Understanding custom card formats in Continuum

General overview:

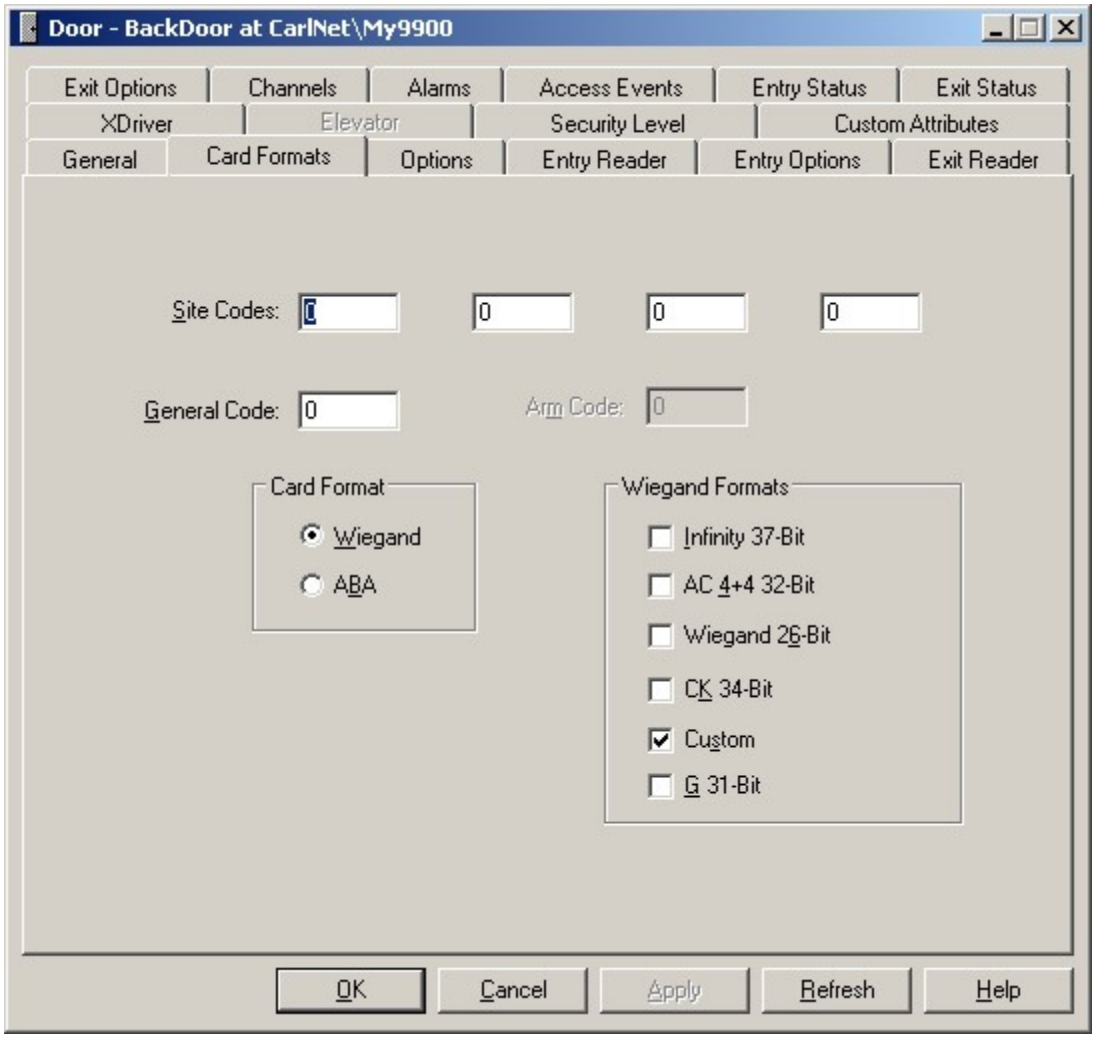
This document will step through the process of deciphering custom cards bit formats in order to set up custom cards access through Continuum. The step by step examples may vary depending on your custom cards, but the general knowledge presented will allow you to setup your system to use custom cards.

