

CID Messenger v1.0 Technical manual

© JSM Micro Inc, by J.F. Samson



1.0 Overview

Several kind of telephone emergency calling devices exist on the market today. Majority of them use analog lines and dial pre-programmed phone numbers to report previously recorded alarm messages to users. These units can dial several times the same number and escalate to a second number until somebody answers the call and dial a code to reset the unit or hang-up (if the unit can detect the disconnect signal) or until there is a timeout if nobody answers.

Some units (dialers) are designed to use public analog lines (POT), others are designed to work behind a PBX and exploit PBX features like phantom DN, hunt groups and call forwarding. These dialers are installed in homes, aged people apartment buildings, care facilities and even in hospitals.

CID Messenger is a firmware who interpret and react to caller-id information. Using JSM Config you can program the WSX1 CPU unit to decode caller-id information and do the following:

- Determine call types like bathroom, pendant or bed.
- Identify the caller and rename it if you wish. (102, Room 102...)
- Dispatch the call to pagers and scrolling displays.
- Do call escalation in case of no response.
- Accumulate calls information for reporting & statistics.

Applications are limited by your imagination and the equipment you will use. In the following pages we will present different scenarios and explain how to program the equipment.

1.1 Caller-id decoders

You can use two models of caller-id decoders. The basic one (Whozz Calling POS) will extract the caller's phone number and name. The other model (Whozz Calling) will also give on-hook, off-hook indications, call duration and is able to decode DTMF digits dialed for the first 60 seconds of the call beginning. Those units does not pick-up, go off-hook or answer the telephone line. They do not affect the operation of any other device connected to the telephone line.

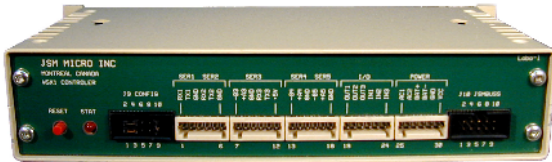


These caller-id decoders exists in three models:

- 2 lines (WCPOS-2 or WC-2)
- 4 lines (WCPOS-4 or WC-4)
- 8 lines (WCPOS-8 or WC-8)

1.2 WSX1 CPU unit

This CPU module connects to caller-id decoders, to a computer for initial programming and reporting purpose, to paging systems and to scrolling displays. The WSX1 can also store up to 2000 calls in its internal RAM memory. When correctly programmed the following caller-id information will be available for programming:



- Line number (01-32) on which the caller-id information is present.
- Start or End of call (WC-x units only)
- DTMF/Hook information (WC-x units only)
- Caller's phone number
- Caller's name

2.0 Wiring & hook-up

Connect the Whozz Calling decoder to the WSX1 ser3 port using the supplied serial cable. If you have more than one caller-id decoder module connect them together using supplied RS232 cables. See caller-id decoder technical manual for more information on dip switches settings.

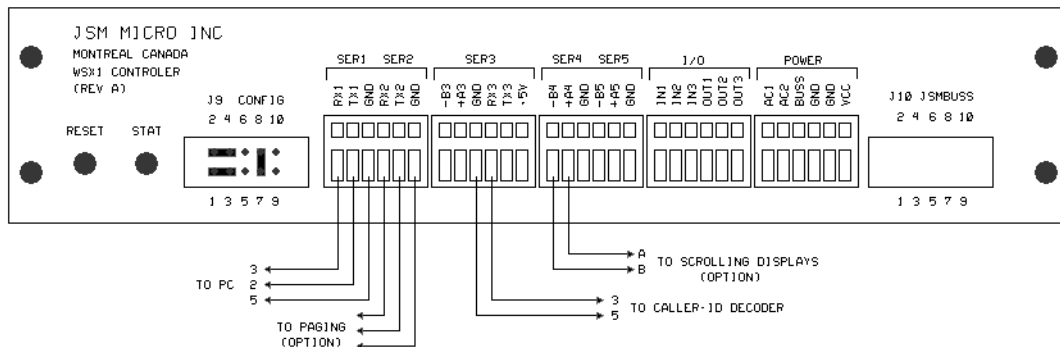
Whozz Calling to WSX1 serial DTE cable pinout:

DB9 pin 2 (tx out) red to RX3 of WSX1
 DB9 pin 5 (gnd) black to GND of WSX1

Connect your PC serial port to the WSX1 ser1 port using supplied serial cable. This cable will be used to program the unit, to debug your programming and to connect with JSM Report software for calls statistics and reporting.

WSX1 (programming) to PC serial cable pinout:

DB9 pin 3 (tx out) red to RX1 of WSX1
 DB9 pin 2 (rx in) white to TX1 of WSX1
 DB9 pin 5 (gnd) black to GND of WSX1

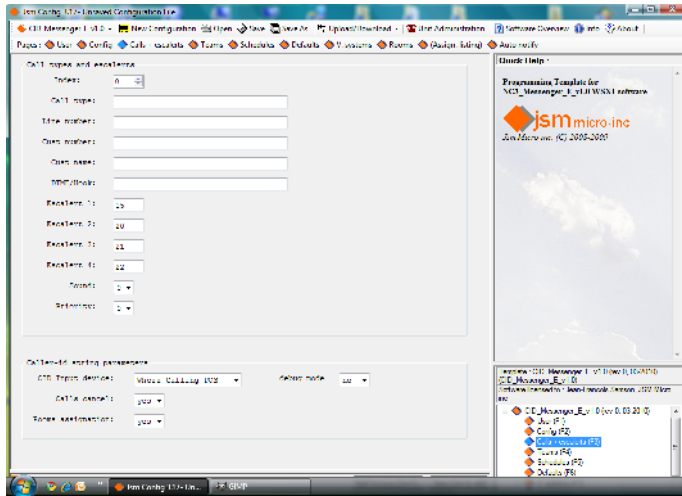


WSX1 wiring

3.0 Programming

Before programming the unit, lets discuss on what is needed and how to do it. As previously described, the caller id decoder will give us:

- Line number on which caller-id information is present
- Customer phone number
- Customer name
- Deluxe models also give:
- Start or End of calls
- DTMF digits or on/off hook condition (we only use on/off hook signal)



Calls + Escalerts section.

