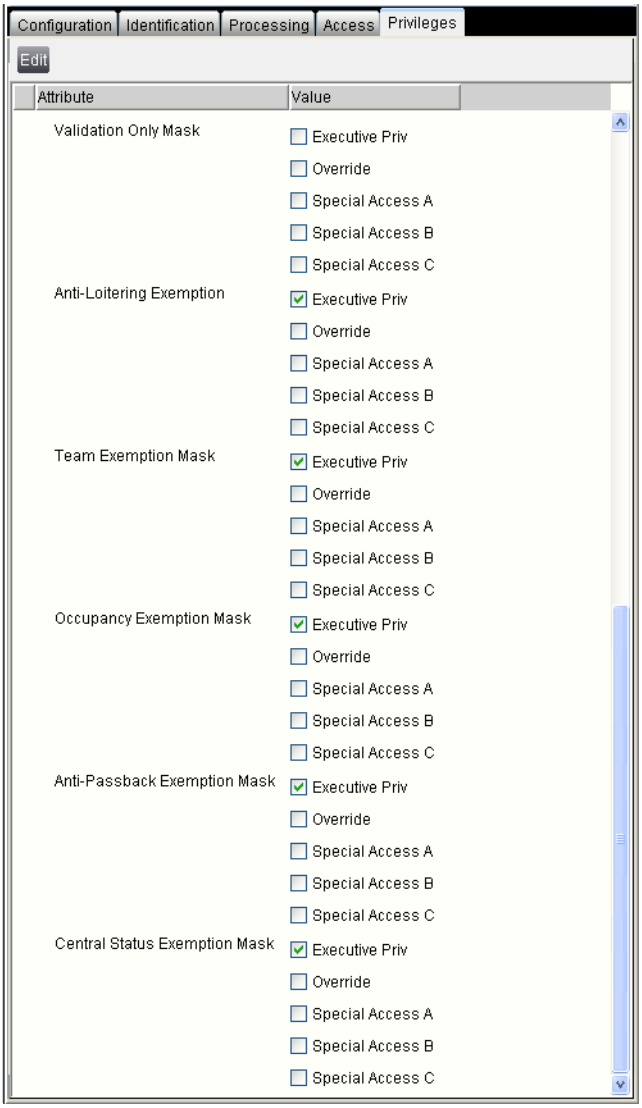


Figure 7: Privileges View - Part II

Refer to “Using Masks” on page 63 for several examples of how the privilege flags can be used.

DESCRIPTION OF OPERATION

Identification and Verification of Access Profiles and Entities

The Access Control object receives identifiers from the reader data input handlers through the primary and the secondary channels. It first filters out any identifiers received on channels not selected by the *Primary Reader Object* and *Secondary Reader Object* attributes. If the *Set 1 Enabled* flag is set, the received identifier is run through all identifier formats specified in the *Set 1...Identifier Format* attributes.

Unused identifier format attributes must be set to "None," so they don't negatively interfere with access profile identification. If an algorithm can successfully decode the received information, the Access Control object checks whether all algorithms specified in the *Set 1...Identifier* attributes have successfully interpreted at least one identifier during the timeout specified in the *Identifier Timeout* attribute. If the *Set 1 PIN Required* attribute is set, it also checks whether a PIN keypad entry has been received during the identifier timeout. If all conditions specified in the *Set 1...* attributes are satisfied, the access request is triggered. If not, the received identifier is handled by the *Set 2...*, *Set 3...* and *Set 4...* attributes accordingly.

If badges or keyed in numbers are used in access requests, their identifier formats should always be specified in the *Set...First Identifier Format* attributes. Identifier formats that produce a facility code will search the identifier database by facility code and card number. Identifier formats that do not produce a facility code will search the database for identifiers by card number only.

All identifier formats developed by the Johnson Controls Security Solution Design Center have a unique name. They include Card ID (keypad number), Common PIN, and other card formats as listed in the *CK722 Commisionning Guide*.

New identifier formats can be developed in the field. The definition capabilities allow very flexible as well as extremely precise identifier definitions, down to requiring an exact bit sequence to be received from a reader to satisfy the identifier format.

The result of the identification gathering process is a set of data that is subsequently used to search for the identifier data records in the controller's or the host's database. For most applications this set of data comprises the card number, the facility code, and the issue level.

Each identifier format is either an identifying or verifying algorithm by its nature. Identifying algorithms ultimately produce a reference to a unique identifier, whereas verifying algorithms can only be used to verify a match between the read identifier and the identifiers referenced by an already known access profile. An identification set must have at least one identifying algorithm. The Access Control object's PIN algorithm can only verify an entity, whereas the Access Control object's "Keypad Number" algorithm can identify an entity.

The entity is identified by following the entity reference stored in the access profile.

An ID set of mode “validation only” allows entities to be identified with less strict identification and verification requirements, such as just presenting a badge and not having to be verified by PIN or Biometrics. If an entity is identified under such circumstances, the Access Control object will not grant access, but, at most, make a decision of type “Valid.” This setting is useful when a group leader wants to lead a group of visitors through a door that requires verification by PIN or Biometrics, but it would be impractical to require each group member to go through that procedure. The validation only identification sets are intended to be enabled by a controller event, and disabled after a timeout, by the portal being opened, or another controller event.

An ID set of mode “tracking only” is used for long range asset or entity tracking.

While each of the four ID sets can have its mode individually set, a single Access Control object must not have a “tracking only” ID set enabled at the same time as a “validation only” or “regular access” ID set that requires more than one identifier or verifier. The reason is that each identification performed, e.g. by the “tracking only” ID set, would disrupt an ongoing identification for the multi-identifier “validation only” or “regular access” ID set.

In case an ID set of “tracking only” is required at a portal that also has a multi-identifier “validation only” or “regular access” ID set enabled, the ID set of “tracking only” shall use its own Access Control object, which is referenced by the multi-identifier Access Control object through its *Tracking Object* attribute. See “Asset Processing” on page 33 for details.

Asset Processing

An entity of type “Asset” can be configured to have one or more owners, of which at least one must accompany the asset to avoid generating an alarm or a lock-out in case the asset is detected at a portal.

In a typical application, an Access Control object is connected to a Reader object which is fed by a long range asset tag reader. When an asset is identified, it is subject to the access decision algorithm based on the type of identification set that it was identified by.

When the identification set is of type “Tracking Only,” an access decision of type “Asset Tracked” is made right after the entity and access profile expiration are checked. See “Access Decision Algorithm” on page 40 for details.

When the identification set is of type “Validation Only,” an access decision of type “Entity Validated” is made right after the entity and access profile expiration are checked. See “Access Decision Algorithm” on page 40 for details. This setting is intended for people that actively need to present credentials to identify themselves at an Access Control object, hence, it is not recommended for long range asset tracking.

When the identification set is of type “Regular Access,” the asset is subject to the entire access decision algorithm. See “Access Decision Algorithm” on page 40 for details. This is useful for raising notifications when an asset is seen at an Access Control object where it should not be.

When the *Asset Processing* attribute is set to “Disabled,” no further asset processing is performed, that is, an asset is not added to the list of unaccompanied assets, and no Unaccompanied Asset notifications are generated.

When the *Asset Processing* attribute is set to a value other than disabled, and an asset is identified, its owning entities are determined. An asset can be owned by any entity that is referenced by the asset via a “must be accompanied by” association. If the owning entity is an entity group, only one member of the entity group needs to have a recent identification for the asset to be accompanied. In case none of the associated entities have a recorded identification time stamp for this processing sector, that goes back fewer than the number of seconds specified in the *Asset Trailing Time* attribute, the entity is added to a list of unaccompanied assets, and the *Unaccompanied Assets* attribute is increased by one. As long as the *Unaccompanied Assets* attribute is greater than zero, all subsequent identifications trigger all unaccompanied assets in the list to be re-checked for their owning entities. When the *Asset Processing* attribute is set to “Deny on Unaccompanied Assets,” all access requests are denied, unless the requestor has executive privilege. See “Access Decision Algorithm” on page 40 for details.

The *Unaccompanied Assets* attribute is decreased by one, and the asset is removed from the list when an owning entity is identified.

