Johnson Controls

Object Library

Interlock Object

Version 1.0, December 2008
24-10239-189 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.

INTERLOCK OBJECT

INTRODUCTION

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

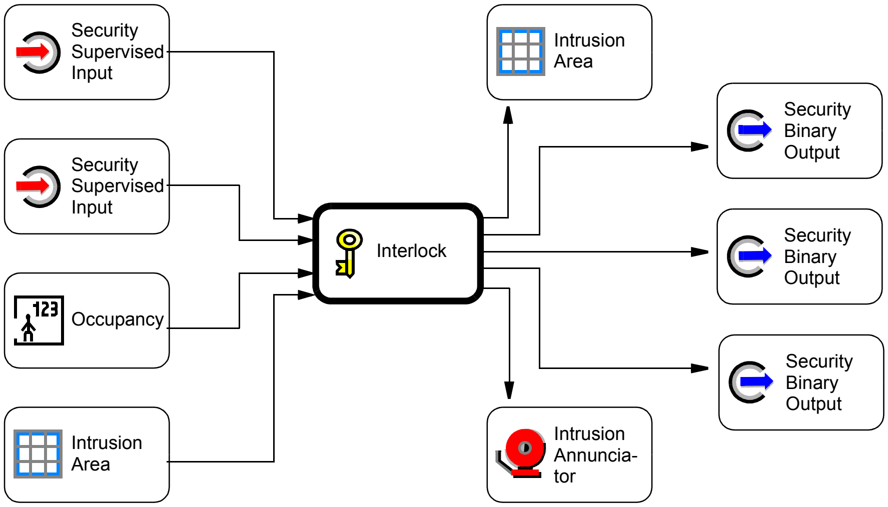


Figure 1: Interlock Object

Three essential parameters of the IF conditional statement, shown in the following list, determine the Interlock object's mode of operation:

- List of attribute references
- Constants
- Operators

ATTRIBUTES

This object contains attributes common to all objects in the P2000 Security Management System. For details, see the *General Object Information* document. The table below lists only the attributes specific to the Interlock object.

Table 1: Interlock Object Attributes

Attribute Name	Attribute Number	Data Type	Notes	Initial Value	Values/Options/ Range
Action Tbl 1 (True)	793	List of Command References	CW	-	Object Name Delay (0-65,535 seconds) Command ID Parameters
Action Tbl 2 (False)	794	List of Command References	CW	-	Object Name Delay (0-65,535 seconds) Command ID Parameters
Commands Priority	1020	One state from a set	CW	16	See Table 2 on page 6
Executing	790	True or False	-	False	-
Last Executed Date	1039	Date	N	-	-
Last Executed Time	1044	Time	N	-	-
Logic	1070	One state from a set	CW	0	Match All = 0 Match Any = 1 Complex = 2
Logic Equation	550	Text	CW	-	Maximum 56 characters
Present Value	85	One state from a set	DRW	-	-
Reliability	103	One state from a set	-	-	Not used
Source	751	List of value sets	CW	-	Object Name Attribute Relation (Equal, Not Equal, Greater Than, Less Than, Greater Or Equal, Less Or Equal) Value Differential (required and only used when the value is analog)
States Text	931	One state from a set	CW	-	Defaults to False/True, but you can use any of the Two State Values

C - Configurable, D - Default attribute for display, N - Value not required, R - Affected by object reliability, W - Writable

Action Tbl 1 (True) – Lists commands issued by this object when the condition is true:

- Item - The name of the object to be commanded
- Delay - The amount of time, in seconds, each command waits to be sent, after a change in condition

- Command ID - The type of command to send to the object
- Parameters - These vary according to command.

Action Tbl 2 (False) – Lists commands issued by this object when the condition is false:

- Item - The name of the object to be commanded
- Delay - The amount of time, in seconds, each command waits to be sent, after a change in condition
- Command ID - The type of command to send to the object
- Parameters - These vary according to command.

