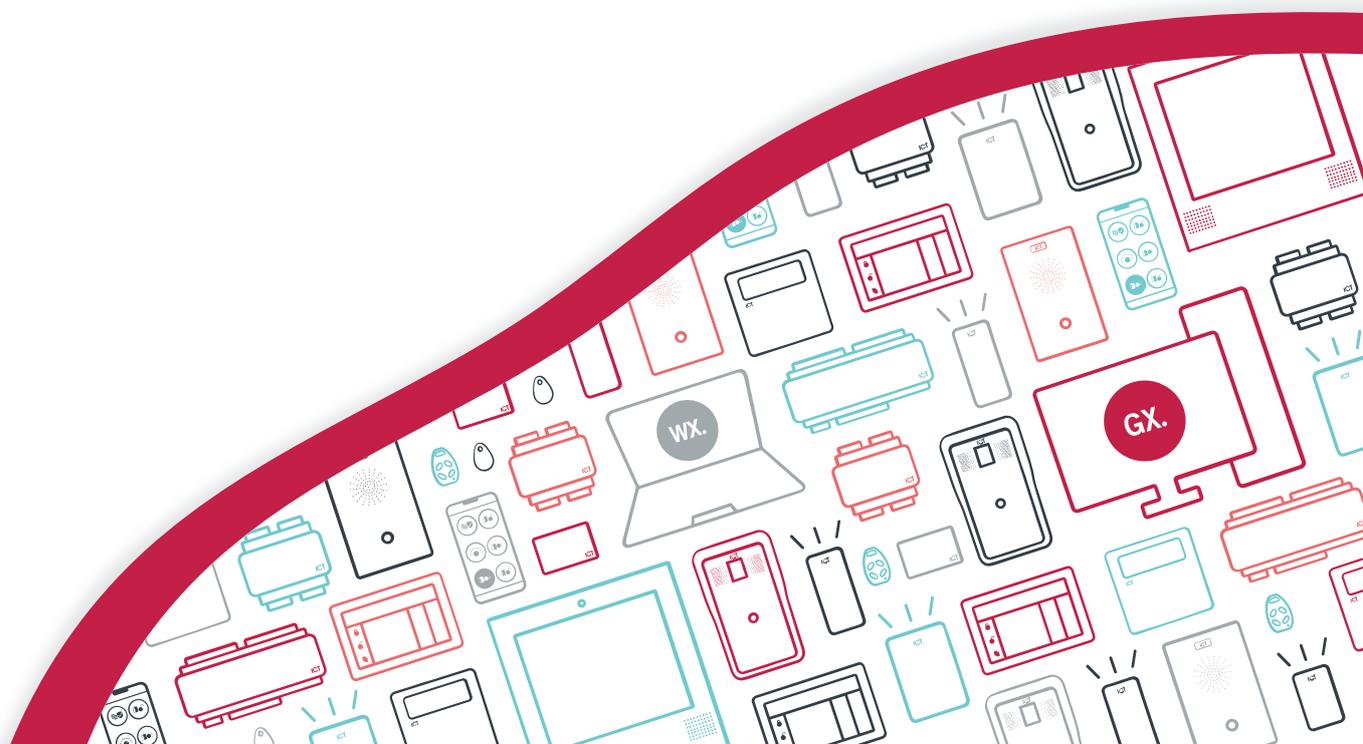




AN-296

Protege GX Otis Medium Level Interface Elevator Integration

Application Note



The specifications and descriptions of products and services contained in this document were correct at the time of printing. Integrated Control Technology Limited reserves the right to change specifications or withdraw products without notice. No part of this document may be reproduced, photocopied, or transmitted in any form or by any means (electronic or mechanical), for any purpose, without the express written permission of Integrated Control Technology Limited. Designed and manufactured by Integrated Control Technology Limited, Protege® and the Protege® Logo are registered trademarks of Integrated Control Technology Limited. All other brand or product names are trademarks or registered trademarks of their respective holders.

Copyright © Integrated Control Technology Limited 2003-2023. All rights reserved.