In the JSM Config software we have a section named "Calls + Escalerts". In this section we program caller-id parameters for each type of calls we need to detect, assuming the dialers will dial different numbers to identify different call types.

4 fields are important to decode calls:

- Line number
- Customer's phone number (DN)
- Customer's name
- Hook condition (for some applications)

For example, if you have a PBX programmed with phantom DN's (to identify calls) and hunt groups, the information needed will be available in the customer's name field.

Assume extension 201 has "Rm 21" name and phantom DN 600 has "Bed Call" name. When patient press his bed call button, the dialer goes off-hook and dials 600. The call is immediately forwarded to a hunt group and one line or several lines of this hunt group are connected to the caller-id decoders. Other lines are connected to IP phones. All hunt group phones will ring and the decoder will output the ringing line number, **201** as customer number and "**RM21>Bed Call**" as customer's name. If the unit is programmed in the following way:

Index:	00	
Call Type:	BED	<---- call type we want to define
Line Number:	ALL	<---- this field match on any line number
Cust Number:		<---- nothing to match here
Cust Name:	>Bed Call	<---- this field also match (" RM21>Bed Call ")
DTMF/Hook:		<---- nothing to match here

The string ">Bed Call" will be removed from received caller-id name field, leaving "**RM21**" as room name. So, the call type will be "**BED**" and the room name will be **RM21**.

Now, when the user call is answered or if the user press his call cancel button, the dialer will hang-up, wait, then go off-hook and dial a hunt group DN to signal the end of call.

Again, we will receive the information on one line of our caller-id decoder module. Assume we have the following programming:

Index: 00	Here we have defined our first call type
Call Type: BED	Its name is: BED
Line Number: ALL	It can be detected on all lines
Cust Number:	
Cust Name: >Bed Call	Customer's name must contains >Bed Call
DTMF/Hook:	

Index: 01	Here we define our next call type
Call Type: *CANCEL	Its a call cancel type (because it starts by *)
Line Number: ALL	It can be detected on all lines
Cust Number:	
Cust Name: *	Anything in customer's name will be valid
DTMF/Hook:	

How it works:

When the user press his call button, the dialer go off-hook and dial 600 which is the phantom DN named "**Bed Call**". This cause the caller-id string "**RM21>Bed Call**" to appear as customer's name. The CID Messenger firmware will identify the call as a "**BED**" call from "**RM21**". When the user press his cancel button, the dialer hang-up, then goes off-hook and dial the hunt group DN (not the 600 number). The call is received with the customer's name "**RM21**" only. The firmware will try to match at index 00 first and it will not match (because it does not contain >Bed Call). The firmware will then try to match at index 01. This time it will match because anything is accepted as customer's name. So, the call type will be **CANCEL** and the room will be "**RM21**".

The firmware will now search if there is a pending call for RM21 and will find it. It will then delete the call and log its start and end time for later retrieval by JSM Report.

At the end of the manual there is a section containing different scenarios and explain how to program and also supply programming and test files for each scenarios. The test files contains data "as sent" by caller-id decoder. You can send these test files from your PC using HyperTerm or TeraTerm (<http://www.ayera.com/teraterm/>). Simply connect your PC serial port to WSX1 ser3 port (RX3 & GND). Use 9600 N 8 1 as serial port settings, no hardware handshake is needed.

The debugging mode:

In the Calls + Escalerts section you can turn on the debugging mode. This will allow debugging information to be sent on ser1 port. You will be able to see what the WSX1 receive and how it is interpreted by the firmware and what actions are taken.

When finished, turn-off debugging mode as it may interfere with JSM Report software.

Typical debugging information:

```
valid cid string received
line number= 05
cust_number= 770-263-7111
cust_name= Samson J.F.
call duration= 0000
searching in cust_number field.. (01)
room= 770-263-7111
call type= A0EMERGENCY CALL
New call..
no room assignation, default to index 0
stored new call= A0EMERGENCY CALL 770-263-7111 (SYS assigned)
proceed to page the event..
```

```
valid cid string received
line number= 06
cust_number= 770-263-7111
cust_name= Samson J.F.
call duration= 0000
searching in cust_number field.. (01)
room= 770-263-7111
call type= A0EMERGENCY CALL
New call..
no room assignation, default to index 0
deleting previous event: A0EMERGENCY CALL 770-263-7111
stored new call= A0EMERGENCY CALL 770-263-7111 (SYS assigned)
proceed to page the event..
```

```
valid cid string received
line number= 07
cust_number= 770-263-7111
cust_name= Samson J.F.
call duration= 0000
searching in cust_number field.. (00)
room= 770-263-7111
call type= A0*CANCEL
End of call..
no room assignation, default to index 0
deleting & logging event: A0EMERGENCY CALL 770-263-7111
```

This debug information was generated from case 1 scenario:

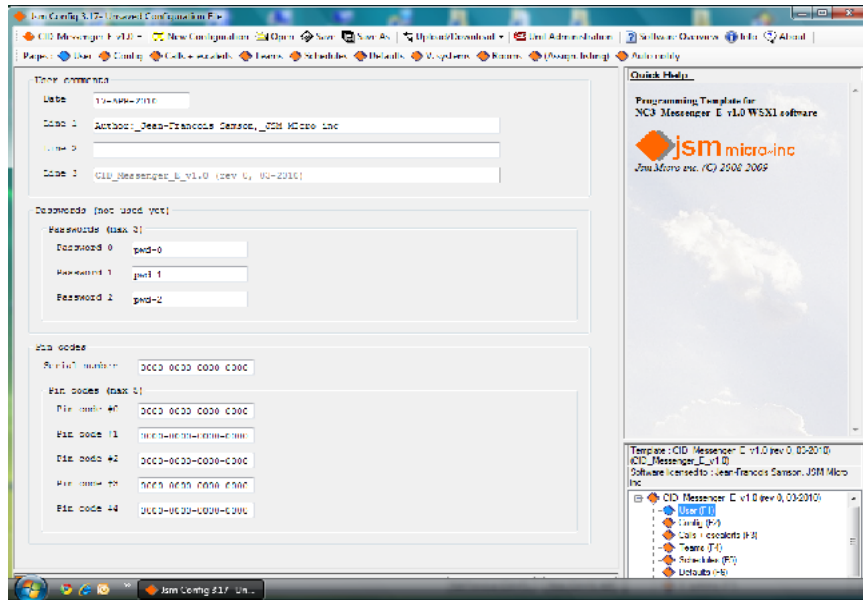
[6 C.O. lines for calls, 2 C.O. lines for cancels (Whozz Calling POS)]

Test file sent on serial 3 port:

```
05 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Samson J.F.
06 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Samson J.F.
07 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Samson J.F.
```


4.0 Programming software

Open JSM Config, load CID_Messenger_E_v1.0 template file. This window will appear:



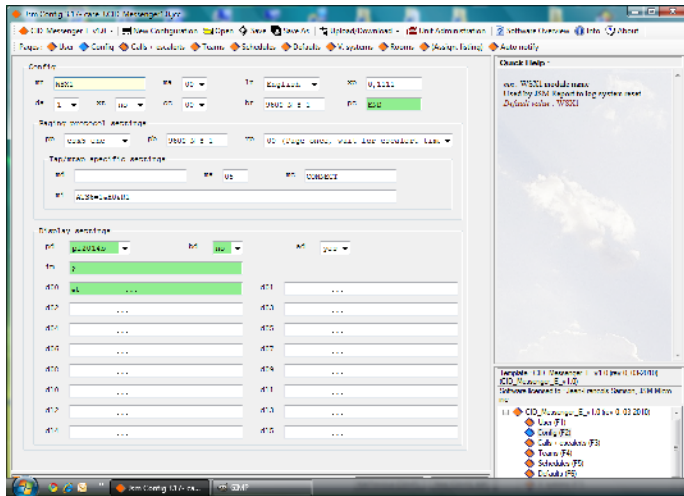
From there, you have three choices. Selecting from top menu you can:

1. Select New Configuration and start to fill the form.
2. Open an existing file saved to your PC.
3. Download an existing file from WSX1 unit.

You can navigate and fill different programming pages by selecting them from the second top row. The current displayed page is represented by a blue lozenge icon while the others are orange. Alternatively you can select pages by pressing functions keys (F1-F10) or by selecting them from the bottom right window pane. As you can see, you have ten pages to program the unit:

User	Enter personal data and features activation pin codes
Config	Mainly specify if you have paging and scrolling displays
Calls + Escalerts	Where you program different call types to decode
Teams	Fill them with pagers, displays and wireless phones
Schedules	If you have calls assignation based on work shifts, define work periods.
Defaults	Repetitive default programming is entered here if needed
V. Systems	You can define virtual systems (nurses call systems)
Rooms	You can program specific rooms assignations and parameters
Assign. Listing	Used only to show programmed and renamed rooms in one view
Auto notify	To program automatic messages for pagers, repeated every day.

4.1 Config page



Quick help window:

When you select a field with your mouse, its description appears in the upper right window pane.

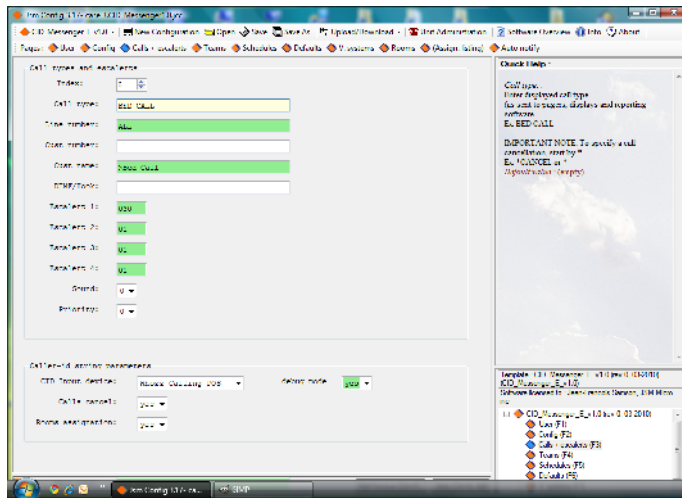
In this page you give a name to the WSX1 module, select system language, determine if we send calls cancellations to pagers, select the paging system and its parameters and program defaults messages for scrolling displays.

Fields ma, cc & xp are not used. Please leave them to their default values.