You'll need at least three cards with card numbers in succession. Any information that you have about the cards, bit patterns or site code information is helpful. It is very difficult to decipher the site code without knowing it. Most cards have the card number on them, so figuring out the bit pattern for the card number can be easily done. But you'll need to know the lowest card number and highest card number so that you make sure you are using the correct bit count to allow the card with the highest number to be able to access the system.

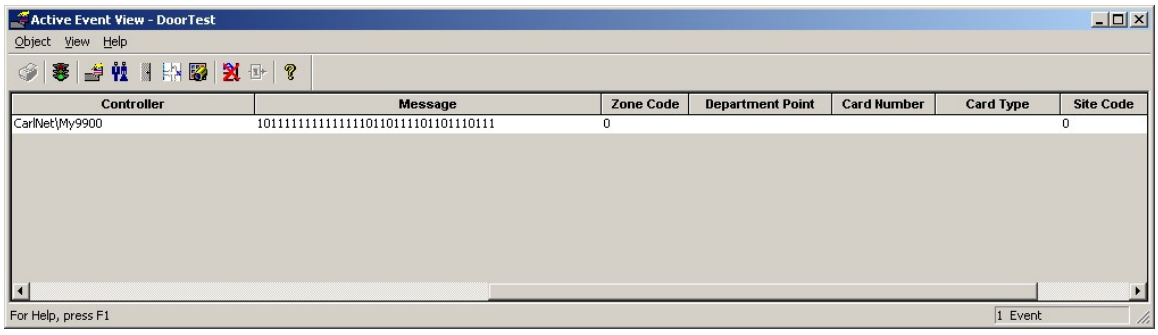
The cards that we will be using in this example are HID 35 bit custom cards. Your cards could vary and yours may be 36, 37 or 26 bit. These could vary, but after going through the examples in this document, you'll get the general idea of how to read these cards and set up the Continuum system for custom card access.

For the following examples, we will test 3 HID 35 bit custom cards. I had the benefit of being told that the site code was 4095. The card numbers were 900539, 900540 and 900541. These were the numbers printed on the cards.

The first step to figuring out custom card bit patterns is to have a door configured and an active event view setup to show the events of the door to test cards on. Once you have the door setup, select the custom checkbox and swipe the card.

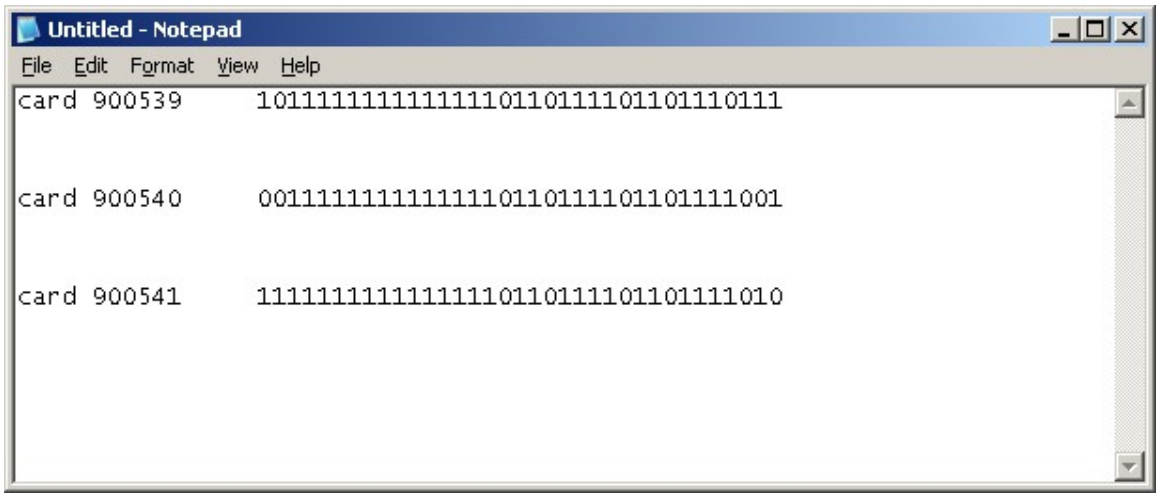


You should see similar results as the image below.