The *Unaccompanied Assets* attribute is also decreased by one, and the asset is removed from the list when an owning entity is not found, but the asset's last identification time stamp dates back more than or equal the number of seconds specified in the *Asset Timeout* attribute. In this case, an Unaccompanied Asset notification is generated by the Access Control object that owns the tracking sector, provided that the *Asset Notification* attribute is not set to “None.”

When assets tags are identified by a long range reader, but the owners are identified by actively presenting their credentials, it is strongly recommended to use one Access Control object for the assets and another one for the owners. This allows for continuous long range tracking of assets, even when the other Access Control object is in a hold pattern because it is waiting for more input from the entity requesting access or is waiting for data to be provided by the host. This procedure also allows to secure multiple adjacent doors or turnstiles with individual Access Control objects, but a single long range asset tag reader. To accomplish these solutions, the Access Control object that performs the long range tracking is referenced by the other Access Control objects through their *Tracking Object* attributes. When this attribute is set, an Access Control object not only considers its own lists of unaccompanied assets, but also the one maintained by the referenced Access Control object. This also means that an Access Control object can contribute identified owners to the referenced Access Control object.

Entity Processing

An entity of type different than “Asset” can be configured to have one or more associated entities, with any one of these associated entities being able to serve as an escort for that entity. However, the escort requirement may not apply at all portals or at all times. Therefore the Access Control object provides the *Entity Processing* attribute. As a writable attribute, this attribute may be scheduled or changed by an external algorithm.

When the identification set is of type “Tracking Only,” an access decision of type “Entity Tracked” is made right after the entity and access profile expiration are checked. See “Access Decision Algorithm” on page 40 for details. This setting is most useful for identifying entities through a long range reader, whereas regular entities actively need to present credentials to identify themselves.

When the identification set is of type “Validation Only,” an access decision of type “Entity Validated” is made right after the entity and access profile expiration are checked. See “Access Decision Algorithm” on page 40 for details. This setting is intended for people that actively need to present credentials to identify themselves at an Access Control object, hence, it is not recommended for long range entity tracking.

When the identification set is of type “Regular Access,” the entity is subject to the entire access decision algorithm. See “Access Decision Algorithm” on page 40 for details.

When the *Entity Processing* attribute is set to “Disabled,” no special entity processing is performed, that is, an entity is not added to the list of unaccompanied entities, and no Unaccompanied Entity notifications are generated.

When the *Entity Processing* attribute is set to a value other than disabled, and an entity is identified, its associated entities are determined. An entity can be escorted by any entity that is referenced by the entity via a “must be accompanied by” association. If the escorting entity is an entity group, only one member of the entity group needs to have a recent identification to be an escort. In case none of the associated entities have a recorded identification time stamp for this processing sector, that goes back fewer than the number of seconds specified in the *Entity Trailing Time* attribute, the entity is added to a list of unaccompanied entities, and the *Unaccompanied Entities* attribute is increased by one.

When the *Entity Processing* attribute is set to “Deny Unaccompanied Entities,” and the entity requesting access is in fact unaccompanied, the access decision will be denied, unless the requestor has executive privilege. See “Access Decision Algorithm” on page 40 for details.

As long as the *Unaccompanied Entities* attribute is greater than zero, all subsequent identifications trigger all unaccompanied entities in the list to be re-checked for their associated entities. When the *Entity Processing* attribute is set to “Deny All on Unaccompanied Entities,” all access requests are denied, unless the requestor has executive privilege. See “Access Decision Algorithm” on page 40 for details.

The *Unaccompanied Entities* attribute is decreased by one, and the entity is removed from the list when an associated entity is identified at the Access Control object.

The *Unaccompanied Entities* attribute is also decreased by one, and the entity is removed from the list when an associated entity is not found, but the entity's time stamp dates back more than or equal the number of seconds specified in the *Entity Timeout* attribute. In this case, an Unaccompanied Entity notification is generated, provided that the *Entity Notification* attribute is not set to "None."

When entities that need to be escorted are identified by a long range reader, but the escorts are identified by actively presenting their credentials, it is strongly recommended to use one Access Control object for the escortees and another one for the escorts. This allows for continuous long range tracking of escortees, even when the other Access Control object is in a hold pattern because it is waiting for more input from the escort requesting access or is waiting for data to be provided by the host. This procedure also allows to secure multiple adjacent doors or turnstiles with individual Access Control objects, but a single long range entity tag reader. To accomplish these solutions, the Access Control object that performs the long range tracking is referenced by the other Access Control objects through their *Tracking Object* attributes. When this attribute is set, an Access Control object not only considers its own lists of unaccompanied entities, but also the one maintained by the referenced Access Control object. This also means that an Access Control object can contribute identified owners to the referenced Access Control object.

Group Processing

An entity of any type can be configured to be a member of an entity group. In this function, the entity can only be granted access when all members of the entity group are present at a portal. However, this requirement may not apply at all portals or at all times. To optimize the processing of access decisions when group processing is not required, the Access Control object provides the *Group Processing* attribute. As a writable attribute, this attribute may be scheduled or changed by an external algorithm.

When the *Group Processing* attribute is set to "Disabled," the group processing is not enabled at this portal, and each access decision is made as if the identified entity was not a member of any group.

When the *Group Processing* attribute is set to a value other than disabled, and an entity with an entity group association of type "Is Member of" is identified, that entity group is added to the list of incomplete groups whenever not all of the group members have a recorded identification time stamp for this Access Control object, that goes back fewer than the number of seconds specified in the *Group Timeout* attribute, and the *Incomplete Groups* attribute is increased by one. An empty entity group is not considered an incomplete group, as none of its members are missing.

When the *Group Processing* attribute is set to "Deny Incomplete Groups" or "Deny All on Incomplete Groups," and the entity requesting access references an incomplete or an empty entity group, the access decision will be denied, unless the requestor has executive privilege. See "Access Decision Algorithm" on page 40 for details.

As long as the *Incomplete Groups* attribute is greater than zero, all subsequent access grants trigger all incomplete groups in the list to be re-checked for expiration. When the *Group Processing* attribute is set to "Deny All on Incomplete Groups," all access requests are denied, unless the requestor has executive privilege. See "Access Decision Algorithm" on page 40 for details.

The *Incomplete Groups* attribute is decreased by one and the entity group is removed from the list when all of its members have been identified at this Access Control object at a time that goes back fewer than the number of seconds specified in the *Group Timeout* attribute. In this case the last access request that completes the group determines the access rights.

The *Incomplete Groups* attribute is also decreased by one, and the entity group is removed from the list when all of its members' time stamps date back more than or equal the number of seconds specified in the *Group Timeout* attribute. In this case, an Incomplete Group notification is generated, provided that the *Group Notification* attribute is set to "True."

Team Processing

The Access Control object can require more than one person to be present within a specified time in order to open the door. This requirement can be based on the number of persons, on their roles, or on a combination thereof.

Other than the Asset, Entity, or Group Processing features, which require an entity to be merely identified, the Team Processing feature requires an entity to be qualified within the specified timeout. See "Access Decision Algorithm" on page 40 for details.

As long as all team processing rules are not fully satisfied, all qualified team members are given a Team Member Validated decision. The team member which first satisfies all team processing rules is given an access grant. To prevent anti-tailgating, the moment that all team processing rules are satisfied, the accumulated number of persons and their roles are reset.

Whenever the identified access profile's normal mask, or security mask in the emergency mode, has a set bit in common with the *Team Exemption Mask* attribute, the entity is exempt from being given only a Team Member Validated decision by the team processing rule, but is given an access grant instead. Such an entity does not negatively interfere in accumulating the number of persons and their roles, but may actually contribute towards satisfying all team processing rules.

N-Man Rule Team Processing

In order to be granted access, at least as many different entities as specified in the *Team N-Man Rule* attribute must be qualified within the time specified by the *Team Timeout* attribute. If the *Team Processing* attribute is set to "False" or the *Team N-Man Rule* attribute is set to 0, this mode of team processing is disabled.

The member of the last team satisfying the team processing rule is given an access grant, all other previous team members are given a Team Member Validated decision.

Role Based Team Processing

In order to be granted access, the combination of all normal masks (or security masks in the emergency mode) that were qualified during the team timeout must fully match or exceed the team mask.

(The masks are defined in "Access Mask Model" on page 39, the team timeout is defined by the *Team Timeout* attribute, and the team mask is defined by the *Team Mask* attribute.)

