Configuring custom Wiegand formats

Many access control systems use a Wiegand format for their user cards with as many as 50 bits of information stored. We need to filter out the parts of this data string that are useful to Net2 and check the unique user and security (site code) information.

Net2 has two fixed formats for 26 bit cards (the most common). One allows the site code to be specified when selecting the reader on the Doors screen. (See: AN1125 - Configuring Wiegand 26 bits with a site code < http://paxton.info/1753 >) and the other is a more basic 'Wiegand 26 bit' setting, but this combines the site and user code data into a single hybrid 8 digit number for Net2 to use. This makes it easy to set up but the 8 digit number has no relationship to any number printed on the card and may be unacceptableif they want to enrol a user by card number.

We can configure the Net2 software to accept Wiegand formats by setting up filters. These use rules that are defined in the Net2 Server Configuration Utility.

It is important to know the exact format of the data on these cards before attempting to set up the configuration rules. If these are incorrect, the reader may simply ignore the cards and will make the setting up of rules a very time consuming business.

Wiegand readers with Net2

A Wiegand card will contain a site code (sometimes called the facilities code) and a user code. Many will also have additional facilities codes, distributor codes, date codes, etc.

Consider the following 37 bit Wiegand format example: (spaces added for clarity)

P SSSSSSSS FFFF AAAAAAAAAAAAAAAAAAAAAAA P

P is a parity bit. These confirm to the reader that the data is complete - one at each end

S is the site code	8 bits
F is a facility code	4 bits
A is the card number	23 bits

For a card with the following information:

Site code	193	(11000001 in binary)
Facility code	12	(1100 in binary)
User number	1234	(00000000000010011010010 in binary)

The card data is:

```
1 11000001 1100 00000000000010011010010 0
```

We can set up a rule that will check the 193 site code and pass the Card number (1234) to Net2.

```
X 11000001 XXXX AAAAAAAAAAAAAAAAAAAAAA X
```

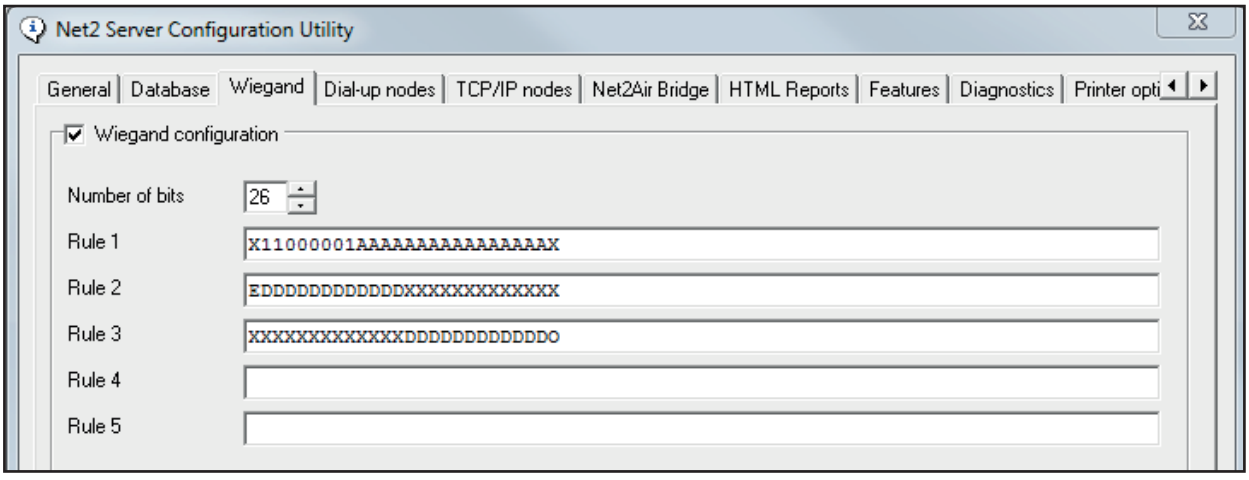
X - is ignored by the reader.
0 - must be zero (checked by the ACU but not passed to Net2)
1 - must be one (checked by the ACU but not passed to Net2)
A - is the card number (Passed to Net2)

This rule is entered into the Wiegand filter as follows:

Rule 1: `X11000001XXXXAAAAAAAAAAAAAAAAAAAAAA`

This represents the best option for Net2. (site code and card number)

NOTE: It is good practise to include the site code check on Wiegand cards as they are often supplied to customers in a batch starting at number 001. This gives the possibility that any 37bit card numbered 001 will work on this system.



Parity checks - why you should use them

Parity bits are used to ensure that the data received by Net2 is accurate and complete. If the ACU just accepts the data without additional checks, a bad read with a corrupted data bit will ask Net2 to act on a wrong number.

At best, this could just deny access to a valid user but is could also grant access using another users token number and their user name. As the Event log may be used for timekeeping or roll call, every effort should be made to ensure that this data is correct.

A parity check will look for bad data by running a calculation on the number of bits received. A bit that has been dropped or added will be detected.