Fields description:

name	Description
mn	Give a name to the WSX1 module. Reporting software use it to report resets.
ln	Select system language. Mainly used when displaying date on scrolling displays.
ds	Select daylight saving mode.
xt	To append current time/date to pager messages. Useful for debugging.
br	Caller-id decoders serial port settings. (Whozz Calling use 9600 N 8 1)
pc	For pagers: Appended to cancelled calls. Ex: END (Leave empty for no paging)
pp	Select paging protocol. Default to csx5-enc. (Pin code needed for others)
pb	Paging system serial port settings.
rp	Chose paging repetition rate for pending calls. (Repeat until first escalate)
md	If you have selected mtap as paging protocol enter a phone number for the modem.
ms	Specify how many paging transactions can be made on a single modem call.
mc	Connection string returned by modem. Do not change unless instructed to do.
mi	Modem initialization string. Do not change unless instructed to do.
pd	Scrolling displays protocol. (Pin code needed to use displays)
bd	Specify if displays will beep on calls.
ad	Specify if displays are addressable.
im	Your company name displayed at power-up. (enter x for nothing)
d00	Display 00 message to display when no calls. (scrolling dots or time & date..)
d..	You can have a message for each display address. (Several displays can share the same address) Displays addressing allow calls to be dispatched by floors or areas.

4.2 Calls + Escalerts page



This page is divided in two sections:

Call types & escalerts: Where you specify each call type parameters needed to define and validate the call.

Caller-id parameters: Where you specify the model of caller-id decoder used, specify if you have calls cancellation, if you have rooms assignation (for pagers & displays). You can also turn on/off debug mode.

Call types & escalerts parameters:

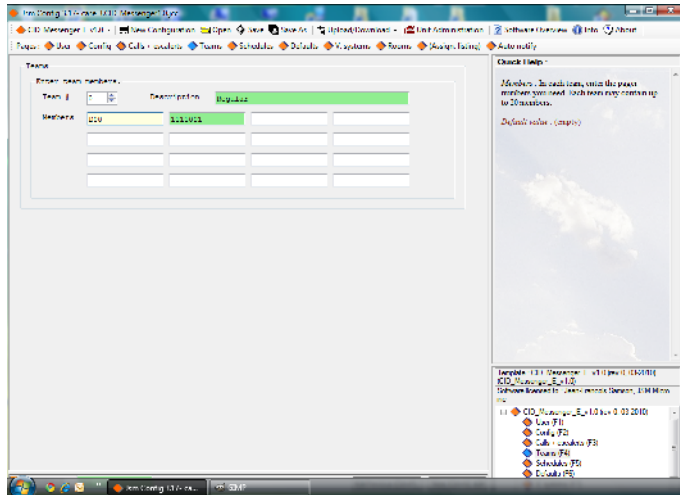
name	Description
Index	Its a program index. You have one index per call type, starting from 0.
Call type	The name of the call as you wish it to appear on pagers or displays.
Line number	The line numbers range on the caller-id decoder where the call information should appear. Ex: 01-03 or ALL
* Cust number	Enter a specific number you wish to match. Usually the wildcard chr * is used for that field (also with rooms assignation)
* Cust name	Enter a specific string to be found in the customer's name caller-id data. Ex: >Bed, Ex: >Bath Usually this field is used with PBX specific programming (phantom DN...)
DTMF/Hook	Some information may be needed here for some calls cancellation detection. Actually DTMF information is not used but you can use "ON-HOOK" here.
Escalerts 1-4	Enter the escalert time in seconds (3 digits) or in minutes (2 digits) for this call type. Escalert is used with pagers for unanswered calls.
Sound	For calls sent to displays, you can program a beeping pattern here.
Priority	- This field is presently unused -

* These fields are mutually exclusive. You cannot use both of them.
You can validate a call type using customer's phone number or customer's name.

Caller-id parameters:

name	Description
CID input device	Choose which model of caller-id device is used.
Calls cancel	Select "yes" if you are able to detect calls cancellations from caller-id. If you cannot detect cancellations, calls will be dispatched to pagers immediately upon reception and no escalerts will be available. Reporting will be available with calls duration= 0 minutes.
Rooms assignation	If you select "yes" you will be able to rename rooms, do specific rooms to pagers assignations, convert customer's phone numbers to apartment or room numbers ... If you select "no" the received information will be used and default virtual system 00 assignation will be used.
Debug mode	Will cause debug information to be sent on WSX1 ser1 port. This can help to program the unit for complex situations. Please turn-off that feature when debugging is finished as it will interfere with JSM Report reporting software.

4.3 Teams page



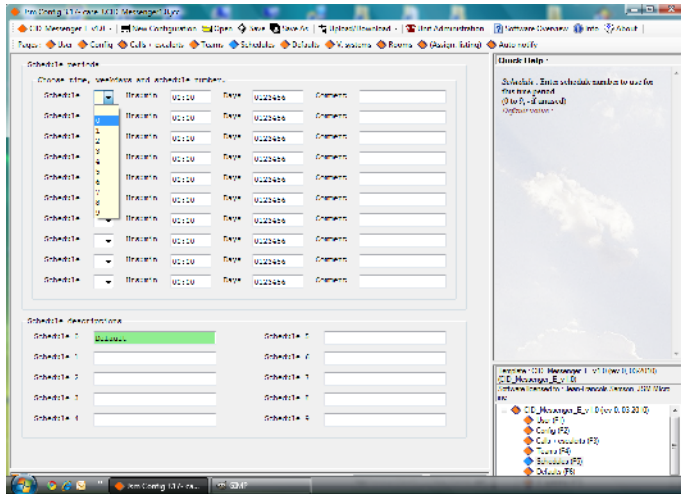
When a call is detected CID Messenger can send it to a team. Teams contains pagers, displays and Kirk wireless Spectralink phones.

CID Messenger allow up to 100 teams and each team can contain up to 20 members.

You can also give names to teams for ease of programming.

Team member	Syntax to use
Pager	For CSX5-ENC: Enter its 7 digits capcode. Ex: 1111001 For others: See their instructions manual
Display	Enter "D" followed by its 2 digits adress Ex: D00, Ex: D01 ...
Spectralink phone	S-0tr-Uxxx Ex: S-004-U001 or S-004-308 (ext 308) t= type (0 for page, 1 for page with call back) r= how it will ring (1-9 see WSX1 SPLNK manual) U= user number as programmed in WSX1-SPLNK2 unit

4.4 Schedules page



If you have assignments based on work shifts, you must define schedules. Otherwise leave this page empty.