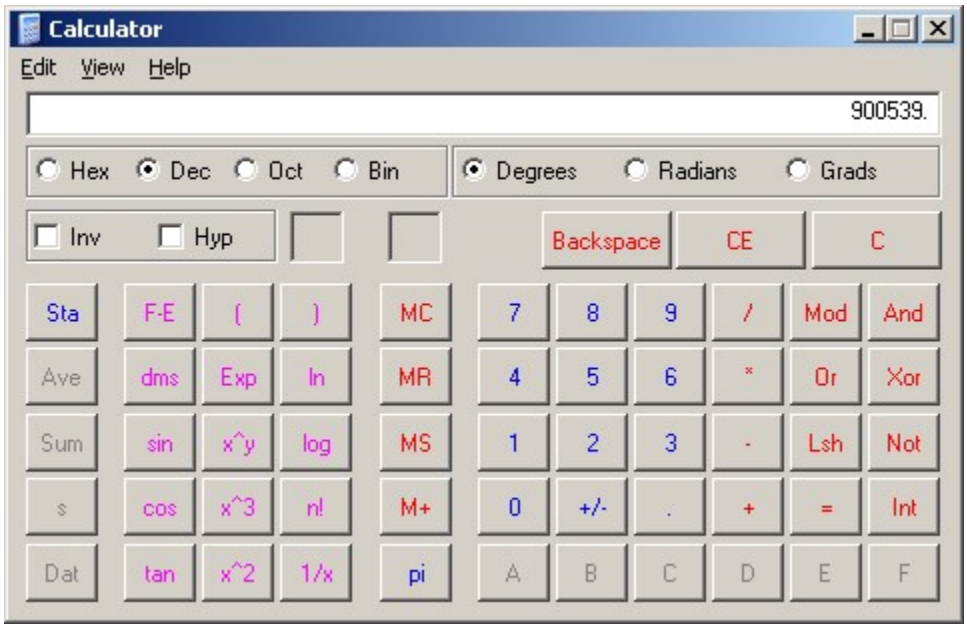


The above card was read by the reader and in the message field we see the bit pattern for that card. The card I swiped was 900539 and its bit pattern is 10111111111111110110111101101110111. To simplify deciphering the bit

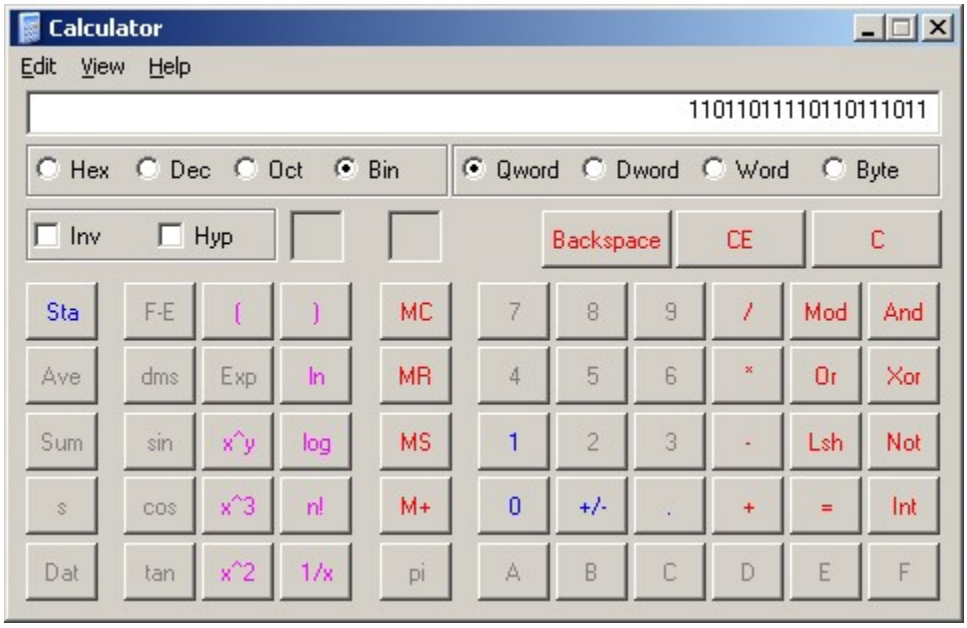
pattern, I use notepad hold the information. In the image below, I have read all three cards and copied the bit patterns accordingly.