These combinations allow for up to 100 different roles to be present. If any one of the *Team Processing* attribute or the *Team Timeout* attribute is set to "False," or the *Team Mask* attribute has no bits set, this mode of team processing is disabled.

Combinations

When both the *Team N-Man Rule* attribute and the *Team Mask* attribute are different from zero, both team processing conditions have to be fulfilled, that is, for access to be granted, all required roles and the minimum number of different entities have to be qualified.

Access Models

The ultimate decision whether or not access is granted at a portal can be the result of a complex logic process, which may involve the evaluation of permanent as well as temporary conditions.

Permanent conditions include configuration settings for entities, access profiles, access groups, and the Access Control object's configuration attributes. Even though these conditions may be modified over time, or be combined with time or time-of-day dependent conditions, they are generally considered permanent, because they do not involve random events or frequent, temporary configuration changes.

Temporary conditions are typically influenced by random events or user interaction. Although there is no necessity to divide conditions up into either category, it helps to do so to understand the purpose of certain features offered in the access control system.

Access Group Model

In the access group model, access is denied if the Access Control object's portal is not included in the access profile's access group, or if the Access Control object's portal is included in the access profile's access group, but the associated Schedule object's *Present Value* attribute is 0. This fulfills the requirement for precision access, even if one access group needs to be created per access profile.

Although this mechanism is very powerful and flexible, it may not meet the requirement to quickly set-up some other basic access logic schemes. To accommodate these needs, two parallel and independent access models are available for basic portal access right determination.

Access Mask Model

In the access mask model, access is denied if the access profile's normal mask does not share at least one set bit with the Access Control object's *Access Mask* attribute. An access mask can be thought of a set of functional categories or roles, such as "security," "custodial," "IT personnel," etc. This simplistic model does not allow precision access, but allows easy configuration of role-based access rights.

Access Level Model

In the access level model, access is denied if the access profile's access level is lower than the Access Control object's *Access Level* attribute. This simplistic model does not allow precision access, but allows easy configuration of different zones of access rights.

Combining Access Models

All three permanent access models are independent access models. To add a powerful level of access sophistication, it is possible to combine these three models in any combination at any portal, which results in 18 different combinations, definable through the Access Control object's *Access Model* attribute.

Security Models

The Access Control object enters its emergency mode, when the value of the *Security Level* attribute meets or exceeds the value of the *Security Mode Active Level* attribute. For access not to be denied in emergency mode, the access profile's security mask must share at least one bit with the Access Control object's *Security Mask* attribute.

When the *Security Mode* attribute is set to "Restricted," entering the emergency mode adds the comparison between the Access Control object's *Security Mask* attribute and the access profile's security mask to the currently used access model. This allows a rapid restriction of access rights in the emergency mode without having to change the configuration of access profiles or Access Control objects.

When the *Security Mode* attribute is set to "Exceptional," entering the emergency mode uses the comparison between the Access Control object's *Security Mask* attribute and the access profile's security mask instead of the currently used access model. This allows a rapid restriction, but also a broadening of access rights in the emergency mode without having to change the configuration of access profiles or Access Control objects.

In emergency mode, the access profile's security mask is used for the comparisons with the following Access Control object's attributes:

- *Executive Privilege Mask*
- *Override Mask*
- *Team Mask*
- *Alternate Access Mask*
- *Validation Only Mask*
- *Anti-Loitering Exemption Mask*
- *Team Exemption Mask*
- *Occupancy Exemption Mask*
- *Anti-Passback Exemption Mask*
- *Central Status Exemption Mask*

It is also used for the comparison with the following Controller Event object's attribute:

- *Trigger Mask*

Independent of whether the Access Control object is in the emergency mode, the access profile's security level is always compared against the *Security Level* attribute, except when the access profile has executive privilege.

Access Decision Algorithm

The access decision algorithm is invoked when the gathering of the minimum set of required identifiers is complete, and ends when an access decision has been made.

The Access Control object produces access decisions that can be divided into the following categories:

- Unidentified
- Deferred
- Denied
- Silent
- Valid
- Granted
- Granted alternate

In the following description, each possible access decision is underlined, indicating that the access decision algorithm ends there. The category of each access decision is shown in parentheses.

If the access decision is deferred to the host, all output of the identifier formats is sent to the host. Although the algorithm specified in this section applies only to controllers, it may be defined identically for the host.

1. Entry Point

```
If duress condition is determined by identifier formats
    Mark duress

If biometric mismatch is determined
    Biometric Mismatch (Unidentified)

If expiration date on the credential has passed
    Credential Expired (Unidentified)

If facility code is determined but is not allowed at this portal
    Invalid Facility Code (Unidentified)
```

2. Entry Point

```
Search database for all identifiers involved in the access request
    If smart card signature is found but incorrect
        Invalid Smart Card Signature (Unidentified)
    If badge type does not match (i.e. real badge vs. keypad number)
        Invalid Identifier (Unidentified)
    If badge is found, but issue level is incorrect
        Invalid Issue Level (Unidentified)
If found identifiers point to different access profiles
    Inconsistent Identifiers (Unidentified)

If identifier or access profile cannot be found
    If no central data check or 2nd try after central data check
        If identifier cannot be found
            Invalid Identifier (Unidentified)
        Invalid Access Profile (Unidentified)
    Request Central Data (Deferred)
    Case decision timeout expired
        Central Data Unreachable (Unidentified)
    Case host sent data
        Rerun algorithm for 2nd try from 2nd entry point

If entity cannot be found
    Invalid Entity (Unidentified)

If time is not between access profile's start and end time
    Access Rights Expired (Denied)

If time is not between entity's start and end time
    Entity Expired (Denied)
```

ENTITY IS IDENTIFIED. (ENTITY CAN NOW ACCOMPANY ENTITIES AND COMPLETE GROUPS)

```
Update Unaccompanied_Assets attribute locally and at Tracking_Object
Update Unaccompanied_Entities attribute locally and at Tracking_Object
Update Incomplete_Groups attribute locally and at Tracking_Object

If identified by tracking-only ID set
    If entity is an asset
        Asset Tracked (Silent)
    Entity Tracked (Silent)

If identified by validation-only ID set or entity meets validation only mask
    Entity Validated (Valid)

If PIN was produced by identifier format
    If PIN duress
        Mark duress
    If PIN incorrect
        If maximum PIN retries exceeded
            PIN Retries Exceeded (Denied)
        Invalid PIN (Denied)

If access profile has no executive privilege
    If access is not enabled or access model is set to None
        Invalid Reader (Denied)

    If access level is part of required access model
        Determine access level access rights
        If access level is required but doesn't grant access
            Invalid Access Level (Denied)

    If access mask is part of required access model
        Determine access mask access rights
        If access mask is required but doesn't grant access
            Invalid Access Mask (Denied)

    If access groups are part of required access model
        Determine access group access rights
        If no access groups lists this portal
            Invalid Access Group (Denied)
        If time is not between access group's start and end time
            Access Group Expired (Denied)
        If access groups don't grant access based on time zone
            Invalid Access Group Timezone (Denied)

    If security level doesn't grant access
        Invalid Security Level (Denied)

    If in emergency mode and security mask doesn't grant access
        Invalid Security Mask (Denied)

    If access denied on portal open, portal is open and no duress is marked
        Portal Open Violation (Denied)
```

```
If access profile is not exempt from anti-passback check
    If anti-passback's time rule is enforced and is violated
        Anti-Passback Violation (Denied)

    If anti-passback's entry-exit rule is enforced and is violated
        Entry-Exit Violation (Denied)

If central anti-passback decision requested or central status check required
    Request Central Status (Deferred)*

Wait until all off-box status is in or timed out
    If host did not reply before timeout
        Central Status Unreachable (Denied)
If host denies access
        Central Access Denied (Denied)
If anti-passback rules are enforced and timed out
        Anti-Passback Violation (Denied)
If anti-passback's time rule is enforced and is violated
        Anti-Passback Violation (Denied)
If anti-passback's entry-exit rule is enforced and is violated
        Entry-Exit Violation (Denied)
```

