

Interface through TCP/IP (for GestHotel 6.1 & OsAccess)

Following document explains how to make an interface to Omnitec apps that receives some parameters and can issue the cards.

This service has an associated cost. Please, contact your distributor to know the amount.

Omnitec apps interface is fully compatible with **FIAS interface** (Fidelio Interface Application Specification) from **Micros Fidelio**.

This file must be installed with Omnitec apps, in a normal installation and Omnitec apps could be used as normal installation. You must register GestHotel to work with the interface, in other case, you cannot issue new cards.

All information and required files will be provided by Omnitec. In the interface files there is one resident exe file which hear orders in TCP/IP port. You have to copy into GestHotel folder (C:\Omnitec\GestHotel 6.1).

To make tests you can download the latest GestHotel version from internet or you can request an OsAccess demo. Please, contact us for information.

You can download GestHotel manual in different languages from here:

https://drive.google.com/open?id=1_g9ZVRNDQMjS9MN8a0wRbLwqaZ5Vc9x

OsAccess manual is available at the following link:

<https://doc.osaccess.net/>

Additional notes

- In the remote computer is not necessary to have GestHotel installed, only in the front desk computer
- Programs communicate each other using TCP/IP protocol by a port number you have to set up

Register format

- Registers have STX and ETX characters at the beginning and at the end of every register. STX is the chr 02 and ETX is the chr 03.
- Fields are separated by the | character (pipeline, chr 124). At the end of every register always must be a | and a ETX.
- To send parameters you have to know the correct order to send (check PDF file). ALL parameters must be received. If any field has no data or is empty, you must send || (2 pipelines).



Register types

- GHHU -> is the "InterGH2_GuestCard". You must send parameters in the same order as in the PDF file "Interface_INFO_v2_EN.pdf".
- GHRE -> is the returned parameter. Look at the end of the PDF the "Error Code" parameter for complete description. When this parameter returns 25 it means "Incorrect parameter number".
- GHLH -> is the "InterGH2_ReadRoom". You must send parameters in the same order as in the PDF file "Interface_INFO_v2_EN.pdf". Only you need to send first parameter (ReaderNo). The function then will return the rest of the parameters in the following order:

```
GHRE|ErrorCode|RoomNo|DateIn|TimeIn|DateOut|TimeOut|Track2|UserHost|
```

If we want to read also data in track 2 (Track2), we'll need to send "READ" in this parameter, so we'll need to send parameters in this way:

```
GHLH|1|||||READ||
```

The same happens with the parameter UserHost: if we want to tell for which machine (hostname or IP address) we want to read the card, we must indicate in this parameter, so we'll send information in the following way:

```
GHLH|1|||||PC-RECEPTION|
```

Or

```
GHLH|1|||||192.168.1.77|
```

- GHLT -> is the "InterGH2_ReadCard". You must send parameters in the same order as in the PDF file "Interface_INFO_v2_EN.pdf". Only you need to send first parameter (ReaderNo) and optionally the last 2 parameters (UserHost and StringToDecode).

For the parameter "UserHost": if we want to tell for which machine (hostname or IP address) we want to read the card, we must indicate in this parameter, so we'll send information in the following way:

```
GHLT|1|||||||||PC-RECEPTION|
```

Or

```
GHLT|1|||||||||192.168.1.77|
```

For the parameter "StringToDecode": if we want to decode a card read with another encoder, we must send in this parameter, in the same format as received in parameter:

Guest card

Depending on installed software:

- If **GestHotel**: You can issue GestHotel guest cards.
- If **OsAccess**: You can issue OsAccess guest cards.
- If **e-Rent**: You can print the ticket with code to programme the safe

Received parameters

Name	Description	Format	Size	Comments
RoomNo	Room number	Numeric		Between 0 and 9999
DateIn	Check in date	String	8	Format "YYYYMMDD"
TimeIn	Check in time	String	4	Format "HHMM"
DateOut	Checkout date	String	8	Format "YYYYMMDD"
TimeOut	Checkout time	String	4	Format "HHMM"
RoomNo2	2nd additional room	Numeric		Between 0 and 9999
RoomNo3	3rd additional room	Numeric		Between 0 and 9999
SafeBox	Safe box	String	1	Allowed values: "0" (don't have), "1" (have)
CommonDoors	Common areas A and Lifts	String		Binary
CardQty	Number of cards to issue	Numeric		Between 0 and 99If you pass 0 no card issued.
ReaderNo	Reader to use	Numeric		Between 0 and 99