Last Published: 15-Sep-23 3:15 PM

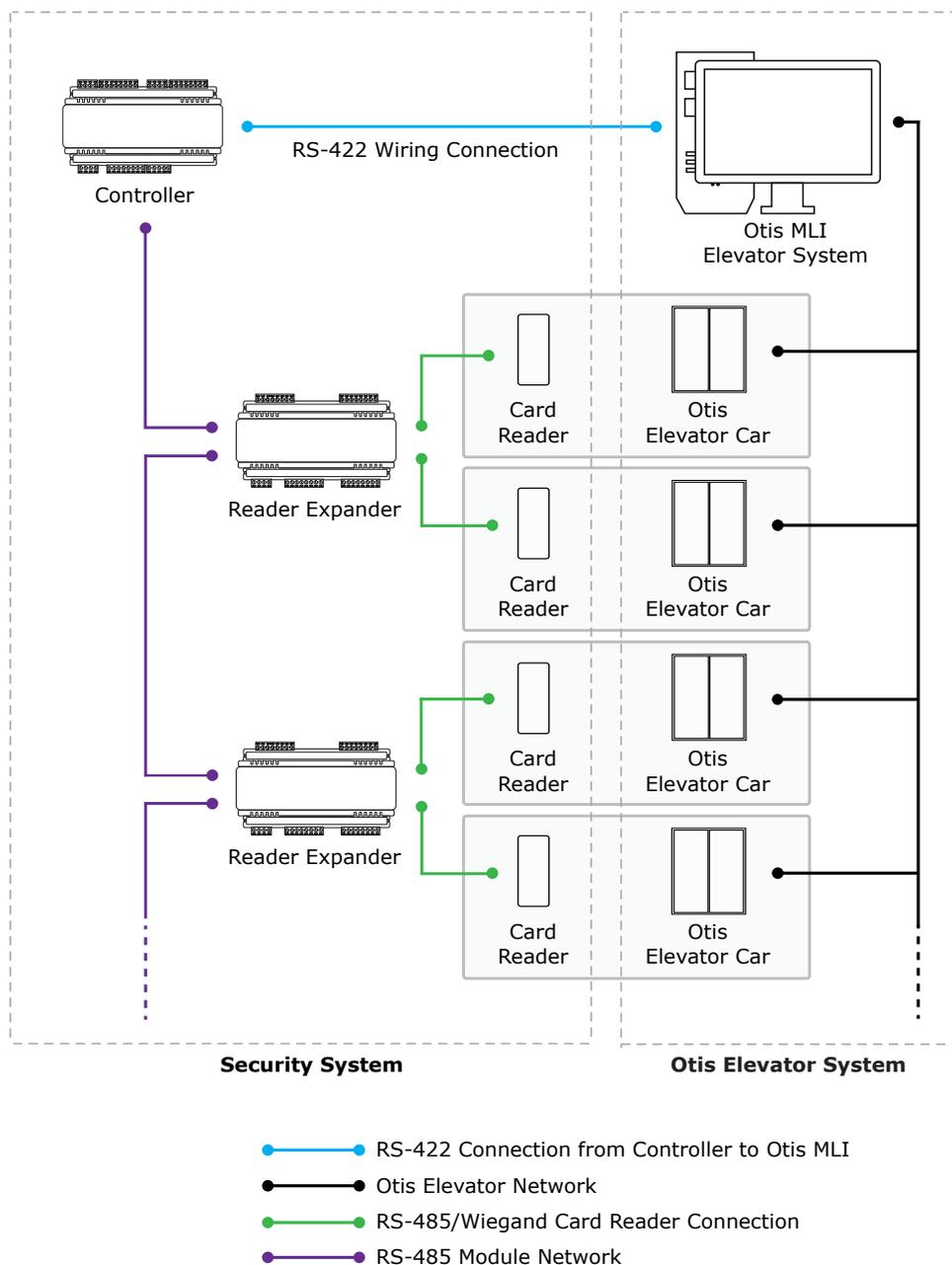
Contents

Introduction	4
Prerequisites	5
Floor Mapping	6
Floor Relay Numbering	6
Skipped Floor Numbers	6
Floors with Rear Doors	7
Connecting the Controller	9
Programming the Otis MLI Integration	10
Programming the Controller's Onboard Reader	10
Programming the Service	10
Configuring up to 128 Floors	11
Adding Floors	11
Programming Floor Groups	12
Programming the Elevator Cars as Doors	12
Programming Antipassback	12
Programming a Placeholder Elevator Car	13
Programming Reader Expanders	14
Configuring Access Levels	14
Troubleshooting	15

Introduction

Protege GX integration with the Otis Medium Level Interface (MLI) elevator system provides a destination dispatch solution integrated with access control and intruder detection.