3. Entry Point

```
*If decision was deferred because of anti-passback or central status check
    If access profile has no executive privilege
        If access is not enabled or access model is set to None
            Invalid Reader (Denied)

        If access level is part of required access model
            Determine access level access rights
            If access level is required but doesn't grant access
                Invalid Access Level (Denied)

        If access mask is part of required access model
            Determine access mask access rights
            If access mask is required but doesn't grant access
                Invalid Access Mask (Denied)

        If access groups are part of required access model
            Determine access group access rights
            If no access groups lists this portal
                Invalid Access Group (Denied)
            If time is not between access group's start and end time
                Access Group Expired (Denied)
            If access groups don't grant access based on time zone
                Invalid Access Group Timezone (Denied)

        If security level doesn't grant access
            Invalid Security Level (Denied)

        If in emergency mode and security mask doesn't grant access
            Invalid Security Mask (Denied)
```

```
        If access denied on portal open, portal is open and
        no duress marked
            Portal Open Violation (Denied)

If area is armed and entity does not have disarm rights
    Intrusion Area Armed (Denied)

If access profile is not exempt from occupancy check
    If occupancy rule is enforced and is violated
        Occupancy Violation (Denied)

If access profile has no executive privilege
    If unaccompanied assets are seen but are not allowed
        Unaccompanied Asset Rule (Denied)

    If entity is an unaccompanied entity and is not allowed
    or unaccompanied entities are seen but are not allowed
        Unaccompanied Entity Rule (Denied)

    If entity is a member of an incomplete or empty group and is not allowed
    or incomplete groups are seen but are not allowed
        Incomplete Group Rule (Denied)

If timed override sequence is keyed in
    If entered override time is invalid for this Access Control Object
        Invalid Override Time (Denied)
    If access profile does not have override rights
        Invalid Override Privilege (Denied)
    If entered override time is not allowed at the Door Sequence Object
        Invalid Override Time (Denied)

If manual controller event sequence is keyed in
    If event code is invalid for this Access Control Object
        Invalid Event (Denied)
    If access profile does not have correct event privilege
        Invalid Event Privilege (Denied)
    If controller event is locked
        Controller Event Trigger Locked (Denied)

ENTITY IS QUALIFIED. DECISION WILL BE POSITIVE FROM THIS POINT ON.

If access profile is not exempt from team check
    If team members from all required teams were not qualified recently
    or not enough entities were recently qualified
        Team Member Validated (Valid)

If manual controller event sequence is keyed in (continued)
    Inform Controller Event Object about access decision
        Controller Event Object impacts its referenced object
    If no door strike unlocking desired
        Manual Controller Event (Valid)
```

```
If manual reader
     Manual Reader (Valid)

If Door Sequence Object indicates a lockdown condition
     Entity Validated (Valid)

If alternate access is requested
     Alternate Access Granted (Granted Alternate)

Access_Granted (Granted)
```

Once the access decision is made, and the access decision algorithm has ended, the following access wrap up sequence is executed.

Wrap Up

```
If automatic controller events apply
     Inform Controller Event Objects about access decision
          Controller Event Objects impact their referenced objects

If anti-passback transition occurred
     Inform Anti-Passback Objects
If occupancy transition occurred
     Inform Occupancy Objects
If anti-loitering transition occurred & entity is not exempt from
anti-loitering
     Inform Anti-Loitering Objects

If entity is granted (alternate) access
     Inform Keypad Display Object about access decision
     Inform Elevator or Cabinet Object about access decision
          Elevator or Cabinet Object impacts its referenced objects

Update all access decision related attributes
Inform Door Sequence Object about access decision
     Door Sequence Object impacts its referenced objects
Generate notification according to access decision
```

Deferred Decisions

The entire access request process may experience delays because configuration data or status data needs to be acquired from external sources before proceeding. The Access Control object's *Central Data Timeout* and *Central Status Timeout* attributes specify the maximum time in which the access request process must be concluded. During these times the Access Control object waits for the required data to come in, it ignores all input from any of its channels. This is necessary to avoid collision of multiple access requests.

Central Data Requests

Central data requests are made when the *Central Data Check* attribute is set, and not all the configuration data is present at the controller to make an access decision. The controller is expected to make the decision the next time the same access request is made. Central data is requested by sending a Central Data Request notification to the host.

When an access decision is deferred to the host because of a central data request, the Access Control object will wait for the time specified in the *Central Data Timeout* attribute for the host to download all required data. When the host has done so, it then instructs the Access Control object to resume its operation.

The Access Control object then runs the entire access decision algorithm as described in “Access Decision Algorithm” on page 40. In this second execution the decision cannot be deferred anymore, and a decision will be made locally based on the data in the controller at that time.

The following data records need to be present for the Access Control object to make a local decision.

- Access Profile record
- All Identifier Data records applicable to this Access Control object
- All Identifier Access Group Data records applicable to this Access Control object
- Entity record, if specified by the Access Profile record
- All Entity Group records specified by the Entity record

When the time specified in the *Central Data Timeout* attribute has elapsed before the Execute attribute was written to “True,” the access decision algorithm produces a Host Unreachable for Central Data decision.

Central Status Requests

Independent of the *Central Data Check* attribute, certain features may require central status information to be checked before a decision can be made. Central status can be requested not only from the host, but also from peer controllers that provide anti-passback status. Central status can be requested not only from the host, but also from peer controllers that provide anti-passback status. Central status is requested from the host by sending a Central Status Request notification.

Whenever the identified access profile's normal mask, or security mask in emergency mode, has a set bit in common with the *Central Status Exemption Mask* attribute, the entity is exempt from requesting central status from the host.

When an access decision is deferred because of a central status request, the Access Control object will wait for the time specified in the *Central Status Timeout* attribute for all requested status to be provided.

When all requested status information is in, the access decision is resumed from where it left off. When the *Central Status* attribute is set to "Denied," the access decision algorithm immediately ends in a Central Access Denied decision. Similarly, when any other status information indicates a denied access decision, the decision algorithm ends immediately, and subsequently incoming central status is ignored.

The table below specifies how the *Central Status Check* attribute affects the access decision based on the central status.

Central status decision	*Central Status Check*		
	Disabled	**When online**	**Always**
OK (access request passed host's status check)	Continue	Continue	Continue
Deferred (host is known to be online and access request is pending)	Continue	Wait	Wait
Denied (access request failed host's status check)	Continue	Deny	Deny
Offline (host is known to be offline or central status request timed out)	Continue	Continue	Deny

When the time specified in the *Central Status Timeout* attribute has elapsed before the *Central Status was received*, the access decision algorithm produces a Host Unreachable for Central Status decision when the *Central Status Check* is set to "Always." If set to "When Online," and the host cannot be reached, the access decision algorithm continues.

Once the access decision algorithm has ended, the access wrap up sequence is executed as described in "Access Decision Algorithm" on page 40.

Timed Overrides

A portal may requested to be temporarily overridden into the "Unlocked" state by pressing ***0**, followed by the number of minutes the portal shall be unlocked.

The number of minutes can range between 0 and 1440, in which 0 cancels an existing timed override. The Access Control object generates an Invalid Override Time notification if either of the following occurs:

- The number of minutes exceeds 1440
- The Door Sequence object does not allow timed overrides
- The access profile does not have the Override Enable attribute set

If the entity's access profile's normal mask or security mask in emergency mode does not have at least one set bit in common with the Access Control object's *Override Mask* attribute, the Access Control object generates an Invalid Override Privilege notification.

Controller Event Triggers

Overview

The Access Control object is solely responsible for triggering controller events.

The *Trigger Type* attribute of the Controller Event object specifies under which conditions the controller event should be executed. The choices are:

- Always on Unidentified - The controller event is triggered automatically every time an access decision of type "Unidentified" is made, independent of any event privilege considerations.
- Always on Positive Decision - The controller event is triggered automatically every time a positive access decision is made (provided that event privilege of the access profile equals or exceeds event privilege of the controller event).
- Also Require Positive Decision - The controller event is triggered automatically by a mask or manually by an event code, but only when the access decision is positive (provided that event privilege of the access profile equals or exceeds event privilege of the controller event).
- Do Not Require Positive Decision - The controller event is triggered automatically by a mask or manually by an event code, independent of the access decision (provided that event privilege of the access profile equals or exceeds event privilege of the controller event).
- Always On Negative Decision - The controller event is triggered automatically every time a negative access decision is made, independent of any event privilege considerations

Positive decisions are defined as decisions of category granted, granted alternate, valid and silent.