Track2 (Optional)	Track 2	String	48	This parameter is extra PMS information to be recorded in the proximity card
SeqMode (Optional)	Sequence to use	String	1	Allowed values: "N" (new card), "C" (copy card), "D" (default treatment)
ShowMessage (Optional)	Show confirmation msg	String	1	Allowed values: "0" (don't show), "1" (show). Default = "0"
UserHost (Optional)	User/Machine	String		Name of the user or machine who calls the function
UIDCard (Optional)	UID of the card	String		Unique Identification Number of the card in Hexadecimal format (this field is necessary for Osaccess and if the interface is of the "kiosk" type, so that the cards can be recorded by UID) e.g.: "1A2B3C4D"

Returned parameters

Name	Description	Format	Size	Comments
UIDCard (Optional)	UID of the card	String		Unique Identification Number of the card
StringToWrite (Optional)	String to write	String		String to write in the card

Read room number from card

This function is used to read the room number of a guest card.

Received parameters

Name	Description	Format	Size	Comments
ReaderNo	Reader to use	Numeric		Between 0 and 99
UserHost	User/Machine	String		Name of the user or machine who calls the function

Returned parameters

Name	Description	Format	Size	Comments
RoomNo	Room number	String	4	Between "0000" and "9999".
DateIn	Check in date	String	8	Format "YYYYMMDD"
TimeIn	Check in time	String	4	Format "HHMM"
DateOut	Checkout date	String	8	Format "YYYYMMDD"
TimeOut	Checkout time	String	4	Format "HHMM"
Track2	Track 2	String	48	Extra data to be read from the proximity card.
UIDCard (Optional)	UID of the card	String		Unique Identification Number of the card

Read card

Read the card in the encoder and returns some parameters of the card.

Received parameters

Name	Description	Format	Size	Comments
ReaderNo	Reader to use	Numeric		Between 0 and 99
UserHost	User/Machine	String		Name of the user or machine who calls the function
StringToDecode	String data read from card	String		String read from the card in an external reader

Returned parameters

Name	Description	Format	Size	Comments
CardType	Card type	Numeric		Between 0 and 99
CardNo	Card number	Numeric		Between 0 and 99
RoomNo	Room number	String	4	Between "0000" and "9999".
Valid	Is Valid	String	1	Allowed values: "0" (it is not valid), "1" (it is valid)
SecNo	Sequence number	String	10	Between 0 and 9999999999
DateIn	Check in date	String	8	Format "YYYYMMDD"
TimeIn	Check in time	String	4	Format "HHMM"
DateOut	Check out date	String	8	Format "YYYYMMDD"
TimeOut	Check out time	String	4	Format "HHMM"

Track2	Track 2	String	48	Extra data to be read from the proximity card.
UIDCard (Optional)	UID of the card	String		Unique Identification Number of the card
Zones (Optional)	Zones of the card	String		Zone codes to which the card has access. Format: comma separated, eg "4,5"
Floors (Optional)	Floors of the card	String		Floor codes to which the card has access. Format: comma separated, eg "1,2,3"
Lifts (Optional)	Lifts of the card	String		Lift codes to which the card has access. Format: comma separated, eg "2,6"



Received parameters commentaries

- All received parameters are obligatory, except which are indicated as optional.
- Never pass Null values.
- If any String parameter is not necessary or don't know what to pass, use as parameter a length 0 string ("")
- If any Numeric parameter is not necessary or don't know what to pass, use as parameter a number 0

Reader to use

You define in Omnitec apps many readers in which you want to issue cards. With this number you indicate starting by 1, the reader number defined in Omnitec apps. If you pass 0, a dialog to select a reader will be shown.