The most common format for a 26 bit card is shown below as an example. (spaces added for clarity)

Even Parity Rule: `E DDDDDDDDDDDD XXXXXXXXXXXX X`

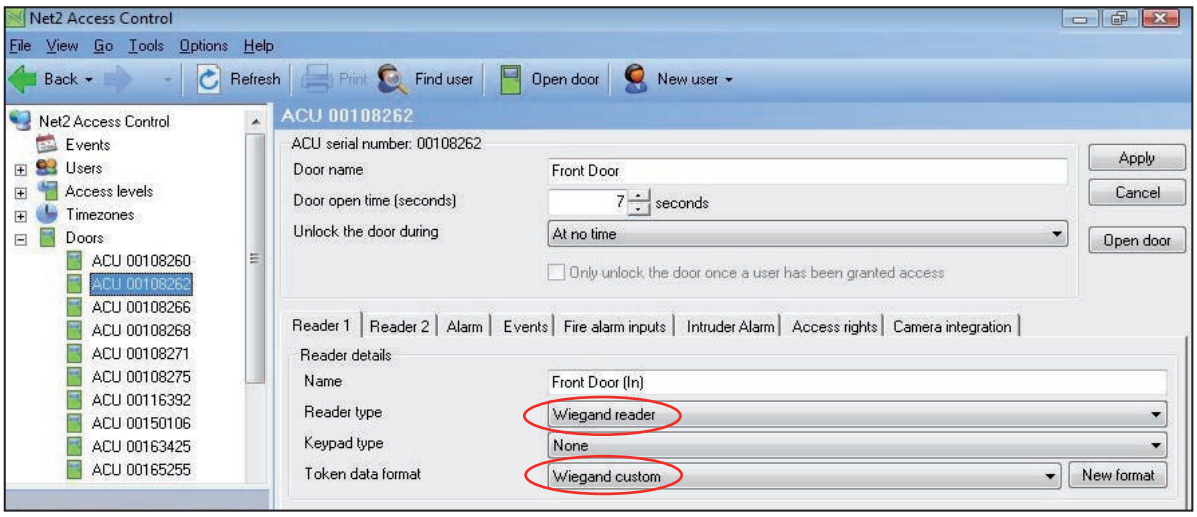
Take the first 13 bits (1 parity bit and 12 data bits), add them up and they must give an Even number. This is denoted by using 'E' as the parity bit and a 'D' for every bit to be considered in the calculation. (If there are 9 D bits then an E bit is included to make the total even)

Odd Parity Rule: `X XXXXXXXXXXXX DDDDDDDDDDDD O`

The last 13 bits are treated in the same way (12 data bits and 1 parity) but these must add up to an odd number denoted by an 'O' in the filter. Put an 'X' in the other locations to give a 26 bit rule.

The parity rules do not form part of the main filter rule 1 but are usually added as rules 2 and 3. Once the first rule has been determined, these extra rules should be added for completeness.

Once a custom Wiegand format has been entered into the Net2 Server Configuration Utility, a new format called 'Wiegand Custom' will appear in the 'Token data format' option. As below:



The filter information is stored in each ACU controller. Any changes to the Wiegand filter will require the Net2 server to be restarted so that it can update the system.

Net2 systems after v4.06 should restart the server automatically.

How to discover Wiegand card information

In most cases, the manufacturer of the card will be able to advise how many bits of data are encoded on the card (26, 34, 37, etc). They will not reveal the data layout or any site code information.

To read the card, run the Net2 software and set the Wiegand door reader to a 'Desktop reader'.

26 bit tokens

If it is a 26 bit card, set the format to 'Wiegand 26 bit' When you present the token, the reader should beep and the 'Add user' screen should display with a Paxton 8 digit number in the bottom right corner. Note this is a 24 bit number as the Paxton 26 bit converter ignores the two Parity bits located at each end. This number needs to be converted into Binary.
(Microsoft Windows has a Calculator with a Scientific view)

For example, if the number displayed was 13175734, enter this into the calculator and then select the Binary display (Bin). This converts the decimal number into Binary. (110010010000101110110110).

The layout is normally Site Code (8 bits) and Card Number (16 bits)

Therefore 13175734 (110010010000101110110110) gives:

11001001 (First 8 bits - Site Code = 201)
0000101110110110 (Last 16 bits - Card Number = 02998)

A Rule 1 filter set to 26 bits: X11001001AAAAAAAAAAAAAAAX will check the site code (201) and then pass the card number (02998) to Net2.

More than 26 bits

If we have a card has more than 26 bits we need two operations to discover all the data bits.
For example, for a 32 bit card:

Firstly, set a Wiegand filter with 24 A's at the start, and then X's to the end.

AAAAAAAAAAAAAAAAAAAAAAAAXXXXXXXX.

Apply this change and present the card to the reader. The 'Add user' software will display an 8 digit number. Copy this number into a text document for reference.

Now change the filter rule so the last 24 digits are A's and the start is 8 X's

XXXXXXXXAAAAAAAAAAAAAAAAAAAAAAAA.

Repeat the process of reading the number and then convert the two numbers into binary.

If the two number read and converted were:

13067029 `11000111011000110010101`
6493574 `01100011000101011000110`

By placing them one above the other, we can see where these overlap.

```
11000111011000110010101
       01100011000101011000110
```

We can then deduce that the 32 bits on this card are:

`11000111011000110001010110000110`

If the card we were using also has a number printed on it, for example 3246787, we can use this to confirm the position of the card number within the 32 bit data string.

32 bit card: `11000111011000110001010110000110`
3246787: `01100011000101011000011`
Filter: `X1000111AAAAAAAAAAAAAAAAAAAAAAAX`

The above filter should be fine for this card set. To confirm this, we need to repeat the whole process with at least two more cards. The site code will always be the common bits to the left of any card number. Note that any 0's will be missing from the left end of any card number when displayed on the PC screen.

We can see that the site code appears to be 1000111 (or 71) but without the exact definition, the actual position where the site code ends and the card number begins can only be assumed.

It is possible to set the filter to allow more than one site code; for example 70 and 71)
To do this, check for the common bits in both codes and then use X for the bits that differ.

Code 70: `1000110`
Code 71: `1000111`
Filter: `100011X`