Negative decisions for automatic triggering are defined as decisions of category denied and unidentified.

For automatic events to be triggered by mask, the person's access profile's normal mask, or security mask in the emergency mode, must have at least one set bit in common with the controller event's trigger mask. If triggering by mask is not desired, the trigger mask must be set to 0.

For manual events to be triggered, the person must enter the controller event's event code at the keypad. If manual event triggering is not desired, the event code must be set to 0. See "Manual Controller Event Triggers" on page 49 for details.

At the end of every access decision, the Access Control object checks all controller event triggers to see if any controller events need to be triggered. All controller event triggers other than "Always on Unidentified" and "Always On Negative Decision" require the *Event Privilege* value contained in the access profile to be equal to or greater than their own *Event Privilege* value.

Controller events are triggered by the Access Control object during the access decision algorithm as described in section 3.8. Controller Event objects write their targeted object's attribute immediately upon triggering. This means that controller events can be used to change any targeted objects' attributes before the access decision algorithm finishes. This allows for powerful logic applications, but also requires the logic's designer to be familiar with the access decision algorithm as described on page 40

Manual Controller Event Triggers

Manual activation of a controller event is done by pressing ***1**, followed by the controller event's panel code.

Manual deactivation of a controller event is done by pressing ***4**, followed by the controller event's keypad code.

If an event code entered at a keypad does not exist in any Controller Event object defined in the *Controller Event Object List* attribute, an “Invalid Event” decision is made.

If an event code entered at a keypad does exist in any Controller Event object defined in the *Controller Event Object List* attribute, but the *Event Privilege* value contained in the access profile does not match or exceed the Controller Event object's *Event Privilege* attribute, the Access Control object makes an “Invalid Event Privilege” decision.

If the Controller Event Object has its *Trigger Lock* attribute set to “True,” the Access Control object makes a “Controller Event Trigger Locked” decision.

Manual controller event triggers need to let the user at the keypad reader know whether or not the intended action was in fact accomplished. Whenever the intended action was not executed, for whatever reason, the Access Control object will tell the Door Sequence object to initiate the “access denied” sequence. See the *Door Sequence Object* document for details.

A user who is activating or deactivating a manual controller event, may not want to unlock the door strike, which would be the normal consequence of the access grant. Manual controller event triggers therefore have a that causes the Access Control object not to tell the Door Sequence object to operate the door strike unlocking sequence.

Common PINs

An access request may be made at a portal by pressing ***2**, followed by an identification number of up to 19 digits, followed by the **#** key. The entered identification number is then used to search the badge database for Common PINs. The badge record type must allow being used as “keypad number.” Otherwise access is denied per Invalid Identifier. This prevents holders of real badges to misuse the Common PIN function to request access.

Anti-Passback

The Anti-Passback object enables both security applications commonly known as anti-passback and entry-exit.

By consulting an Anti-Passback object, the Access Control object can prevent an entity from repeatedly entering an area if it has not waited a configurable amount of time since the last entering, or prevent an entity from repeatedly exiting an area if it has not waited a configurable amount of time since the last exiting. The Access Control object allows the option of merely reporting without enforcing any violations of the anti-passback rule.

Also, by consulting Anti-Passback objects, the Access Control object can prevent an entity from exiting an area without previously having properly entered the area, and can prevent an entity from entering an area without previously having properly left the area. This mode is also referred to as entry-exit mode. The Access Control object allows the option of merely reporting, but not enforcing any violations of the entry-exit rule.

By design of the Anti-Passback object, an anti-passback application can involve several Access Control objects on several controllers. Also, a single controller can host several anti-passback applications.

The anti-passback application is applied by setting the *Anti-Passback Check* attribute to either “Monitor Only,” “Enforce Only When Operational,” or “Enforce, and Deny During Trouble” states. An anti-passback decision other than “OK” is reported to the host by including the *Anti-Passback In Status* or *Anti-Passback Out Status* attributes in the notification.

Whenever the identified access profile’s normal mask, or security mask in the emergency mode, has a set bit in common with the *Anti-Passback Exemption Mask* attribute, the entity is exempt from being checked against the anti-passback rules. In this case the *Anti-Passback In Status* and *Anti-Passback Out Status* attributes are set to “OK.”

The Access Control object requests entry or exit for an entity at on-box Anti-Passback objects, which immediately return either a final decision, or a status of deferred. A deferral occurs when the local Anti-Passback object is configured to go off-box to obtain the ultimate anti-passback decision for that area from another Anti-Passback object. In this case, the Access Control object will defer the access decision for the time specified in the *Central Status Timeout* attribute. The returned decisions are reflected in the *Anti-Passback In Status* and *Anti-Passback Out Status* attributes.

The table below specifies how *Anti-Passback Check* attribute affects the access decision based on the anti-passback decision:

Anti-Passback decision	*Anti-Passback Check*			
	Disabled	**Monitor Only**	**Enforce, Only When Operational**	**Enforce, and Deny During Trouble**
OK	Continue	Continue	Continue	Continue
Deferred	Continue	Wait	Wait	Wait
Denied (entry-exit violation)	Continue	Report and deny	Report and deny	Report and deny
Offline	Continue	Report and continue	Report and continue	Report and deny
Not operational	Continue	Report and continue	Report and continue	Report and deny
Undefined status	Continue	Report and continue	Report and continue	Report and continue
Unknown entity	Continue	Report and continue	Report and deny	Report and deny
Denied (anti-passback violation)	Continue	Report and deny	Report and deny	Report and deny

The Access Control object cooperates with the local Anti-Passback objects by keeping them informed about any transitions into and out of the anti-passback area. The *Anti-Passback Transition* attribute allows the user to specify what is considered a transition. The choices range from considering an identification, an access grant, or an access grant or a valid access, or not indicating a transition at all. The later option is useful when the transition is determined by a different Access Control object. The *Anti-Passback Transition Needs Open* attribute allows the user to specify considering the transition only when the portal was actually opened during the access time. This option is useful when an access profile is identified at the portal, but because the portal never opened, should not have its anti-passback status changed.

If the transitioning entity accompanies any assets or entities, or if it completes any entity groups or teams, all involved entities also transition at that time.

See “Anti-Passback Scenarios” on page 59 and the *Anti-Passback Object* document for details.

Occupancy

The occupancy feature is designed to restrict the access rights of entities to an occupancy space based on the number of entities that are currently in that space. If there is a discrepancy between the projected number of entities in the space, and the upper or lower limits of the number of entities that are supposed to be in the space, an occupancy violation occurs. Occupancy violations can be reported to the host, and may cause access to be denied for the entity who would be violating the

occupancy rule. An occupancy space may be accessible through several portals, that all must be handled by Access Control objects residing on the same controller. A single controller may hold several occupancy spaces.

The Access Control object can work with two Occupancy objects, one serving for entering an occupancy space, and one for exiting a different occupancy space. This capability covers both nested and adjacent occupancy spaces.

The occupancy rule is applied by setting the *Occupancy Check* attribute to either "Monitor Only," "Enforce, Only When Operational," or "Enforce, and Deny During Trouble" states. An occupancy decision other than "OK" is reported to the host by including the *Occupancy In Status* or *Occupancy Out Status* attributes in the notification.

Whenever the identified access profile's normal mask, or security mask in the emergency mode, has a set bit in common with the *Occupancy Exemption Mask* attribute, the entity is exempt from being denied access by any occupancy rule. In this case the *Occupancy In Status* and *Occupancy Out Status* attributes are set to "OK." These masks do not prevent an entity from being counted as an occupant.

The Access Control object requests occupancy status at on-box Occupancy objects, which immediately return a final decision.

The table below specifies how the *Occupancy Check* attribute affects the access decision based on the occupancy decision:

	Occupancy Check			
Occupancy Decision	**Disabled**	**Monitor Only**	**Enforce, Only When Operational**	**Enforce, and Deny During Trouble**
OK	Continue	Continue	Continue	Continue
Denied	Continue	Report and deny	Report and deny	Report and deny
Not operational	Continue	Report and continue	Report and continue	Report and deny
Already in	Continue	Report and continue	Report and continue	Report and continue
Never counted	Continue	Report and continue	Report and continue	Report and continue

The Access Control object co-operates with the local Occupancy objects by keeping them informed about any transitions into and out of the occupancy space. The *Occupancy Transition* attribute allows the user to specify what is considered a transition. The choices range from considering an identification, an access grant, or an access grant or a valid access, or not indicating a transition at all. The later option is useful when the transition is determined by a different Access Control object. The *Occupancy Transition Needs Open* attribute allows the user to specify considering the transition only when the portal was actually opened during the access time. This

option is useful when an access profile is identified at the portal, but because the portal never opened, the number of entities in the occupancy space did not change. For precision counting, the Occupancy object should be informed about transitions by external objects representing people counters or turnstile counters.