Integration with the Otis MLI allows access to floors to be controlled from Protege GX. Users and associated access levels are configured within Protege GX and used to validate which floors people can travel to and when. ICT readers provide user identification and validation for floor access. When a user presents their card and selects a floor in an elevator, their credentials are verified in Protege GX. Once access has been verified, a message is sent to the Otis system which calls an elevator to transport the user to the selected floor. To the end user, this integration is as simple as presenting a card at a card reader and selecting the floor they wish to travel to.



The Protege GX Otis MLI elevator integration supports up to 128 floors.

This integration is a licensed feature.

Prerequisites

The Protege GX Otis Medium Level Interface integration requires the following software and hardware. All items must be installed and operational.

Required Item	Software/Firmware Version	Notes
Protege GX	4.3.285 or higher	
Protege GX Controller	2.08.886 or higher	<p>Only two-door DIN rail controllers with RS-485 reader ports support this integration.</p> <p>Each Protege GX controller supports integration to one Otis MLI system elevator group. The controller's onboard reader ports must be available for connection to the Otis system (see page 9).</p> <p>If the integration includes more than one Otis elevator group, the full integration programming instructions must be completed, including creating a placeholder car (see page 13), for each connected Protege GX controller.</p>
Otis Medium Level Interface elevator system	-	At the time of publication no version information is available for the Otis MLI elevator system. It is the responsibility of the installation professional to confirm with the elevator system provider that the system complies with the specifications required for this integration, as per Otis Elevator Company documentation.

Licensing

The following licenses must be applied to the Protege GX server.

License	Order Code	Notes
Protege GX Otis Elevator Medium Level Interface License	PRT-GX-ELV-MLI-OT	1 license per controller used for this integration
Protege GX Door License	PRT-GX-DOR-1	1 license per Otis elevator car programmed
	PRT-GX-DOR-10	
	PRT-GX-DOR-50	

It is the responsibility of the installation professional to verify the version of the proposed third-party system and supported components with the version listed in this document. ICT will not accept responsibility for the failure to verify integrated system versions and requirements.

This document describes how to program the Protege GX components to interface with the Otis MLI elevator system. It does not cover the programming or hardware setup required within the Otis system.

Floor Mapping

Before beginning to program the elevator integration, it is important to correctly map the layout of the elevator system. All elevator-accessible floors need to be identified and mapped in sequential order from bottom to top. You need to identify which (if any) floors have rear doors, and which (if any) floors are considered 'below ground' in the elevator system programming. This is essential for Protege GX to correctly map floors for event reporting.

Floor Relay Numbering

Each floor accessed by the elevator system needs to be programmed in Protege GX, with a unique **Floor relay** number assigned. The floor relay number tells the controller where the physical floor is located, creating a map in Protege GX of accessible floors.

Floor relay numbers must be **unique**, programmed in **numerical order** (starting at 1), and beginning at the **lowest accessible floor**, including any basement floors.

The lowest accessible floor must be assigned a floor relay of **1**, as shown in the example below.

Physical Floor	Floor Relay Number
Basement 2	1
Basement 1	2
Ground Floor	3
Level 1	4

Skipped Floor Numbers

Numbers are commonly skipped in floor labels due to superstitions surrounding specific numbers, and technicians may also skip these in the elevator controller programming. Where this occurs a placeholder floor record must be created in Protege GX to maintain the required sequential floor relay numbering, as in the example below.

Physical Floor	Elevator System Floor Relay	Floor Relay Number
Ground	0	1
Level 1	1	2
...
Level 12	12	13
N/A	Not programmed	14 (Placeholder)
Level 14	14	15

In this example a placeholder floor record is added with relay number 14 to maintain the sequential numbering and represent the skipped programming in the elevator system. Had the sequence been maintained in the elevator system (i.e. Level 14 programmed with elevator system floor relay 13) no placeholder floor would be required.