Any object capable of accepting commands can be defined in either action table. The available commands depend on the type of object and the user's access capability. An object may be sent more than one command by identifying the object more than once and specifying a different command. The user does not have to specify objects for both the “True” and “False” Action Tables. If no command is specified for an object, no command is sent.

The total number of commands specified in the Action Table command set should not exceed 40.

Every object accepts the “Change Attribute” command. This command can be used to change the object’s writeable attributes of the following types: Boolean, Number (Signed8, Unsigned8, Signed16, Unsigned16, Signed32, Unsigned32), Ennumeration, and Float.

Prioritized attributes are written to the specified value at the priority described in the *Commands Priority* attribute.

To release a prioritized attribute at the specified priority, leave the value of the “Change Attribute” command blank.

Commands Priority – Determines the order by importance for all commands sent. If the “All Commands Priority” option is not selected, the commands are sent at each individual priority specified in the *Action Tbl* attribute. If the “All Commands Priority” option is selected, then every item in the Action Table is sent a command at the priority specified. For example, if the command chosen is Operator Override, the command priority automatically sets to Operator Override (8) for every item listed in the Action Table. When the All Commands Priority option is selected, you cannot change the command priority within the Action Table.

The *Commands Priority* attribute uses the priorities in the Write Priority Set (see Table 2 on page 6).

Executing – Specifies the current execution status of the object. It is “True” when the object receives a state command and starts sending commands. When all actions are completed for a given state, this flag is set to “False.”

Last Executed Date – Indicates the date stamp for the beginning of executing the current state.

Last Executed Time – Indicates the time stamp for the beginning of executing the current state.

Logic – Specifies the Boolean logic (And/Or) to be employed in combination with source inputs in deriving a “True” or “False” outcome. The options are:

- Match All - Every one of the source conditions must be “True.”
- Match Any - One or more of the source conditions must be “True.”
- Complex - Use the *Logic Equation* attribute for complex logic.

Logic Equation – Contains a logic equation when the *Logic* attribute is set to “Complex.”

Use the logic equation to define the relationship between the master conditions using logic operators and parentheses for determining precedence. Valid characters in the *Logic Equation* string are:

*	AND operator
+	OR operator
(	Open parenthesis (for grouping)
)	Closed parenthesis (for grouping)
Numbers	Master condition numbers
Spaces	Spaces are allowed for readability

Equation Syntax:

- Open and close parenthesis must match.
- Empty parenthesis are not allowed ().
- The equation must be at least 3 characters in length, not including spaces.
- The numbers must not exceed the master condition numbers.
- There is no implied precedence for the operators (*,+).
- Any combination of different operators must be grouped with parenthesis to determine precedence.
- Operators of the same type do not require the use of parenthesis.
- A grouping may contain only one operator type.
- Numbers or groupings must be separated by an operator (*,+).
- Groups can be nested.

Examples:

Incorrect	Correct
1+2+3*4	1+2+(3*4) or (1+2+3)*4
1*2+3*4	(1*2)+(3*4) or 1*(2+3)*4
1*2*3+4	1*((2*3)+4) or (1*2)*(3+4) or (1*2*3)+4
(1+2)*(3+4)+(6*7)	((1+2)*(3+4))+(6*7) or (1+2)*((3+4)+(6*7))
(3*4)(1*2)	(3*4)+(1*2)
(3*+1)	(3*1)
(3*	(3*1)

Sample logic equation:

((1+2)*(3+4))+(6*7)

Where the equation says that master condition ((1 OR 2) AND (3 OR 4)) OR (6 AND 7) results in a TRUE (Action Tbl 1) execution.

Objects are given numbers in the left column that correspond to the order you add them to the Interlock Definition list (the first added is 1, the second added is 2, and so on). Use these numbers to represent the corresponding objects in the *Logic Equation*.