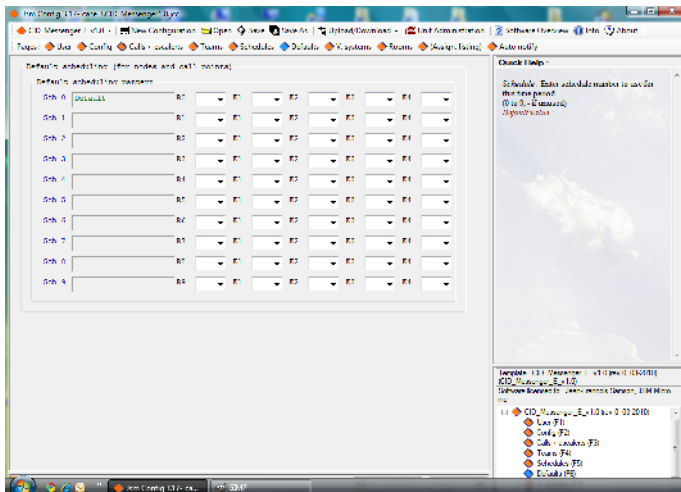
A schedule is a number (0-9) validated by a starting time (hr:mm) on weekdays (0-6, 0 being Sunday)

Once started, a schedule is valid until another one is started.

For ease of programming you should give names to your schedules.

See programming examples files to see how we can use and program scheduling. The configuration file is named: "Scheduling example."

4.5 Defaults page



This page is there to help you program repetitive assignments.

For example:

If your application is based on customer's phone numbers and you wish to use their apartment numbers instead, therefore must have room assignment turned on and you must program every tenant's phone number and write their corresponding apartment number. You also wish to send all calls to the same pagers,

based on 3 work shifts. You can then use the default page to program your assignments for the 3 schedules. When you will program the rooms (see Rooms page) the firmware will use default values programmed here every time a question mark (?) appears on the page. This will be done for all rooms.

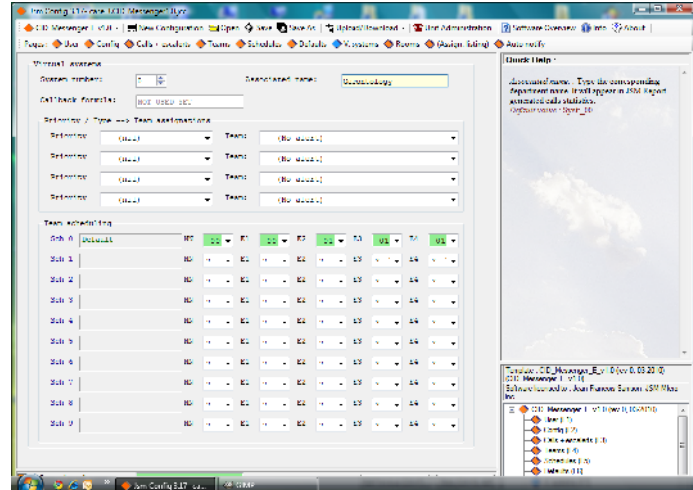
4.6 V. Systems page

The concept of Virtual Systems origins from nurses call systems programming and you may wish to apply this concept to your application because you want to group your customers by sections, areas or floors.

You program your assignments by systems rather than by rooms. The only requirement is when you program your rooms, you must assign them to a virtual system.

Note: If your application have no rooms assignment, virtual system 00 assignment will be used by default.

Virtual systems are numbered from 00 to 14 and you can name them.

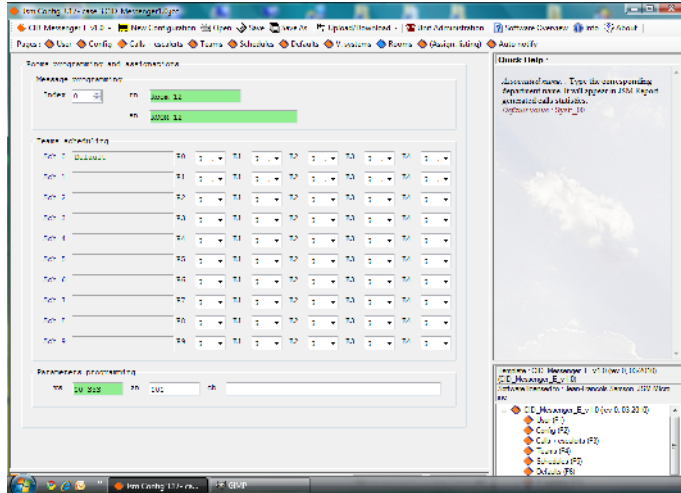


Programming fields description:

name	description
System number	Chose the system number you want to program.
Associated name	Give it a name. Ex: First floor, East building...
Priority + Team	In these four lines, you can select a previously defined call type and assign it to a team. This is useful if you wish to assign different pagers for a specific type of call. As example, maybe you wish to alert a different team for EMERGENCY CALLS.
SCH 0-9	This field will show you all your previously defined schedules. As you can see, you will be able to do assignments for each schedule number.
NN	This is the team to alert when there is a call from that system.
E1	This is the first escalert team. Remember, escalert times are based on call types and are programmed in the Calls + escalerts page.
E2	Escalert 2 team
E3	Escalert 3 team
E4	Escalert 4 team. This will be the last team alerted.

4.7 Rooms page

If you have selected rooms assignment, you must program all rooms from that page. The firmware can handle up to 1024 rooms (0000-1023).