If the transitioning entity accompanies any assets or entities, or if it completes any entity groups or teams, all involved entities also transition at that time.

See the *Occupancy Object* document for details.

Anti-Loitering

The anti-loitering feature is designed to monitor entities in an anti-loitering area. An anti-loitering area may be accessible through several portals, and may span controllers. Also, a single controller may hold several anti-loitering areas. Per anti-loitering area there is exactly one Anti-Loitering object. This object is responsible for keeping a list of monitored and loitering entities inside the anti-loitering area.

In order to be exempt from the anti-loitering feature, the identified access profile's normal mask, or security mask in the emergency mode, must have a set bit in common with the *Anti-Loitering Exemption Mask* attribute.

The Access Control object may inform one Anti-Loitering object about access profiles transitioning into its anti-loitering area, and may inform another Anti-Loitering object about access profiles transitioning out of its anti-loitering area. This capability covers both nested and adjacent anti-loitering areas.

The *Anti-Loitering Transition* attribute allows the user to specify what is considered a transition. The choices range from considering an identification, an access grant, or an access grant or a valid access. The *Anti-Loitering Transition Needs Open* attribute allows the user to specify considering the transition only when the portal was actually opened during the access time. This option is useful when an access profile is identified at the portal, but because the portal never opened, the monitoring status never changed.See the *Anti-Loitering Object* document for details.

Intrusion Detection System

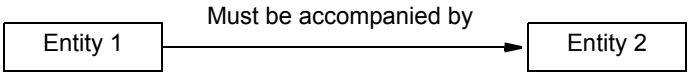
The Access Control object reads the *Arm Status* attribute of the object referenced in its *Intrusion Area Object* attribute when making an access decision. The object referenced in the *Intrusion Area Object* attribute must return a value of "Disarmed" to allow access for access profiles that do not have the right to disarm the intrusion area. See the *Intrusion Keypad/Display Object* document for details.

SCENARIOS

Asset Processing Scenarios

Laptop Detection

A laptop (Entity 1) has a long range asset tag, that is read by readers installed at all perimeter portals with a range of approximately 10 feet.



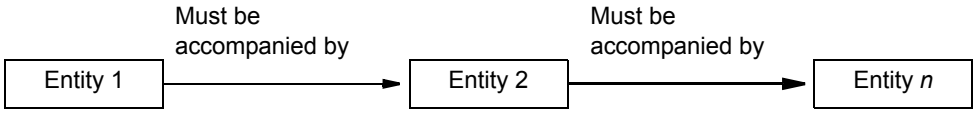
As soon as the laptop is detected, and its owner (Entity 2) was not identified in the recent past as defined by the *Asset Trailing Time* attribute, the *Unaccompanied Assets* attribute is increased, and an Asset Tracked notification is generated. The notification can be used to trigger events at the host, such as a CCTV action. The change in the *Unaccompanied Assets* attribute can trigger actions in the controller.

When the laptop's owner is detected, the *Unaccompanied Assets* attribute is decreased, and a notification for the owner requesting access is generated. The change in the *Unaccompanied Assets* attribute can be used to trigger actions in the controller.

When the laptop's owner is not detected and the laptop is not seen anymore, after the time specified in the *Asset Timeout* attribute has passed, the *Unaccompanied Assets* attribute is decreased and an Unaccompanied Asset notification is generated. The notification can be used to trigger events at the host, such as raising an alarm.

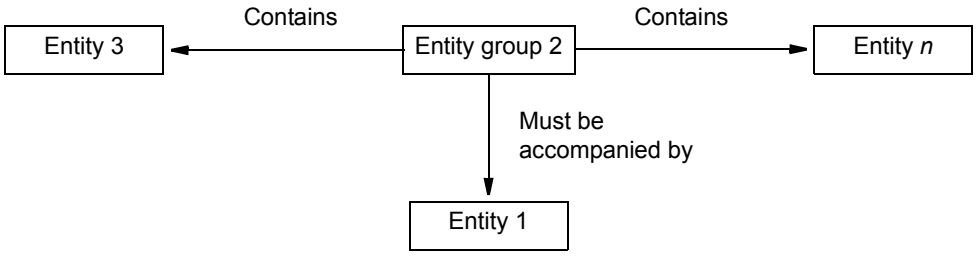
Multiple Owner Equipment

Lab equipment (Entity 1) may be taken from the lab by either one of two entities (Entity 3, Entity *n*).



Group Owned Lab Equipment

Lab equipment (Entity 1) may be taken from the lab by anyone belonging to the engineering group (Entity Group 2). This entity group lists all members of the engineering group (Entity 3, Entity n). This scenario is similar to the one above, but requires less maintenance in managing the ownership rights.

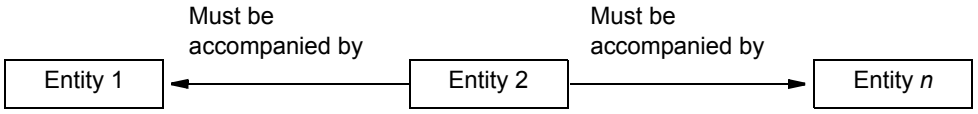


Note that neither Entity 3 nor Entity *n* have an association of type "Is a Member of" to Entity Group 2. This allows these entities to individually go through portals subject to group processing. Also, an association of type "Must Be Accompanied by" does not invoke any group processing actions.

Entity Processing Scenarios

Escorted Visitor

A visitor (Entity 1) may use certain portals without escort, while requiring an escort at other portals. In this example, any one of two individuals can be the visitor's escort (Entity 2 and Entity 3).



These two individuals are free to move around the facility based on their access rights, that is, they could leave the visitor in an area for a while until they escort him out of that area.

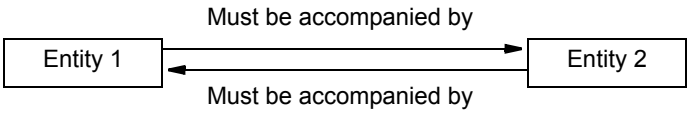
When a visitor is detected, and none of its escort was identified in the recent past as defined by the *Entity Trailing Time* attribute, the *Unaccompanied Entities* attribute is increased, and a notification is generated. The notification can be used to trigger events at the host, such as a CCTV action. The change in the *Unaccompanied Entities* attribute can trigger actions in the controller.

When any one of the visitor's escort is detected, the *Unaccompanied Entities* attribute is decreased, and a notification for the escort requesting access is generated. The change in the *Unaccompanied Entities* attribute can be used to trigger different actions in the controller, such as removing the portal lock-out.

When none of the visitor's escorts is detected, and the visitor is also not detected anymore after the time specified in the *Entity Timeout* attribute has passed, the *Unaccompanied Entities* attribute is decreased and an Unaccompanied Entity notification is generated. The notification can be used to trigger events at the host, such as raising an alarm.

Escorted Visitor with Bound Escort

Visitor (Entity 1) must be escorted, and the escort (Entity 2) cannot leave the visitor behind, as his access rights are restricted when the visitor is not detected within the specified time.



Group Processing Scenarios

Reduced Identification and Verification Requirements

In some instances it may be impractical to subject all members of a group to stringent identification and verification requirements, especially when they include PINs or biometrics. In this case the group leader may invoke a controller event, enabling an alternate set of identification requirements, which consists of simply reading the visitors' badges. This activation is done by setting the respective *Set...Enabled* flag to "True." The controller can be programmed to reset this flag as soon as the portal is opened or a certain time has expired, whatever comes first. This is important in not leaving it up to the group leader to do this, as this may pose a security risk.