Common areas A and Lifts

Common areas A are areas to be accessed by the hotel guests (gym, sauna, bar, parking, swimming pool, ...)

The access to these areas is programmed by this entry parameter.

To these areas can access guests (guest card), hotel staff (emergency and service cards) and external services (monouse cards).

To this parameter you must pass N digits binary number: every bit 1 is common areas A allowed from 0 to N-1.

Access to floors of the different lifts are also included in this parameter. If they exist, they are added next to Areas of type A.

For the **FIAS protocol**, this same string will be sent in the KO field of the KR register.

To better understand this parameter, let's see the following examples:

Osaccess:

If we receive "1101" in this parameter, the way to compose it would be the following (looking at checkin page):

Access Common Areas		
Code	Description	<input type="checkbox"/>
1	Main door	<input checked="" type="checkbox"/>
2	Swimming pool	<input checked="" type="checkbox"/>
3	Parking	<input type="checkbox"/>
4	0000 Cylinder test	<input checked="" type="checkbox"/>

GestHotel:

If we receive "101010111101100000001101100000" in this parameter, the way to compose it would be the following:

- Yellow: type A areas that are cylinders. 2 in total.
- Orange: areas of type A. 6 in total.
- Green: lifts. There are 2 lifts (01 and 02), and 11 floors (from 0 to 10).
- The selected value equals 1. The unselected value equals 0.

<input checked="" type="checkbox"/>	Access to the beach	F0
<input checked="" type="checkbox"/>	VIP Lounge	F0
<input checked="" type="checkbox"/>	Safe box	
<input type="checkbox"/>	Garage	
<input checked="" type="checkbox"/>	Swimming pool	
<input type="checkbox"/>	Gymnasium	
<input checked="" type="checkbox"/>	Sauna	
<input checked="" type="checkbox"/>	Bar	

Lift	0	1	2	3	4	5	6
<input checked="" type="checkbox"/> Receptor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Rooftop	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

The result would be the following:

101010111101100000001101100000

- * In **dark green**: it is the lift 01 "Receptor"
- * In **light green**: it is the lift 02 "Rooftop"

Track 2

This parameter is optional.

This parameter is extra PMS information to be recorded in the proximity card. You can record up to 48 bytes.



Sequence to use

This is the Sequence number to store in the card.

Allowed values:

- "N" (new card): the recorded card contains the next sequence, so previous recorded cards will NOT OPEN the door.
- "C" (copy card): the recorded card contains the last sequence, so previous recorded cards will OPEN the door.
- "D" (default treatment): shows a dialog to select "New" or "Duplicate" the card. It will not show any dialog if "Guest sequence control" is not activated.

Show confirmation message

Indicates if you want to show/don't show confirmation message before encoding the card.

Allowed values:

- "1" (show): This is the default value. It will show confirmation message before starting to write all the cards. There are 2 dialog windows:
 - Reader shows a dialog "Insert the card" with 2 buttons "Accept/Cancel"
 - Reader shows a dialog during 10 seconds "Slide/insert the card" with one button "Cancel", the reader waits 10 seconds until a card is used.
- "0" (don't show): It will not show the confirmation msg. It will start directly to write the card.
 - For reader which shows a dialog "Insert the card" with 2 buttons "Accept/Cancel"
 - This dialog is hidden if only one card is recording.
 - This dialog is shown in all the cards, if more than one card is recording.
 - For reader which shows a dialog during 10 seconds "Slide/insert the card" with one button "Cancel", this dialog is hidden
 - This dialog is hidden if only one card is recording, but the reader still waits 10 seconds until a card is used.
 - This dialog is shown in all the cards, if more than one card is recording.



User/Machine

Name of the user or machine who calls the function.

StringToDecode

String data read from the card in an external reader. String to decode and return related parameters: room, dates, validity, etc.

It must be sent in the same format as ***StringToWrite*** returned parameter.

UIDCard

Unique Identifier of the Card, **in Hexadecimal format**. e.g.: "1A2B3C4D"



Returned parameters commentaries

All the functions return the same parameter:

Name	Format	Size	Comments
Error code	Numeric		Between 0 and 99

In addition to this, the ReadRoom function (read room number from card) returns the number of the room and the in and out date/time of the card read.