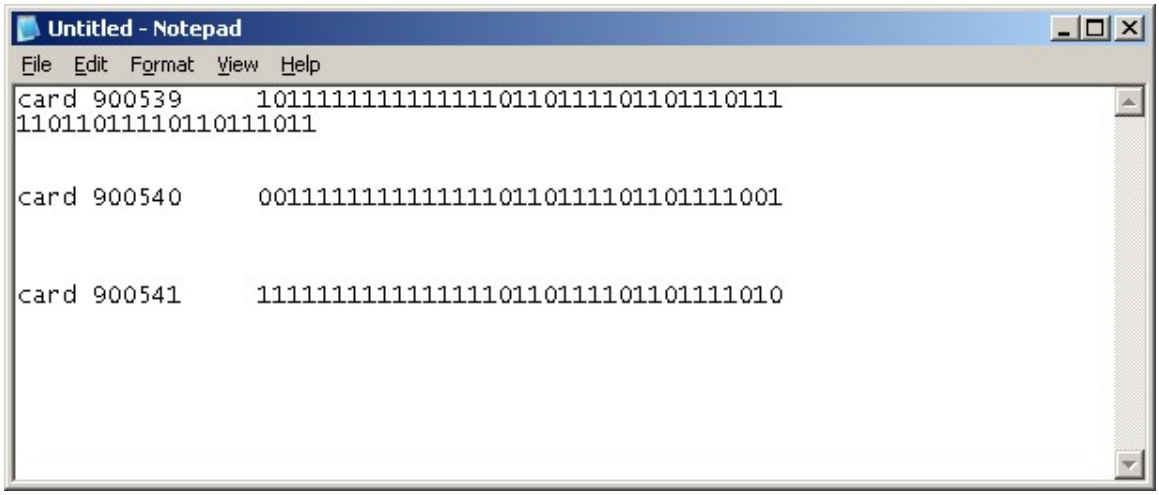
Now I will open up the calculator in windows and enter the card number in decimal.



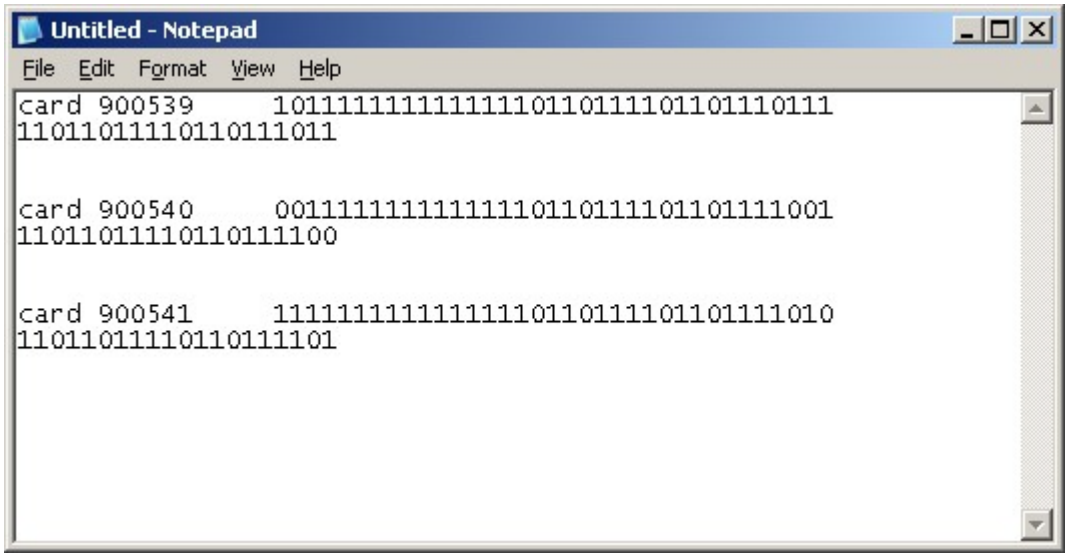
Then select the Bin (binary) radio button and this will translate from decimal to binary.