Floors with Rear Doors

For some floors it may be possible to exit the elevator at the rear as well as the front. Each 'rear floor' must also be programmed in Protege GX.

The rules for programming rear floors are as follows:

- The **Floor relay** number of the rear floor must be equal to that of the corresponding front floor.
- Each rear floor must be programmed with the command **Rear**.
- The controller must be programmed with the command **HLI_128_FLOORS = true**.

The **Rear** and **HLI_128_FLOORS = true** commands are supported in controller firmware version 2.08.1158 or higher. For earlier firmware versions, rear doors must be programmed with relay numbers from 65-128.

The table below demonstrates front and rear floor relays in a building with 50 aboveground floors and two basement floors.

Physical Floor	Front Floor Relay	Rear Floor Relay
Basement 2	1	1
Basement 1	2	2
Ground Floor (no rear access)	3	-
Level 1 (no rear access)	4	-
Level 2	5	5
...
Level 48	51	51
Level 49	52	52

It is not necessary to program rear floor records for any floors that do not have rear access, but you may want to do this to keep the programming tidy.

Rear Floor Relays 65-128

For sites with up to 64 floors the rear floor relays can be programmed from 65-128 without using the **Rear** command. To calculate the rear floor relay, add 64 to the front floor relay.

This is a legacy programming option which provides backward compatibility for existing installations and controller firmware versions prior to 2.08.1158, but is not recommended for new installations. It supports a maximum of 64 floors and is not available if the controller has been configured to support 128 floors.

The table below demonstrates front and rear floor relays in a building with 50 aboveground floors and two basement floors where the **Rear** command is not used and rear floor relays start at 65.

Physical Floor	Front Floor Relay	Rear Floor Relay
Basement 2	1	65
Basement 1	2	66
Ground Floor (no rear access)	3	-
Level 1 (no rear access)	4	-
Level 2	5	69
...
Level 48	51	115
Level 49	52	116

Connecting the Controller

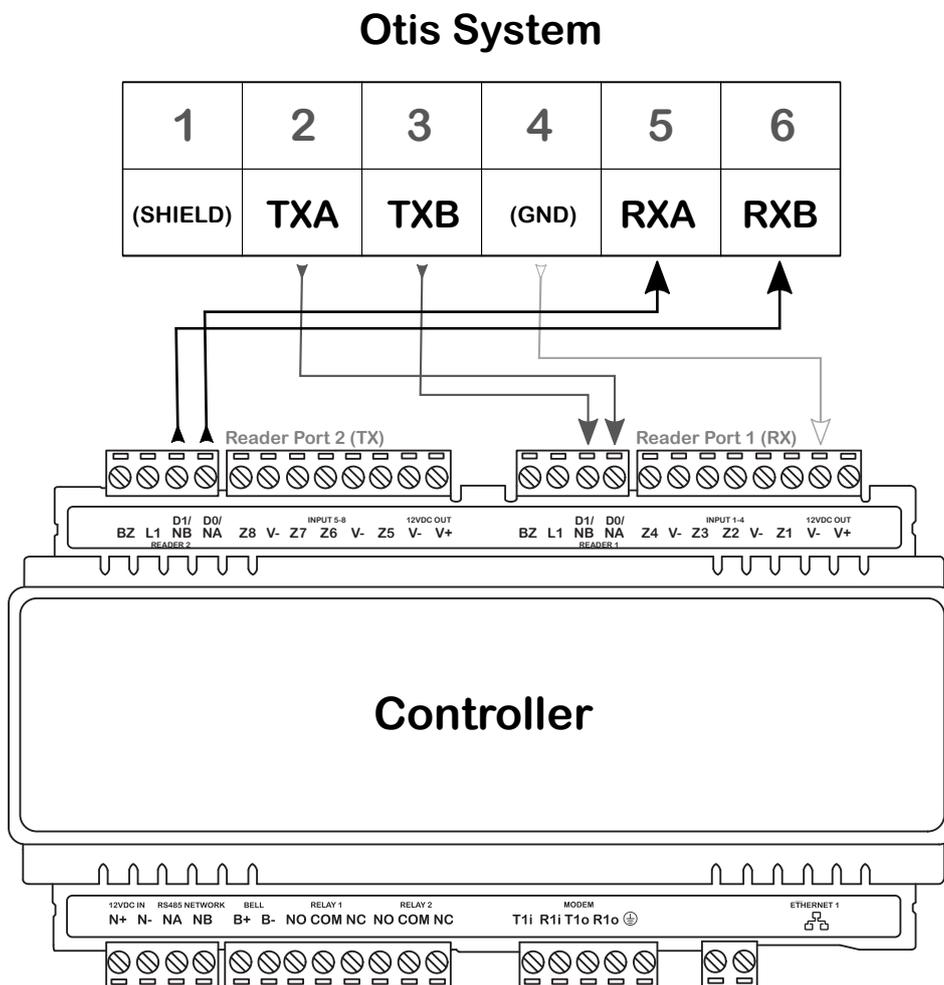
The Protege GX controller used in the integration is physically connected to the Otis system via the reader ports on the controller's onboard reader expander.