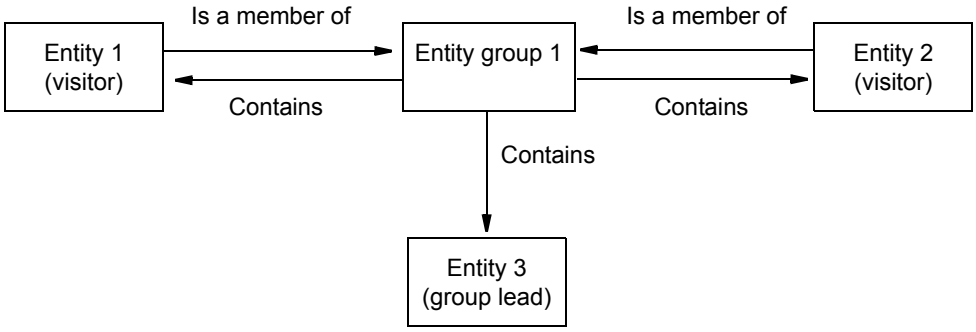
Reduced Access Rights

It is not desirable to give all visitors access rights to the portals they will pass through as a group. Therefore, the group processing feature uses a mere identification rather than a positive access decision to register the visitors at a portal. For a better feedback of correct registration to the visitor, its access profile's *Access Mask* attribute can contain a flag that matches the Access Control object's *Validation Only Mask* attribute. This way the visitor gets a green light whenever he was registered by the Access Control object, independent of his actual access rights.

This is also helpful to filter out any unwanted denied access notifications. In many cases, not only third party visitors with no access rights whatsoever are escorted through a facility, but also employees with access rights to areas other than those visited.

Visitor Group

When an entity is associated with an entity group through an association of type "Is a Member of," the references group is checked for completeness. This includes all entities the group references by a "contains" association, regardless of whether the referenced entities have an association back to the entity group.



In this scenario the group leader (Entity 3) is free to go through all portals by himself without starting a group processing action.

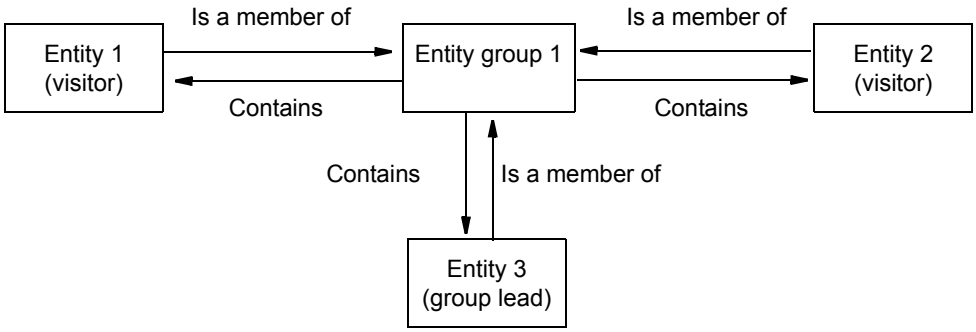
When either visitor (Entity 1 or Entity 2) is detected at any portal that requires complete groups within the time specified by the *Group Timeout* attribute, and not all group members have been recently identified, the *Incomplete Groups* attribute is increased, and a notification is generated. The notification can be used to trigger events at the host, such as a CCTV action. The change in the *Incomplete Groups* attribute can be used to trigger actions in the controller.

When all of the group's members (Entities 1, 2 and 3) have been identified, the *Incomplete Groups* attribute is decreased, and a notification for the last entity requesting access is generated. To open the portal, an entity with access rights needs to be identified to complete the group. Note that a group member being identified after the group is complete does not add again to the incomplete groups, as long as all members of that group have a recent enough identification. The change in the *Incomplete Groups* attribute can be used to trigger actions in the controller.

When all of the group's members (Entities 1, 2 and 3) were identified, the *Incomplete Groups* attribute is decreased and an Incomplete Group notification is generated. The notification can be used to trigger events at the host, such as raising an alarm.

Visitor Group with Bound Group Lead

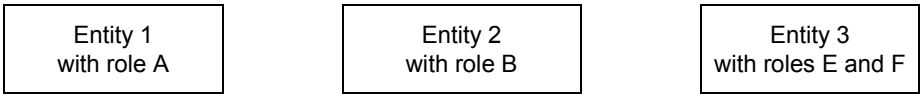
In this scenario the group lead (Entity 3) can only get access to the portals subject to group processing when the entire group (Entity Group 1) is present.



Team Processing Scenarios

Functional Teams

A portal can only be unlocked when members with roles A, B, E and F are present within the time specified by the *Team Timeout* attribute.



The required roles are specified in the *Team Mask* attribute, and have to match the collection of the identified roles based on the identified entity's access profile's normal mask, or security mask in the emergency mode. The *Team N-Man Rule* attribute can be set to 0 or 1, as theoretically a single entity can assume all required roles.

For this feature each entity must be qualified, that is, entities that do not have access rights to the portal cannot contribute to the team processing rule, even though they may have a required role.

N-Man Teams

A portal can only be unlocked when 3 different entities are present within the time specified by the *Team Timeout* attribute. In this case the *Team Mask* attribute must be set to 0, so that every qualified entity counts toward the N-Man minimum. The *Team N-Man Rule* attribute must be set to 3.

For this feature each entity must be qualified, that is, entities that do not have access rights to the portal cannot contribute towards meeting the N-Man requirement.

High Security Access Scenario

Each identification set can individually be enabled or disabled, by schedule, by event, or by custom logic. All different identification modes need to be programmed in the Access Control object. The *Set...Enabled Default* attributes should be set so that immediately after the controller restarts only the most secure identification set is enabled, if any. It is then up to the starting logic or schedule objects to get the identification requirements adjusted to the current needs.

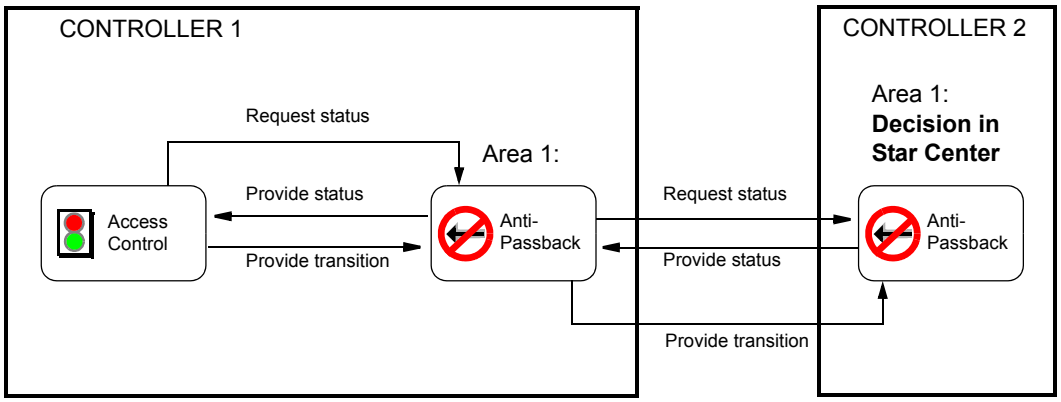
Anti-Passback Scenarios

Local Status, Local Decision

In this case an anti-passback area resides wholly within one controller. The anti-passback status for that area is managed by a single Anti-Passback object, which is local to all involved Access Control objects. This approach ensures a fast access decision which is always based on the most recent anti-passback status.

Star Center Status, Star Center Decision

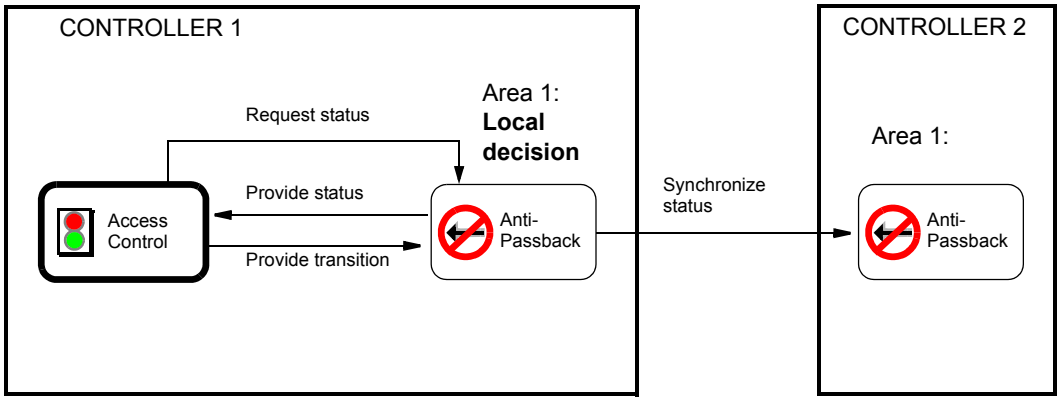
This scenario is applicable for large anti-passback areas with a low to moderate access request rate. All involved Anti-Passback objects are connected in a star configuration, with the decision maker in the center of the star. Each access decision invokes an anti-passback decision from a central Anti-Passback object. This method requires significantly more time than a local decision.