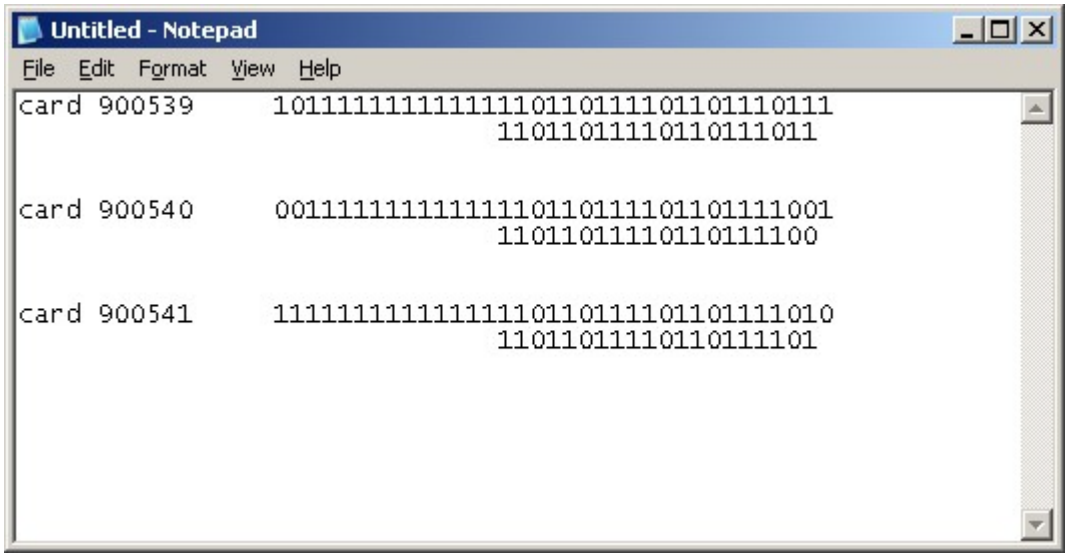
At this point you can either 'control c' or edit copy the binary number and copy it the notepad that we created earlier.



Perform the same operation by transferring the card number to binary and copy these to your notepad. You should have something similar to the image below.



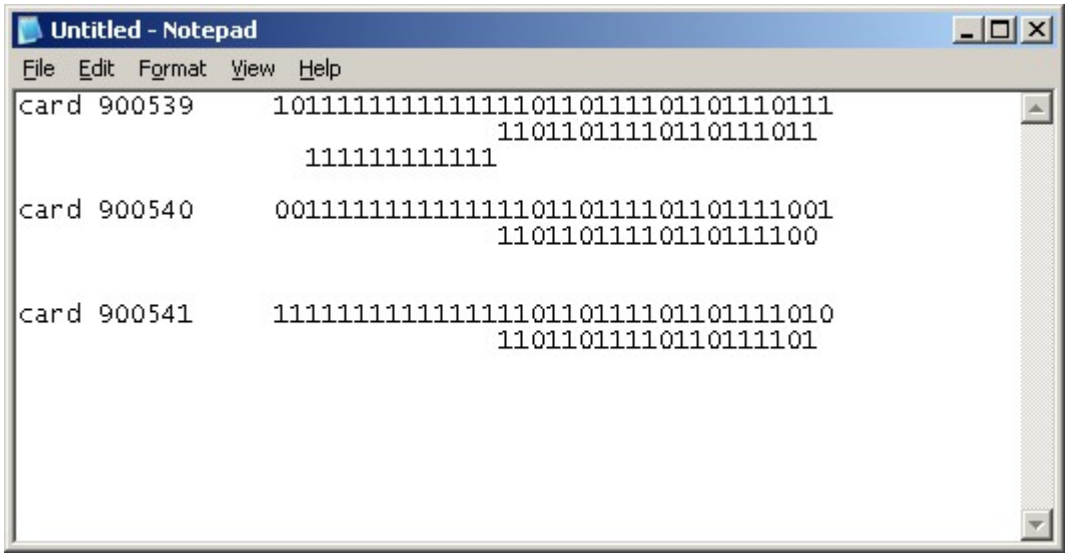
Now we will align the card number bit pattern with the total bit pattern that we received from the doors card reader in the Active Event View. In the image below we have aligned the bit patterns.