- Reader port 1 **receives** via the NA and NB terminals from the Otis system TXA and TXB respectively.
- Reader port 2 **transmits** via the NA and NB terminals to the Otis system RXA and RXB respectively.

Wiring is **RS-422** - 4 wire (2 pairs) plus reference ground and shielding.

* The ground wire may be connected to any ground (**V-**) terminal on the controller.

Typical RS-422 Wiring for Otis MLI Integration



Programming the Otis MLI Integration

Programming the Controller's Onboard Reader

The Otis Medium Level Interface is based on **AEA Type-B** (commonly referred to as the KONE protocol). The MLI differs from AEA Type-B in that baud rate has been altered from 1200 to 9600, and parity has been changed from none to odd.

The Protege GX controller used in the integration is physically connected to the Otis MLI system via the onboard reader expander, where reader port 1 handles RX (receive) and reader port 2 handles TX (transmit).

In order to facilitate this connection the controller's onboard reader expander must be configured via commands to be enabled for the Otis MLI integration.

1. Navigate to **Expanders | Reader expanders** and select the controller's onboard reader expander.
2. In the **Commands** field enter the following commands:

```
PortOne = 6  
PortTwo = 6
```
3. Click **Save**.
4. Wait for the changes to be downloaded to the controller, then right click on the reader expander record and click **Update module**.

Command	Description
PortOne = <value>	Enables the controller's reader port 1 for the Otis Integration. The valid value to enable the Otis Medium Level Interface integration is 6 .
PortTwo = <value>	Enables the controller's reader port 2 for the Otis Integration. The valid value to enable the Otis Medium Level Interface integration is 6 .

Note: The commands must be added for both reader ports in order to enable the integration, and both reader ports must contain the same value.

Programming the Service

The Protege GX Otis MLI elevator integration operates as a specially configured service. The service is configured, started and stopped by adding specific commands to the controller programming.

1. Navigate to **Sites | Controllers** and select the relevant controller.
2. In the **Commands** field enter the following commands:

```
AEA = true  
AEAPort = 1  
AEAGroupNumber = <value>  
AEADebug = false
```
3. Click **Save**.

The parameters and related options to be programmed in the controller commands are as follows:

Command	Description
AEA = <value>	Enables or disables the Otis MLI integration service. Valid values are true (starts the service) or false (stops the service). This must be set to true to enable the integration.
AEAPort = 1	Defines the reader port of the onboard reader expander handling the transmission of data for the service. This must always be set to 1 to represent reader port 2.
AEAGroupNumber = <value>	Defines the Otis system group number of the elevators that this controller communicates with. One elevator group, numbered from 1 to 8 inclusive, is supported per controller.
AEADebug = <value>	Enables or disables the display of debug information for the service. Valid values are true (enabled) or false (disabled). When enabled, debug information appears as system debug message events in the event log.

Configuring up to 128 Floors

By default the controller supports up to 64 floors for this integration. However, with some additional configuration the integration can support up to a maximum of 128 floors.

This feature requires controller firmware version 2.08.1158 or higher.

1. Navigate to **Sites | Controllers** and select the controller to be used for the integration.
2. Expand the **Commands** field and enter the following commands:
 - **HLI_MAX_FLOORS = X**
Where **X** is the total number of floors in the building. This can be a value from 1-128.
 - **HLI_128_FLOORS = true**
This command is required to enable the controller to support using the **Rear** command for rear floors.
3. Click **Save**.