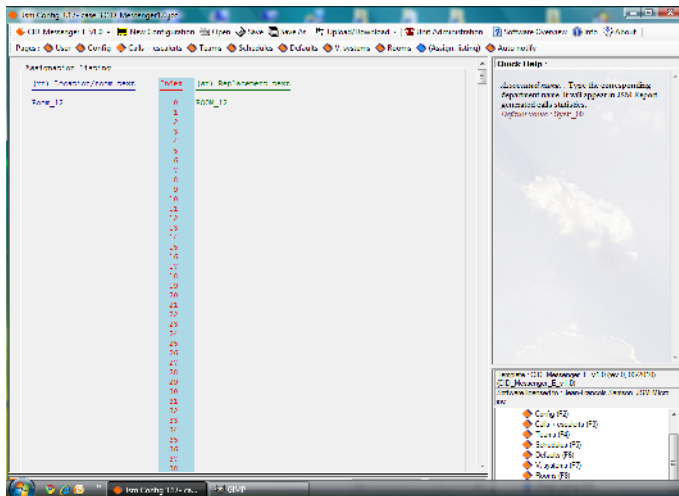
Select room to program using the index field. Start from index 0000 growing up.

The page is divided in three sections:

- Messages programming
- Teams scheduling
- Parameters programming

name	description
rn	Room name (or number) as extracted from caller-id decoder and after call type extraction. Ex: 514-355-5731, Ex: Rm 12 ...
sn	Substitution name: You can rename the room. Ex: for rn 514-355-5731 you can use: "APT 609" or "609 – John Smith"
SCH 0-9	Same as in V. systems page description.
NN	Same as in V. systems page description.
E1-4	Same as in V. systems page description.
vs	Select virtual system number and add "SYS" if the assignation is made at the system level rather than room level. Ex: 00-SYS, 00- ,
zn	Optional zone number (may be useful for reporting purpose)
cb	Call back formula: to be used for Kirk or Spectralink options. This will allow a Spectralink wireless phone to dial and reach the customer. This may be used for some applications but since the call is answered by a PBX, this PBX can forward it directly to the wireless phone.

4.8 Assign.listing page

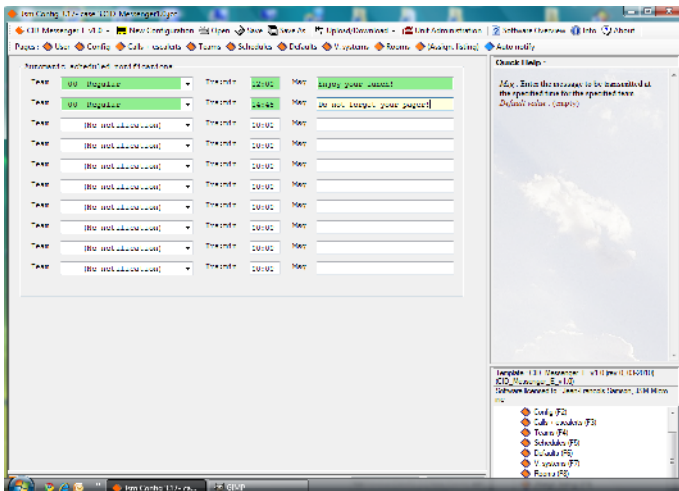


This page will simply list you all programmed rooms and their corresponding replacement names.

Use scrolling bar to scroll to the end.

Its just a convenient way to rapidly see if a room is programmed.

4.9 Auto notify



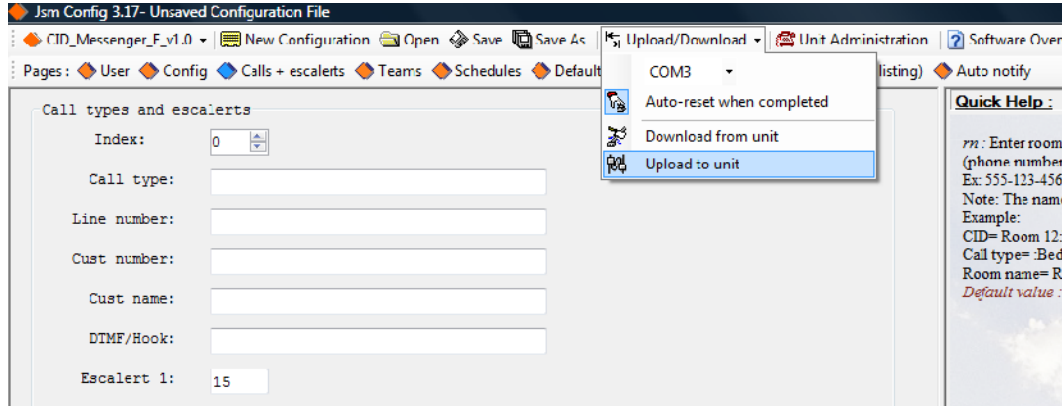
In this page you can program up to ten messages to send at pre-programmed hours to specific teams.

These messages maybe reminders to not forget to remit the pager at the end of work shift or maybe to signal breaks or lunch time.