Present Value – Represents the current value of the object. Present Value is set based on the relationships set up in the Interlock definition.

Reliability – Indicates if the Present Value is questionable.

Source – Lists objects whose attributes are used for input into the logic expression to decide if the condition is “True” or “False.” Information needed:

- Item - Name of the source object
- Attribute - Varies according to object, *Present Value* is the default value
- Relation - Equal, Not Equal, Greater Than, Less Than, Greater Than or Equal, Less Than or Equal
- Value - The value used in the Relation. Varies according to the object.
- Differential - Accounts for variations in the value. Required (and only used) if the value is analog.

States Text – Indicates the displayed strings for the *Present Value*.

Write Priorities

Table 2: Write Priority Set

Numeric Value	Text
0	No Priority - MC
1	Manual Life Safety - High Priority
2	Auto Life Safety - AS High Priority
3	Priority 3 - 03 High Priority
4	Priority 4 - 04 High Priority
5	Critical Equipment - CE High Priority
6	Minimum On Off - MO
7	Heavy Equip Delay - HE
8	Operator Override - OV
9	Priority 9 - 09, System Override – SO
10	Priority 10 - 10, Security Mode - SM
11	Demand Limiting - DL, Lock Down - LD
12	Priority 12 - 12
13	Load Rolling - LR, Timed Override - TO
14	Priority 14 - 14, Elevator - EL, Door Sequence - DS
15	Scheduling - WS
16	Default - MC

COMMANDS

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

Table 3: Interlock Object Commands

Command Name	Description
`Recommand`	Send all slave commands from one of the Action Tables as determined by the current *Present Value* attribute, unless the object is disabled. For more details see "Description of Operation" on page 9.
`Override`	Allow the user to override the object's *Present Value* attribute. See the description below for details. This is a default command.
`Override Release`	Release an `Override` command which clears the priority 8 (Manual Override) entry in the priority array property of the Interlock's object's *Present Value* attribute.

Table 3: Interlock Object Commands

Command Name	Description
`Release`	Release a specific priority of the *Present Value* attribute. This command is handled by the standard object class.
`Release All`	Copy the highest priority write to the *Present Value* attribute and place it in priority 16. In addition, priorities 3 to 15 will be released. This command is handled by the standard object class.
`Enable`	Writes *Enable* attribute to “True.” This command is handled by the standard object class.
`Disable`	Writes *Enable* attribute to “False.” A disabled Interlock never ever sends any commands to its slaves. When disabled, it also stops updating its *Present Value* and *Reliability* attributes, unless *Present Value* is being overridden, upon which *Reliability* is set to “Reliable.” This command is handled by the standard object class.
`Change Attribute`	See the description below.

The `Override` command will place the state defined by the Value parameter of the command into the priority array property of the Interlock object's *Present Value* attribute at priority 8 (Manual Override).

Table 4: Override Command Parameters

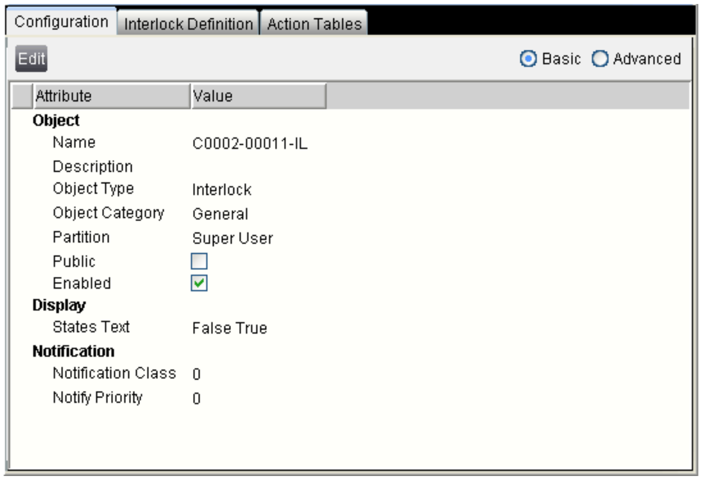
Parameter	Data Type
Value	Enumeration

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 Interlock object. These screens also allow you to set the values of configurable attributes. For more information refer to the *System Configuration Tool (SCT)* manual.