Adding Floors

Each floor accessed by the elevator system needs to be configured in Protege GX with a floor relay number assigned. This tells the controller where the physical floor is located, creating a map of accessible floors.

1. Navigate to **Programming | Floors**.
2. Add a floor record for each elevator-accessible physical floor, assigning the **Floor relay** for each floor as explained in the Floor Mapping section (see page 6).
3. If the floor is a 'rear floor' add the **Rear** command in the **Commands** field.

The **Rear** command is supported in controller firmware version 2.08.1158 and above. The controller must also be programmed with the **HLI_128_FLOORS = true** command.

4. Click **Save**.

Programming Floor Groups

Floor groups are created to group and identify the floors that are physically accessible by each Otis elevator car used in the integration.

Navigate to **Groups | Floor groups** and create a floor group to represent each of the elevator cars in the Otis system, using the guidelines below:

- Be sure to name the floor groups according to the floors they represent.
- Each floor within the floor groups must be assigned a **Schedule** to determine when it can be freely accessed without credentials.
- If a floor is to be freely accessible at all times, set the schedule to **Always**.

Programming the Elevator Cars as Doors

Each elevator car in the Otis elevator system needs to be created as a door in the Protege GX door programming to identify it as an access point.

To configure a Protege GX door as an Otis elevator car:

1. Navigate to **Programming | Doors** and create a door for each elevator car used in the Otis system.
2. In the **Commands** field of each elevator car **Door** record, enter the following commands:
COP = true
GroupNumber = <value>
ID = <value>
FloorGroup = <value>
3. Click **Save**.

Command	Description
COP = true	Sets the door record to act as an elevator car. Valid values are true (enabled) or false (disabled).
GroupNumber = <value>	Defines the Otis system group number that the elevator car belongs to, from 1 to 8 inclusive.
ID = <value>	Specifies the unique Elevator Car ID within the Otis system, from 1 to 8 inclusive.
FloorGroup = <value>	Defines the Protege GX Database ID of the floor group created above that contains all the individual floors that the Otis elevator car can physically access.

Note: The combination of the **ID** and **group number** in the Otis system must be unique for each elevator car.

Programming Antipassback

Optionally, you can use antipassback with operating panels in HLI elevator integrations. For example, you could enable hard antipassback on a turnstile to prevent users from entering the turnstile then passing their card back to someone on the other side.

The operation is the same as standard door antipassback: the system tracks which area each user is in based on their access activity, and reports an antipassback violation if the user is not recorded in the correct area required to enter or exit the door.

Requirements for Elevator HLI Antipassback

- This feature is available in controller firmware version **2.08.1297 or higher**.
- Antipassback should only be used on turnstiles, security gates or similar operating panels with **entry and exit readers** to record movement into and out of the area. When the user enters the turnstile, the system will update their user area and call an elevator based on their home floor or selection. When the user exits the turnstile, it will update their user area without calling an elevator.
- User area tracking is based on the **inside and outside area** programmed in the door record, **not** the floor that the user has selected. For example, the outside area might be the ground floor reception. When a user accesses the turnstile, it grants entry to the elevator lobby and calls an elevator.

Antipassback should not be enabled for operating panels which are only used to select a floor (such as car operating panels), as the user could be going to any floor instead of entering a specific physical area such as a lobby.

- User credentials must be processed by the Protege GX controller, not the elevator system.
- The **Reader 1/2 elevator** in the reader expander programming must be <not set>.

Programming Antipassback for a Turnstile

1. Navigate to **Programming | Door types** and add a new door type.
2. Set the **Entry passback mode** to control entry through the turnstile and **Exit passback mode** to control exit. The options are:
 - **Hard passback:** Access will be denied if a user attempts to enter/exit the turnstile from the wrong area.
 - **Soft passback:** Access will be granted even if a user attempts to enter/exit the turnstile from the wrong area, but a 'Soft Passback Violation' event will be logged.
3. Optionally, enable **Entry/Exit passback is qualified with door opening**. This will prevent the user's antipassback status from being updated unless they actually pass through the turnstile after badging.
4. Program any other settings required for this door type such as the **Entry/Exit reading mode**, then click **Save**.
5. Navigate to **Programming | Doors** and select a door record which represents a turnstile, security gate or other operating panel.
6. Set the **Door type** to the one programmed above.
7. Set the **Area inside door** and **Area outside door** for the turnstile.
8. Click **Save**.