Fields description:

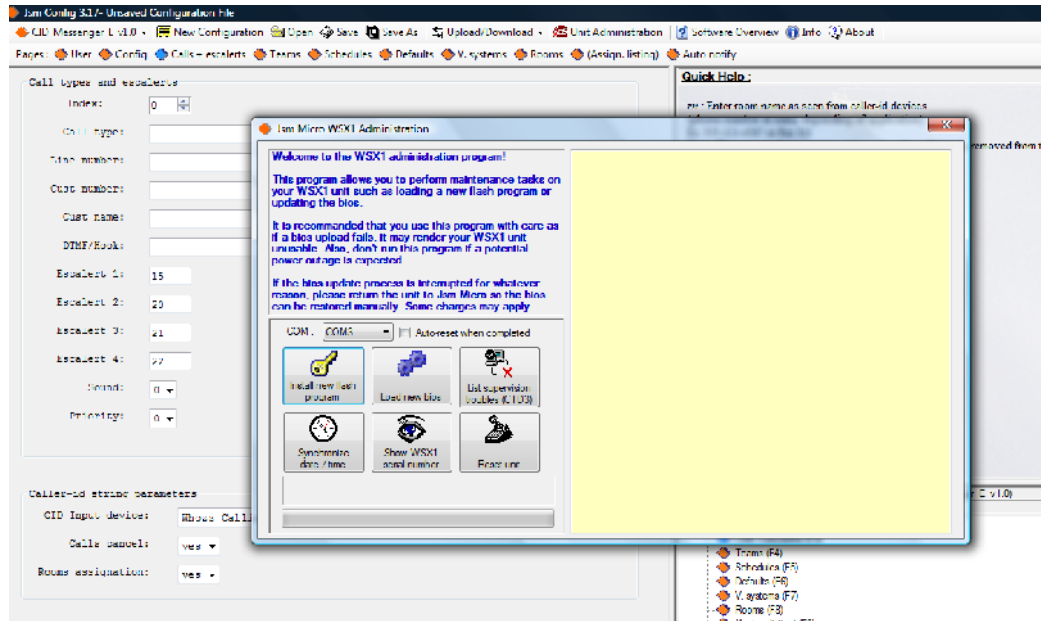
name	description
Team	Select a destination team for your message.
Hrs:min	Set at which time message will be sent (daily)
Msg	Type the message.

5.0 Programming the WSX1



When programming is finished, save your file to your PC then select “Upload/Download” from the top menu row. Select your serial port then verify the WSX1 is connected to your programming cable on ser1 port. Then select “Upload to unit”.

5.1 Administrating the WSX1



Access this utility program from top menu row. It will allow you to install new flash programs or bios, synchronize the unit real time clock to your PC internal clock. You can also display unit serial number.

6.0 Applications scenarios

Whozz Calling 4 or 8 caller-id output format description

```
Line Number          Cust Number          Cust Name (15 max)
'                    ' (14 max)          '
'                    '                    '
01 I S 0000 G A0 09/26 11:28 AM 770-263-7111  CALLERID.COM
```

The caller-id decoder will decode standard North American caller-id signals from Bellcore 202 signaling protocol. Belcore 202 signaling is sent between the first and second ring in the countries such as the US, Canada, Mexico, Australia and others.

Case-1: 6 C.O. lines for calls, 2 C.O. lines for cancels (Whozz Calling POS)

```
01 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | call input line
02 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | call input line
03 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | call input line
04 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | call input line
05 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | call input line
06 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | call input line
07 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | cancel line
08 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.      | cancel line
```

Lets program cancels first in the calls list and set the cancel attribute '*' on Call-Type field (remember: lines 07 and 08 are used to receive cancellations)

```
Index: 00          |
Call Type: *CANCEL | * means its a cancel, word CANCEL is useless
Line Number: 07-08 | call must be between lines 07 and 08
Cust Number: *     | Match on any value on that field
Cust Name:         | field not used
DTMF/Hook:        | field not used
```

If the cancellation is not found (because not on lines 07 or 08), search continue until a match occurs on lines 01 to 06.

```
Index: 01          |
Call Type: EMERGENCY_CALL | Call type used on pagers, reporting
Line Number: 00-06      | on lines 00 to 06 included
Cust Number: *         | Match on any value on that field
Cust Name:             | field not used
DTMF/Hook:            | field not used
```

Test file used:

```
05 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.
06 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.
07 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.
08 I S 0000 G A0 09/26 11:28 AM 770-263-7111  Samson J.F.
```

We simulate two calls from the same customer (on lines 05 and 06). The first call is stored, the second call cause deletion of first call because we cannot have two pending calls for the same room. This second call is then stored. Note that second call (line 06) could have been programmed to cause a higher priority call.

We simulate two cancellation calls from the same customer (on lines 07 and 08). The first cancellation call is detected and the previously stored call is deleted and logged. The second cancellation call does nothing because there is no more pending call for that customer.

Case-2: 4 C.O. lines for calls, DTMF string for cancels (Whozz Calling deluxe only)

```
01 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Samson J.F. | call in (Line 01)
02 I S 0000 G A0 09/26 11:30 AM 555-123-4567 Anybody Paul | call in (Line 02)
03 I S 0000 G A0 09/26 11:37 AM 555-222-3333 Doe John | call in (Line 03)
01 I S 0090 G A0 09/26 11:39 AM 770-263-7111 *0 | cancel (Line 01)
```

Lets program cancels first and set the cancel attribute '*' on Call- Type field.
(remember: DTMF string detected on a line with a call pending, line 01 in this case)

```
Index: 00 |
Call Type: *CANCEL | * means its a cancel, word CANCEL is useless
Line Number: ALL | call/cancels accepted on any line
Cust Number: * | Match on any value on that field
Cust Name: | field not used
DTMF/Hook: ON-HOOK | hang-up string sent to dialer to end call
```

NOTE: Actually the Whozz Calling module can detect DTMF tones on inbound calls only for 60 seconds to limit the monitoring time in order to reduce talk-off. Talk-off is a situation in which human speech (live or recorded) triggers the DTMF detector to capture false DTMF digits. But the module will report when line returns on-hook. This will allow us to determine if the call is answered and terminated without having to detect DTMF. For increased security, the dialer can be programmed to call again if the hangup string is not received correctly.