Possible values of the parameter “error code”

0	No error. All the cards have been read/written correctly.
1	Some of the cards have not been read/written correctly.
2	No error. All the cards have been read/written correctly, but you have less than 15 days to register GestHotel.
3	Some of the cards have not been read/written correctly, and you have less than 15 days to register GestHotel.
4	Test time of GestHotel has finished. None of the cards will be read/written.
5	Hard disk serial number is not the same as stored in database. None of the cards will be read/written.
20	<ul style="list-style-type: none"> • ReadRoom function: No guest card • ReadCard function: Unknown card
22	You have less than 15 days to register GestHotel and: <ul style="list-style-type: none"> • ReadRoom function: No guest card • ReadCard function: Unknown card
25	Incorrect number of parameters
26	Time Out when encoding card (no key pressed in message window)
27	Lost connection to database
28	Cancel pressed when encoding card (in message window)
29	Lost connection to encoder client
30	Cannot connect to encoder client
31	Unusable card: Card UID is included in previous blacklist
32	Reservation has not valid room
33	Device is not intialized
34	Unknown protocol
35	The reservation or QR are cancelled
102	Timeout (Is there a card on the encoder?)
105	No card or invalid card
400	Bad Request (function not created correctly)
401	Unauthorized
404	Not Found

422	Unprocessable Entity (Device not initialized?)
99	Unspecified error (look for into log files, to see information about this error)

Track 2

This parameter is optional. You must send the string READ for returning the info on this parameter, otherwise it will be empty ("").

This parameter is extra PMS information to be read from the proximity card.

This option is available for all proximity encoders. If you are using one of these encoders, you can read up to 48 bytes from the card data, if they have been recorded previously in the "Track2" parameter of the function "GuestCard".

Card type

It returns a numeric code that indicates what type of card we are reading. Possible values are:

Smartpass / Activa	
Type	Card name
0	Unknown
1	Guest
2	Meeting
3	Emergency
4	Service
6	Security Master
7	Audit trail
8	Time
11	Blocking
12	Security Guest
22	Installation
23	Test
24	Initialization

OsAccess	
Type	Card name
0	Unknown
1	Guest
2	Staff/Master
3	Setup



Card number

If service or emergency card, it returns the internal identifier code of the card.

Room number

The room number read from the card. It's only returned if Guest or Meeting room card.

Is Valid

It checks dates and sequence number to return 1 if it is a valid card or 0 if it is an invalid card.

Sequence number

Sequence number encoded in the read card. Only valid for Guest, Service, Emergency and Meeting room cards. The rest of the cards do not return sequence number.

Check in date

Check in date encoded in the card. Only valid for Guest, Service, Emergency and Meeting room cards. The rest of the cards do not have this value.

Check in time

Check in time encoded in the card. Only valid for Guest, Service, Emergency and Meeting room cards. The rest of the cards do not have this value.

Check out date

Check out date encoded in the card. Only valid for Guest, Service, Emergency and Meeting room cards. The rest of the cards do not have this value.

Check out time

Check out time encoded in the card. Only valid for Guest, Service, Emergency and Meeting room cards. The rest of the cards do not have this value.

UIDCard

Unique Identifier of the Card, **in Hexadecimal format**. e.g.: "1A2B3C4D"

StringToWrite 1

It is the string to be encoded in the card using another encoder no-compatible with Omnitec encoders.

The format of this parameter is as follows:

; S, B, AABBCCDD; S, B, AABBCCDD; ;

Where:

; → Block separator

, → Field separator into every block

S → Sector number (decimal value)

B → Block number (decimal value)

AABBCCDD → Data to be encoded in the block indicated by S and B. This string is in Hexadecimal representation. This string is encrypted. To decrypt it, please ask to Omnitec the required information.