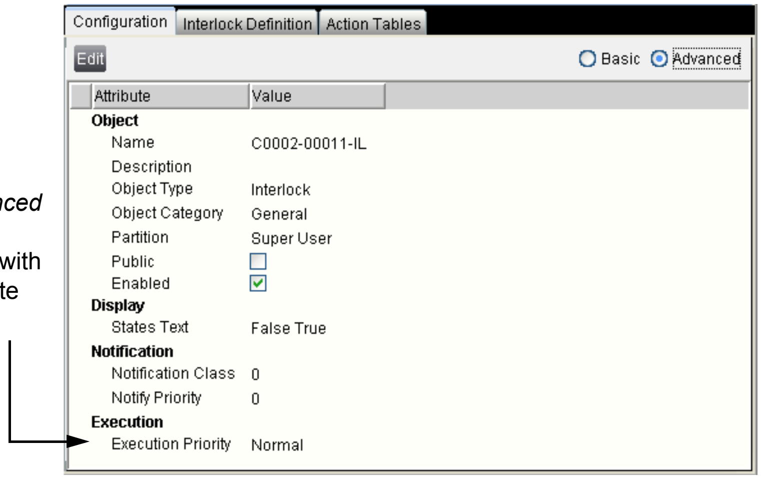


Configuration | Interlock Definition | Action Tables

Edit — ● Basic ○ Advanced

Attribute	Value
Object	
Name	C0002-00011-IL
Description	
Object Type	Interlock
Object Category	General
Partition	Super User
Public	□
Enabled	☑
Display	
States Text	False True
Notification	
Notification Class	0
Notify Priority	0

Figure 2: Configuration View (Basic)



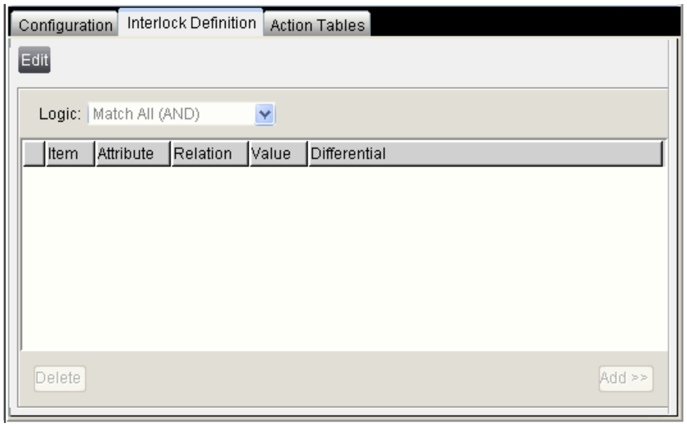
Configuration | Interlock Definition | Action Tables

Edit — ○ Basic ● Advanced

Attribute	Value
Object	
Name	C0002-00011-IL
Description	
Object Type	Interlock
Object Category	General
Partition	Super User
Public	□
Enabled	☑
Display	
States Text	False True
Notification	
Notification Class	0
Notify Priority	0
Execution	
Execution Priority	Normal