If the cancellation is not found (because no DTMF string), search continue until a match occurs.

```
Index: 01 |
Call Type: EMERGENCY_CALL | Call type used on pagers, reporting
Line Number: ALL | call/cancels accepted on any line
Cust Number: * | Match on any value on that field
Cust Name: | field not used
DTMF/Hook: | field not used
```

Test file used:

```
01 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Samson J.F.
02 I S 0000 G A0 09/26 11:30 AM 555-123-4567 Anybody Paul
03 I S 0000 G A0 09/26 11:37 AM 555-222-3333 Doe John
01 I E 0090 G A0 09/26 11:39 AM 770-263-7111 Samson J.F.
02 I E 0100 G A0 09/26 11:30 AM 555-123-4567 Anybody Paul
03 I E 0220 G A0 09/26 11:37 AM 555-222-3333 Doe John
```

Three calls are made then returns on-hook at different times. For this application we use the customer phone number field to determine the origin of the call.

Case-3: VS-600 hunt group for call types, another group for cancellations

```
01 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Room 12>Bed Call | call
03 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Room 13>Pendant | call
02 I S 0000 G A0 09/26 11:28 AM 770-263-7111 Room 12>Cancel | cancel
```

Cancelled or new calls detection order in not important here.

```
Index: 00 |
Call Type: BED | Call type used on pagers, reporting
Line Number: ALL | any line
Cust Number: | not used (empty)
Cust Name: >Bed Call | string to match (only room will remain)
DTMF/Hook: | field not used
```

```
Index: 01 |
Call Type: PENDANT | Call type used on pagers, reporting
Line Number: ALL | any line
Cust Number: | not used (empty)
Cust Name: >Pendant | string to match (only room will remain)
DTMF/Hook: | field not used
```

```
Index: 02 |
Call Type: *CANCEL | * means its a cancel, word CANCEL is useless
Line Number: | any line (field is empty)
Cust Number: | not used (empty)
Cust Name: >Cancel | this field is used for the definition match
DTMF/Hook: | field not used
```

Test file used:

```
01 I S 0000 G A0 09/26 11:28 AM 12 Room 12>Bed Call
03 I S 0000 G A0 09/26 11:28 AM 13 Room 13>Pendant
02 I S 0000 G A0 09/26 11:28 AM 12 Room 12>Cancel
```

Case-4: VS-600 hunt groups for call types, same extension cancellation (Avaya demo)

```
01 I S 0000 G A0 09/26 11:28 AM      Room 12>Bed Call      | bed call
01 I S 0000 G A0 09/26 11:28 AM      Room 12>Pendant      | pendant call
01 I S 0000 G A0 09/26 11:28 AM      Room 12>Bath         | bathrom call
02 I S 0000 G A0 09/26 11:28 AM      Room 12               | cancel
```

Calls must be detected first, cancellations last because there is no call type information.

```
      Index: 00                |
      Call Type: BED           | Call type used on pagers, reporting
      Line Number: ALL         | any line (field is empty)
      Cust Number:             | not used (empty)
      Cust Name: >Bed Call     | string to match (Room 12 will remain)
      DTMF/Hook:              | field not used
```

```
      Index: 01                |
      Call Type: PENDANT       | Call type used on pagers, reporting
      Line Number: ALL         | any line
      Cust Number:             | not used (empty)
      Cust Name: >Pendant      | string to match (Room 12 will remain)
      DTMF/Hook:              | field not used
```

```
      Index: 02                |
      Call Type: BATHROOM      | Call type used on pagers, reporting
      Line Number: ALL         | any line
      Cust Number:             | not used (empty)
      Cust Name: >Bath        | string to match (Room 12 will remain)
      DTMF/Hook:              | field not used
```

```
      Index: 03                |
      Call Type: *CANCEL       | * means its a cancel, word CANCEL is useless
      Line Number: ALL         | any line
      Cust Number:             | not used (empty)
      Cust Name: *             | Match on any value here (Room 12 in that case)
      DTMF/Hook:              | field not used
```

Test file used:

```
01 I S 0000 G A0 09/26 11:28 AM 12      Room 12>Bed Call
03 I S 0000 G A0 09/26 11:28 AM 13      Room 13>Pendant
02 I S 0000 G A0 09/26 11:28 AM 12      Room 12
```

Case-5: 3 lines for bed calls, 3 for bathroom and 2 for cancels

```
01 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | bed call
02 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | bed call
03 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | bed call
04 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | bathroom
05 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | bathroom
06 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | bathroom
07 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | cancel
08 I S 0000 G A0 09/26 11:28 AM 12          Room 12          | cancel
```

Lets program calls first. Lines 01, 02, 03 will be used for bed calls, lines 04, 05, 06 for bathrooms.

```
Index: 00          |
Call Type: BED     | BED CALL (or pendant to save lines)
Line Number: 01-03 | call must be between lines 01 and 03
Cust Number:      |
Cust Name: *      | Match on any value on that field
DTMF/Hook:        | field not used
```

```
Index: 01          |
Call Type: BATHROOM | BATHROOM CALL
Line Number: 04-06 | call must be between lines 04 and 06
Cust Number:      |
Cust Name: *      | Match on any value on that field
DTMF/Hook:        | field not used
```

Then, lines 07 and 08 will be used for cancellations

```
Index: 02          |
Call Type: *CANCEL | * means its a cancel, word CANCEL is useless
Line Number: 07-08 | on lines 07 and 08 only
Cust Number:      | not used (empty)
Cust Name: *      | Match on any value here (Room 12 in that case)
DTMF/Hook:        | field not used
```

Test file used:

```
03 I S 0000 G A0 09/26 11:28 AM 12          Room 12
06 I S 0000 G A0 09/26 11:28 AM 12          Room 12
07 I S 0000 G A0 09/26 11:28 AM 12          Room 12
```