StringToWrite 2 *

Deprecated

** This option can only be applied to bluetooth key for Activa v2 locks.*

~~It also can be the hex string to be encoded into an oak2 file (bluetooth key for Activa v2 locks). To be returned by the interface program, you must send the string "file:" into this parameter when requesting a guest key. It will only return the string and no other action will be done (like encoding a card).~~

The format of this parameter is as follows:

AABBCCDDAABBCCDDAABBCCDDAABBCCDDAABBCCDD... .

Where:

AABBCCDDAABBCCDDAABBCCDDAABBCCDDAABBCCDD → Data to be encoded in the oak2 file. This string is in Hexadecimal representation. This string is encrypted. To decrypt it, please ask to Omnitec the required information.

StringToWrite 3 **

*** This option can only be applied to Rent&Pass type locks with support for Bluetooth keys.*

Another option in this parameter is to generate a mobile key and send it by email to a recipient. In that case, the string to send in this parameter would be: "upkey: name@example.com: 8", where:

- upkey: indicates that the mobile key will be generated
- name@example.com: would be the recipient's email, who will receive the key
- 8: the template that will be used to send the email. These templates must be previously created in GestHotel or OsAccess. In the image, examples of templates defined in GestHotel.

Email templates	
Code	Description
1	ES - Email con llave para huespedes
2	EN - Email with guest key
3	FR - Email avec la clé de l'hôte
4	ES - Email con codigo de acceso
5	EN - Email with access code
6	FR - Email avec le code d'accès
7	ES - Email con UPKEY para huespedes
8	EN - Email with guest UPKEY
9	FR - Email avec UPKEY de l'hôtel
10	DE - eMail mit eSchlüssel - Upkey
11	DE - eMail mit Zugangscode
12	DE - eMail mit Gästeschlüssel - Activa K

StringToWrite 4 ***

*** This option can only be applied to Rent & Pass type locks with access code support.

Similar to the previous option, this parameter can be used to generate an access code or passcode and send it by email to a recipient. It is also possible to return it in the response command. The possible options of this parameter are the following:

```
passcode:recipient:template[:length]
```

Where:

- Passcode: indicates that the mobile key will be generated
- Recipient: would be the recipient's email (name@example.com), who will receive the passcode. If you want to return the code in the response, "return" will be indicated here.
- Template: the template that will be used to send the email. These templates must be previously created in GestHotel or OsAccess. In the image, examples of templates defined in GestHotel.
- Length: is an optional parameter that indicates the length of the code that will be generated. The admitted values are 4 or 6. To be able to generate codes of this size, the hotel must have Gateways to be able to generate this type of code.

Email templates	
Code	Description
1	ES - Email con llave para huespedes
2	EN - Email with guest key
3	FR - Email avec la clé de l'hôte
4	ES - Email con codigo de acceso
5	EN - Email with access code
6	FR - Email avec le code d'accès
7	ES - Email con UPKEY para huespedes
8	EN - Email with guest UPKEY
9	FR - Email avec UPKEY de l'hôtel
10	DE - eMail mit eSchlüssel - Upkey
11	DE - eMail mit Zugangscode
12	DE - eMail mit Gästeschlüssel - Activa K

Return parameter:

```
# ; D, C ; D, C ; ...
```

Where:

→ start of return codes

; → block separator

, → field separator within the block

D → Description of the location (room number, common area name, etc). This parameter comes in hexadecimal format ("0101" → "30313031")

C → code to enter. This parameter comes in hexadecimal format ("123456" → "313233343536")

Notes:

This output parameter can also include the format explained in "String to record 1", in this case the format would be:

`; S, B, AABBCDD; S, B, AABBCDD; ; #; D, C; D, C; ...`

Examples:

"passcode:name@example.com:5" → generates a passcode and sends it by email to name@example.com using template 5

"passcode:return:0:6" → generates a passcode that will be returned in the response. The template is at 0 because in this case no template is used, the parameter sent here will not be used. The code size will be 6 digits. Return parameter example:

`#; 30333032, 343238383331; 4D61696E20646F6F72, 343238383331;`

You can combine bluetooth keys and codes in the same command, separating by the semicolon character ";", for example:

`upkey:name@example.com:8;passcode:name@example.com:5`

