

Will Harper's TCP/IP subnetting and troubleshooting Cheat Sheet (Last update 04/2009)

EVERY answer to EVERY IP subnetting problem is in this window

But you gotta TRUST it!
Will Harper's

TCP/IP CHEAT SHEET

Bit Mask	Mask	Blocks	Nets	Hosts
0000 0000	0	256	1	254
1000 0000	128*	128	2*	126
1100 0000	192	64	2	62
1110 0000	224	32	6	30
1111 0000	240	16	14	14
1111 1000	248	8	30	6
1111 1100	252	4	62	2
1111 1110	254	2	126	X
1111 1111	255	1	254	X

(Notes)

* 128 mask valid for Microsoft and CIDR ONLY

X valid in Class B, CIDR or SuperNetting only

Subnet mask can be expressed in several ways:

Binary (The way computers see it) 1111111111111111111111111111110000

(Rule: There can NEVER be a zero embedded in a mask. The first 0 is the end)

Dotted Decimal 255.255.255.240

Hexidecimal: ff.ff.ff.f0

Classless (CIDR): /28

CIDR *usually* also requires Variable Length Subnet Mask (VLSM) capable hardware

RIP V1 can only be used for default classful routing (NT 4 only offers RIP V1)

RAS uses only the default classful NetMask (Class A=255.0.0.0, B=255.255.0.0, etc)

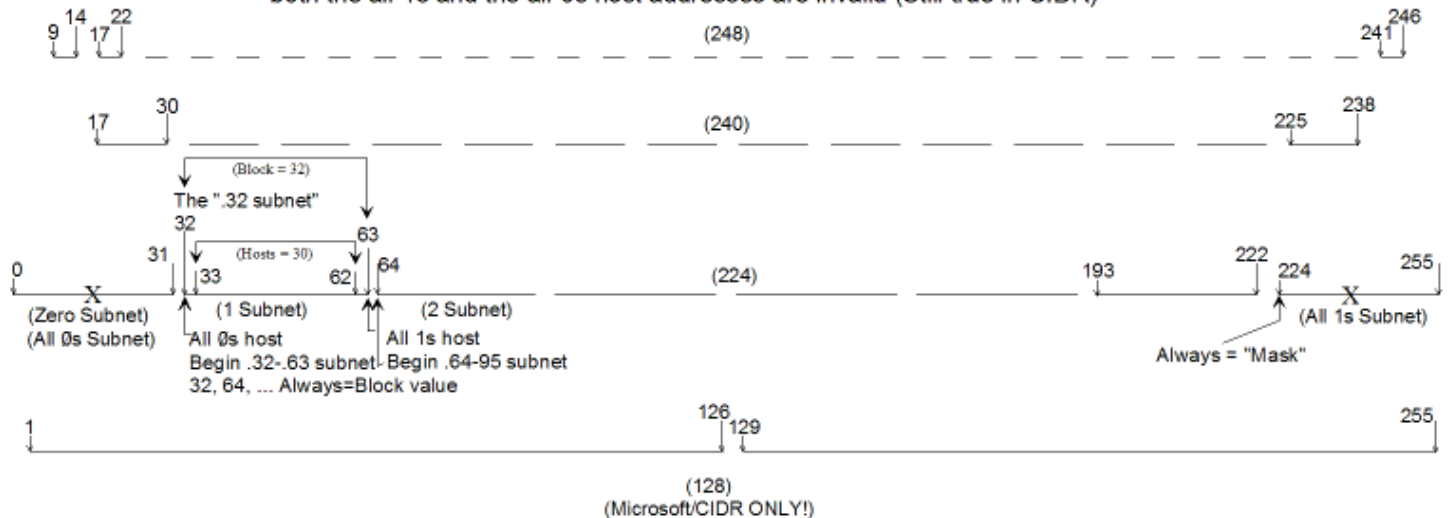
The **Subnet** address is always **hidden** behind the 1s in the mask

The **Host** address is always **hidden** behind the 0s in the mask

(except in special cases)

both the all-1s and the all-0s subnet addresses are invalid (NOT true in CIDR)

both the all-1s and the all-0s host addresses are invalid (Still true in CIDR)



This tool is designed to remove the "memorization" from IPV4 subnetting and troubleshooting. It works for both Classful (the one on the tests) and Classless (CIDR) subnetting - the one you will have to use in the real world. To use the cheat-sheet for CIDR, remember to "put back" the two additional subnet addresses that are the exceptions for CIDR. Do this mentally, so the table continues to work for all.

This is an ALGORITHM, which means that "if you are thinking - you are doing it wrong." The only hard part is remembering what the names of the columns are!

Start in the upper left corner, and just go down, reducing the zeros (0) in the bits, making them 1s. Then, in a yo-yo fashion, if you went down, then go up for the next column. The 2nd column requires a little bit of mental (3rd-grade) arithmetic. Start with 255, then "take away" 1 for the 1s bit that you dropped in then next row, then "take away" 2 for the 2s bit, then "take away" On the Blocks column, you will NOT follow the yo-yo rhythm; because you start at the bottom a second time. Just slide the 1s bit that you dropped in the 1111 1110 row down-and-to-the-right; slide the 2s bit from 1111 1100 down-and-to-the-right; repeat. For the Nets column just COPY the numbers in the Blocks column - since there are rules that apply to the number of legal networks (take away first and last for Classful), you will do a mental "take away" as you copy. Continue the yo-yo in the hosts column, COPYING the numbers from the Nets column to the hosts column. Again, rules apply. The 1111 1111 mask (255) CANNOT have any host addresses "hiding behind the zeros in the mask;" and the 1111 1110 mask can have exactly two (0 and 1) host IDs - but the rules say that "the all-zeros host ID is INVALID" - with no exceptions for CIDR; and "the all-1s host ID is INVALID" - again with no exceptions for CIDR.

The "joints" at the bottom illustrate how to manually lay out (and visualize) the subnets' IP addresses, given a particular mask. The length of each joint is equal to the size obtained from the "Blocks" column - the STARTING IP of each block is equal to zero (0) plus increments of the size in the Blocks column (remember, it's binary). This is VERY HELPFUL for troubleshooting - map the problem IP addresses onto the joints, and you can easily see which IP addresses are not on the same subnet - and those that break the "All-zeros-host-address" or the "All-ones-host-address" rules.

Remember that adding one-bit to any binary number doubles the value of the number - I'll leave it as an exercise for you to see how this same table works equally well for 3rd Octet or 2nd Octet masks. (Hint: from bottom-to-top, the values are: X, 510, 1022, 2046, 4094, 8K-2, 16K-2, 32K-2, 64k-2)

To find the "Subnet IP address," set all of the host bits, for each subnet, in the IP ADDRESS to 0's, then convert back to a people-readable IP address.

This can be done with the cheat-sheet; or with a scientific calculator for very large numbers of networks (Remember: if you are CALCULATING the IP bits for, for instance, the 768th subnet - you have to enter one less (767) in the calculator - It's binary)

To find the "Subnet Broadcast Address" set all of the host bits in the subnet's IP ADDRESS to 1's, then convert to a people-readable IP address