Shared Status, Local Decision

Anti-Passback objects have the ability to synchronize their anti-passback status with other Anti-Passback objects serving the same area as transitions occur. Since the effort to keep the data synchronized grows exponentially with the number of involved Anti-Passback objects, the number of such objects participating in a single area of this type is very limited. See the *Anti-Passback Object* document for details.

This method allows quick local decisions, but does not guarantee to always base the decision on the latest anti-passback status. Therefore, all portals that are in ultimate vicinity to each other, should be handled by the same controller.



A good example for this application is a company that has 3 different parking structures for their employees, but does not want a single employee to take more than one car into any of these structures during a specified time. Since it is not practical to always request the anti-passback status for the area centrally, each parking structure would keep the other two structures synchronized about its anti-passback transitions. This information typically travels faster from one Anti-Passback object to another than a person can travel from one parking structure to another.

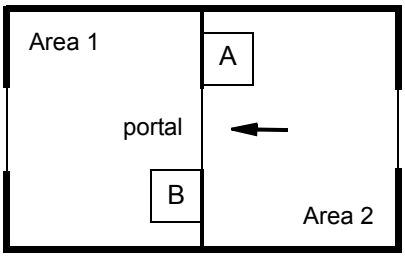
Area Scenarios

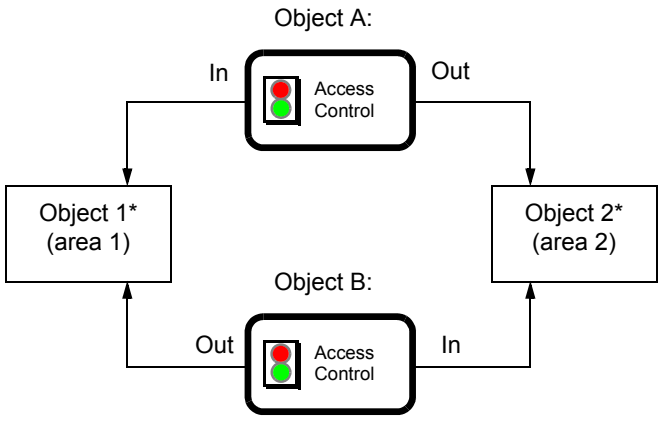
The anti-passback, occupancy, and anti-loitering applications allow transitions between adjacent areas and certain types of nested areas to be handled by a single Access Control object.

Adjacent Areas

An Access Control object can serve for transitions into one and out of another area at the same time.

In the scenario illustrated below, a person transitioning at Access Control Object A enters area 1 and exits area 2.



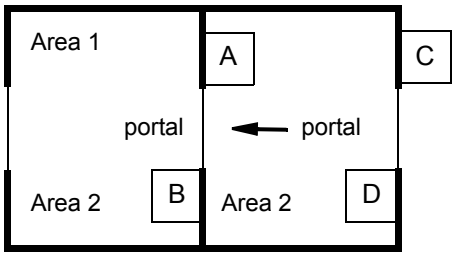


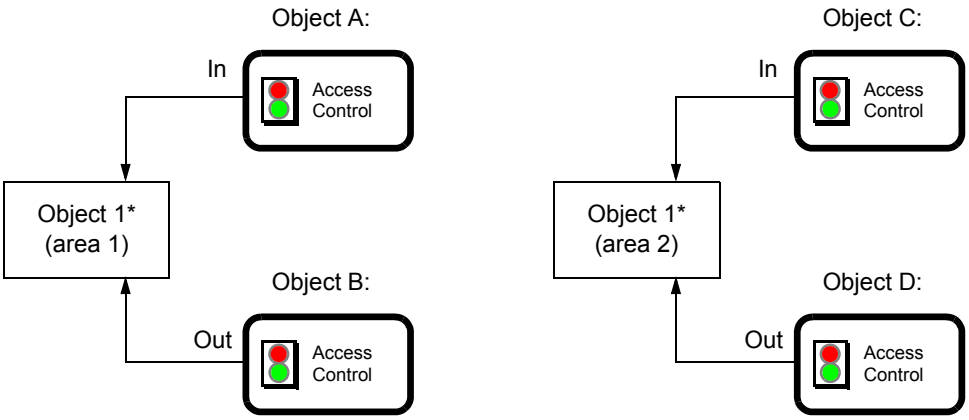
* Anti-Passback object, Occupancy object, or Anti-Loitering object

Supported Nested Areas

The Access Control object supports areas that are fully nested within other areas, as long as the building requires access to an inner area through an outer area.

In the scenario illustrated below, a transition at Access Control Object A enters area 1 while remaining in area 2. Note that for this feature to work correctly, accessing area 1 must not be possible without also accessing area 2.





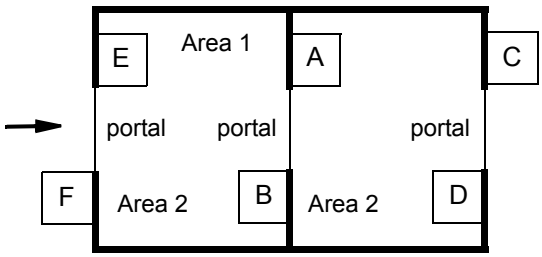
* Anti-Passback object, Occupancy object, or Anti-Loitering object

Unsupported Nested Areas

Some nested area configurations are not supported by the Access Control object.

In the scenario illustrated below, a transition at Access Control Object F would enter areas 1 and 2, which is not supported. Making the two rooms adjacent areas as shown in “Adjacent Areas” on page 60 resolves this problem.

Example of an area configuration that is **NOT supported:**



Using Masks

In Johnson Controls’ previous access control systems, certain privileges were assigned to entities on a global level, such as the executive privilege or the override privilege.

This meant, that a person with override privilege was able to invoke timed overrides at any portal that allowed such overrides.

The introduction of masks makes assigning these privileges more granular, as now each portal determines which access profile’s will be granted certain privileges. It is now possible to give certain people override privileges at certain portals, while giving other people override privileges at other portals.

Furthermore, the Access Control object automatically switches from an access profile’s normal mask to its security mask in the emergency mode. This allows rapidly giving all entities alternative privileges without having to change the entities’ database, which may be too time consuming in an emergency.

The following table shows a summary of all masks that are used by the Access Control object. For more details refer to the respective sections within this document.

Mask	Specifies whether the access profile is...
Executive Privilege Mask	...exempt from certain parts of the access decision algorithm
Override Mask	...allowed to invoke timed overrides
Access Mask	...passing the access mask check of the access decision algorithm
Security Mask	...passing the security mask check of the access decision algorithm
Team Mask	...contributing to the team requirement
Alternate Access Mask	...invoking an alternate access time when access is granted
Validation Only Mask	...exempt from access decision algorithm, but also from access
Anti-Loitering Exemption Mask	...exempt from monitoring by the anti-loitering feature
Team Exemption Mask	...exempt from the team requirement
Occupancy Exemption Mask	...exempt from the occupancy check
Anti-Passback Exemption Mask	...exempt from the anti-passback check
Central Status Exemption Mask	...exempt from the central status check
Trigger Mask	...automatically triggering a controller event

Even though all features share the total of 100 possible bits in a mask, the host may decide to segregate these 100 bits into different sections on the user interface. The host may for example set aside 9 dedicated bits to individually control executive privilege, override, central status exemption, team exemption, anti-passback exemption, occupancy exemption, anti-loitering exemption, and alternate access, while leaving the remaining 91 bits for features like team, access, and controller event trigger. Even within this group the host may decide to segregate the bits into further independent sections.