Now we can see what the start bit of the card number is and the end bit of the card is. In the example above we can see that this is a 35 bit card. Since this is binary, we'll count from 0 to get the start/end bit of the card. Our card number starts at 14th bit and ends at 33rd bit.

Now that we have deciphered the card number, we'll move onto the site code. Not all cards will have a site code, but in our example which is probably not the best, we'll show you how to get the site codes start and end bits. The reason I say that this example is not the best is because of our site code

numbers binary number. As mentioned before our site code number was 4095 and in binary is 111111111111. Now you see the dilemma of trying to align our site code binary number with the total binary number. Had our binary site code number been something other than all 1's, it would have been a better example. But I think we'll get the general idea here. I will use a separate line in our notepad to align the site code.



Since we were lucky enough to have the site code number it was easy. Our site code start bit is 2 and the site code end bit is 13. But you can see that not having the site code number, we would have no idea of which bits and how many to read from. So the more information you have about the cards, the better off you'll be.

Now that we have deciphered the site code and card number from the overall binary number from the card, we can now setup our custom card accessformat numeric that is required to allow Continuum to validate this custom format.

Here is the information on creating a custom format in the Andover System. A custom format must be used when the card format is not one of the standard pre-defined formats. Only **one** custom format can be used in an ACX controller or NetController.

Setting Up a Custom Wiegand Format

1. In each ACX700 and ACX780/781 and NetController, create an InfinityNumeric point called AccessFormat (use this spelling exactly!). During the creation process, attach a manual array (log) with 19 entries to the point.

2. Create the following InfinityProgram in each ACX700/780/781 and NetController. The program should be type FallThru, and AutoStart. This program basically saves you from having to set each array value individually for each controller. The program, once created, can be copied to all other access controllers that need it.

Program Name: SetArray
FlowType: FallThru
AutoStart: True
Code:

```
Set AccessFormat[1] to 35      'overall number of bits
Set AccessFormat[2] to 0       'control mask
Set AccessFormat[3] to 0       ' manufacturer's code
Set AccessFormat[4] to 0       ' start bit of man. Code (first bit is 0, not 1)
Set AccessFormat[5] to 0       ' end bit of man. code
Set AccessFormat[6] to 2       ' start bit of site code
Set AccessFormat[7] to 13      ' end bit of site code
Set AccessFormat[8] to 14      ' start bit of card number
Set AccessFormat[9] to 33      ' end bit of card number
Set AccessFormat[10] to 0      ' odd parity mask, first 16 bits
Set AccessFormat[11] to 0      ' odd parity, next 16
Set AccessFormat[12] to 0      ' odd parity, next 16
Set AccessFormat[13] to 0      ' odd parity, next 16
Set AccessFormat[14] to 0      ' even parity, first 16
Set AccessFormat[15] to 0      ' even parity, next 16
Set AccessFormat[16] to 0      ' even parity, next 16
Set AccessFormat[17] to 0      ' even parity, next 16
Set AccessFormat[18] to 0      ' card number multiplier
Set AccessFormat[19] to 0      ' card number divisor
```

3. In each Door Configuration, select the Custom Wiegand card type. Other card types may also be selected.
4. For each Personnel, select the Custom Wiegand card type, and enter the correct card number and site code.

If the card is not working, check the Event Log and determine the cause. For Invalid Card Parity errors, set all the parity masks to 0 and try again. For proximity cards, the parity check may be eliminated with little consequence.

Once configuration is set up correctly, you'll then be able to assign the custom card to personnel and begin using the custom cards. Below is the results of our custom card that we deciphered earlier being allow access.