Programming a Placeholder Elevator Car

A 'placeholder' elevator car is also required to identify all of the floors that will be downloaded by the controller. This must be created as an elevator car record, not as a door record.

Navigate to **Programming | Elevator cars** and create the placeholder elevator that will represent the controller, using the guidelines below:

- The onboard reader expander of the controller used in the integration must be selected as the **Reader expander**.
- The **Reader port** must be set to Port 2 as the controller's transmission port for the integration.
- In the **Schedules and areas** tab add all of the floors that will be downloaded by the relevant controller.
- A **schedule** or **area** does not need to be set for the floors assigned to the placeholder elevator car, and no other configuration is required.

Programming Reader Expanders

Each Otis elevator car has been created as a door record in Protege GX (see page 12). Each door record must then be assigned to a reader expander, thereby connecting the elevator car to the Protege GX access control functionality by associating it to the physical reader controlling access through the door.

If a door record in Protege GX has been configured to represent an Otis elevator car, assigning the door to a reader expander's reader port will allow the reader to be treated as part of the Otis system.

Navigate to **Expanders | Reader expanders** and configure the reader expanders required to connect the Otis system elevator cars, using the guidelines below:

- A single reader expander must be used for every 2 doors (elevator cars) programmed in Protege GX.
- In the **Reader 1/2** tab, assign doors (elevator cars) to **Reader 1 door** and **Reader 2 door**.
- Each reader port must have its **Mode** set to Elevator in order for the connected reader to be treated as part of the Otis elevator system.

Configuring Access Levels

Access levels link floors and floor groups to users, defining which floors the user has access to and when. An access level can contain a single floor, providing users with access to that floor only, or any number of floors as required by the site and user configuration.

1. Navigate to **Users | Access levels** and create or edit access levels.
2. You can assign either individual floors in the **Floors** tab, or floor groups in the **Floor groups** tab.
3. In both cases you can select a **Schedule** which defines when the user has access to those floors.

Important: If schedules are defined in both the access level programming and the floor group programming, these schedules must both be valid for the user to have access to a floor.

Assigning Home Floors to Access Levels

Setting a home floor on an access level is **not** supported by this integration.

Troubleshooting

Too Many Bytes

In some configurations, the Otis MLI may not function correctly because it is receiving too many bytes from the controller for the total number of floors in the system, as the controller assumes by default that there are 64 floors. If this occurs, the following process may resolve this issue.

1. Navigate to **Sites | Controllers** and change the following command:
AEA = false
Enter the following new command:
HLI_MAX_FLOORS = X (where **X** is the total number of floors used in this integration)
2. Click **Save**.
3. Wait for the changes to be downloaded to the controller, or right click on the controller record and click **Force download**.
4. Change the following command:
AEA = true
5. Click **Save** and wait for the change to be downloaded to the controller.
6. Validate whether this has solved the issues with the integration.

The **HLI_MAX_FLOORS** command should not be used unless it is required. It may cause functioning sites to become non-functional.

Incorrect Floor Number in Event Log

In some configurations, the Protege GX event log consistently reports an incorrect floor number when a floor is accessed or unlocked. You may be able to resolve this issue by entering the following command in the programming (**Sites | Controllers**) of the Protege GX controller used in the integration:

AEAFloorOffset = X

where **X** is the floor offset number (between -8 and 8). For example, if Protege GX is reporting a floor number that is one higher than expected, set the command to **-1**.

Designers & manufacturers of integrated electronic access control, security and automation products.
Designed & manufactured by Integrated Control Technology Ltd.
Copyright © Integrated Control Technology Limited 2003-2023. All rights reserved.

Disclaimer: Whilst every effort has been made to ensure accuracy in the representation of this product, neither Integrated Control Technology Ltd nor its employees shall be liable under any circumstances to any party in respect of decisions or actions they may make as a result of using this information. In accordance with the ICT policy of enhanced development, design and specifications are subject to change without notice.