The *Advanced* view is expanded with this attribute

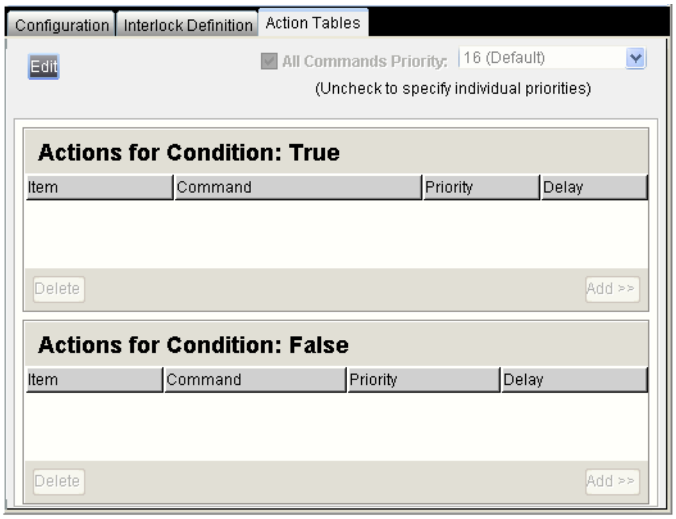
Figure 3: Configuration View (Advanced)



Logic: Match All (AND)

	Item	Attribute	Relation	Value	Differential

Figure 4: Interlock Definitions View



☑ All Commands Priority: 16 (Default)

(Uncheck to specify individual priorities)

Actions for Condition: True

Item	Command	Priority	Delay

Actions for Condition: False

Item	Command	Priority	Delay

Figure 5: Action Tables View

Description of Operation

Evaluation of Logic Expression

The evaluation of the logic expression affects or is affected by the following attributes: *Present Value*, *Enabled*, *Reliability*, and *Status*. The logic evaluation, as described further below, is performed on the following occasions:

- The object goes through its feature start routine
- The object is enabled
- A change-of-state or change-of-reliability is received from a source object

- The Logic or the *Present Value* attribute is written
- The `Recommand`, `Override`, or the `Override Release` command is invoked
- The Execute method is invoked (this method is not available as a user command)

Enabled Attribute

A disabled Interlock object does not send any commands to its slaves. When disabled, it also stops updating its *Present Value* and *Reliability* attributes, unless *Present Value* attribute is being overridden, upon which Reliability is set to RELIABLE.

Status and Reliability Attributes

The *Status* attribute indicates the highest prioritized overall condition of the Interlock object. The following values are supported, ranked from highest to lowest priority:

- **Disabled** - The Enabled flag is reset. In this case the *Reliability* attribute is left in its current state, unless *Present Value* attribute is overridden, upon which *Reliability* is set to “Reliable.” Regardless of override, no slave commands are sent as long as the object is disabled.
- **Unreliable** - The Enabled flag is set, and *Present Value* attribute is not overridden. At least one of the source objects is either offline or unreliable, and the logic expression cannot be evaluated, because among the offline or unreliable objects is at least one required source object. Reliability is set to “Unreliable.” No slave commands will be sent.
- **Trouble** - The Enabled flag is set, and *Present Value* attribute is not overridden. At least one of the source objects is either offline or unreliable, but the logic expression still can be evaluated, because there are still sufficient online and reliable source objects. Reliability is set to “Reliable.” Slave commands can be sent.
- **Normal** - The Interlock object is Enabled, *Present Value* attribute is not overridden and all of its source objects are online and reliable. Slave commands can be sent.

Present Value Attribute

The *Present Value* attribute supports a prioritized override. When overridden, *Reliability* attribute is always “Reliable.” When the override is released, and the Interlock object is enabled, the logic expression is evaluated. In case this evaluation cannot be completely performed, which is indicated by the *Reliability* attribute being set to “Unreliable,” *Present Value* attribute does not change. Otherwise, it holds the result of the logic expression.

Sending Slave Commands

Slave commands are never sent, and already queued commands are cancelled, when the Interlock object is disabled or unreliable. Otherwise, the slave commands are always sent as determined by the *Present Value* attribute in the following cases:

- The object goes through its feature start routine.
- The object transitions into a reliable and enabled state.
- The `Recommand`, `Override`, or the `Override Release` command is invoked.
- The *Present Value* attribute is written.
- The *Present Value* is changed as a result of the logic evaluation.