How to create an item (6/9/01)
Updated by Exxy (8/14/01)

You need to create a new item so first type:

Create_object

Once you've created the object you need to change the name of it to what you want it to be.

Change_object_name <current name of item> <Article> <descriptor> <DESIRED NAME>
Example: change_object_name item a leather sheath

Article: the definite article is the, the indefinite article is a or an, and their force is generally to impart specificity to the noun or to single out the referent from the class named by the noun. (**4**-character limit)
Descriptor: The descriptor is the primary adjective used to identify the item. (**16**-character limit)
Name: The name has a limit of **20** characters.

After you've done that, you'll still see just an item in your hand, now you need to change the description of it.

Change_object_shortdesc <object> <what you want it to look like>
Example: change_object_shortdesc sheath a blackened leather sheath
The short description is limited to **120** characters.

Then you can put a full description on it, which you and others will see when you type LOOK at <object>. You can also put a READ description on the item.

For the full description: **change_object_fulldesc** <object> <and what you want, there is a limit to the length of this description though>

The full description is limited to **320** characters.

For the read description: **change_object_read** <object> <what you want it to say when you READ the object>

The read description is limited to **400** characters.

Now you need to add the values of the item, what it is, what it can do, hold, protect, etc.

To see the object's current values type:

Show_object_data <object>

You will see :

Object data for a blackened leather sheath:
Object ID: 1854
Ar/Ds/Nm: 'a' 'leather' 'sheath'
Script: 0
There are no items inside of it.
Data1: 0 0 0 0 0 0 0 0
Data2: 0 0 0 0 0 0 0 0
LOOK: ''
READ: ''

The values mean the following:

For the Data1 row

Value 1: Is the item altered? 0 equals no. 1 equals yes.

Value 2: How many charges does the item have? This could also be where the amount of mana stored in an item is calculated. If neither apply then just put 0.

Value 3: This one is a bitvector. In order to calculate this number, you will need to open the calculator accessory on your computer, go to view, choose scientific, then select binary (BIN) mode. After you've entered the flag values in binary mode, you'll need to switch to decimal (DEC) to calculate the number you need to enter here. This particular bitvector has 5 flags on it. A bitvector is a binary number (good old 1's and 0's). Each digit represents whether a flag is on or off... true or false. For example, 1001 in binary would mean that the flag 4 is on, flag 3 is off, flag 2 is off, and flag 1 is on. Binary is always read with the highest

digit to the left. Just like 439 in decimal, the hundreds place is the highest digit... and it's on the left.

They are:

Flag 1: Is the container closeable?
Flag 2: Is the container pickproof?
Flag 3: Is the container currently closed?
Flag 4: Is the container currently locked?
Flag 5: Are the contents in the container able to be stolen?

So... if it's a closeable container that is pickproof, open, not locked, and able to be stolen out of, it would be:

10011

After you put this number into the calculator under binary and then switch it to decimal format, you should get the number 19.

Value 4: Extra Flags. The important flags are listed below. Unmentioned flags have no meaning at this time.

Flag 1: Is the item a container?
Flag 2: Is the player blocked from selling this item?
Flag 4: Is this object invisible to normal players?
Flag 8: Is this object usable even if it's invisible?
Flag 9: Is this object blocked from being janitorized?
Flag 10: Is this object janitorizable only after it has been picked up at least once?
Flag 11: Do you want this object janitorized regardless of any other flags?
Flag 18: Can only staff members pick this item up?

If this item was a container, but not "staff-only" or "can't sell it", then our bitvector looks like this:

0000000000000001

Note again, flag 1 appears on the far right."

Value 5: Type of item. There are 16 types, numbered 0-15. This is not a bitvector. The item is only one of the following:

0 - item is a source of light
1 - item is a weapon
2 - item is ammunition of some type
3 - item is armor of some type
4 - item is money
5 - item is treasure (usually gems)
6 - item is a potion
7 - item is a miscellaneous worn item
8 - item is a trap (currently unimplemented)
9 - item is a container for fluids only
10 - item is a key
11 - item is food
12 - item is drink
13 - item is a boat (currently unimplemented)
14 - item is some miscellaneous object
15 - item is a shield (basically armor that only works if it is NOT worn and in the left hand)

Value 6: This is the item's value in gold pieces.

Value 7: This is another bitvector. This bitvector represents what WEAR slots the object takes up. For example, a shirt may only take up your chest, while a suit of armor may take up your chest, arms, legs, etc.: one slot of each. Note that garments/clothing never takes up a back slot unless it distinctly is worn ONLY on the back. This bitvector also determines whether or not a normal player is able to pick this object up.

Flag 1: can the item be picked up?
Flag 2: does the item use up a head slot?
Flag 3: does the item use up an eye slot?
Flag 4: does the item use up a neck slot?
Flag 5: does the item use up a back slot?

Flag 6: does the item use up a chest slot?
Flag 7 does the item use up an arm slot?
Flag 8: does the item use up a hand slot?
Flag 9: does the item use up a finger slot?
Flag 10: does the item use up a waist slot?
Flag 11: does the item use up a belt slot?
Flag 12: does the item use up a leg slot?
Flag 13: does the item use up a foot slot?
Flag 14: is the item just worn?
Flag 15: does the item use up an ear slot?
Flag 16: does the item use up a shoulder slot?
Flag 17: does the item use up a wrist slot?
Flag 18: does the item use up a thigh slot?
Flag 19: does the item use up an ankle slot?

So...if you had armor that was worn/covered your legs, chest, arms, the number would be:

00000001010001100001

There are:

2 head slots
2 eye slots
3 neck slots
2 back slots
1 chest slot
2 arm slots
1 hands slot
8 finger slots
3 belt slots
4 waist slots
2 leg slots
1 feet slot
25 anywhere slots
1 ear slot
3 shoulder slots
1 wrist slot
1 thigh slot
1 ankle slot

Value 8: How much the item weighs in pounds. If the object is type 5 (gems and precious metals), then this is the number of tenth-pounds. For example, if this was a 4½-pound emerald, then the weight would be 45. If it was just a ½-pound emerald, then the weight would be 5. Again, this only applies to object type 5.

Please remember to calculate all the bitvectors into decimal mode.

The second row of data calculates the items armor class/protection and the damage done by a weapon as well as how much a container will hold and some other things.

The values for the second row are:

For containers:

Value 1: capacity in pounds

Value 2-8: meaningless

For armor/shields:

Value 1: adds to your armor rating

Value 5: armor type

0=light armor
1=medium armor
2=heavy armor
3=shield

For weapons:

Value 1: number of dice (for damage)

VALUE 2: sides on the dice (for damage)

VALUE 3: bonus to damage

(For example, if values 1, 2, and 3 are [1 6 2], then the weapon would do 1d6+2 damage.)

VALUE 4: weapon size

0=TINY
1=SMALL
2=MEDIUM

3=LARGE
4=TWO-HANDED

VALUE 5: weapon subtype

0=simple
1=exotic
2=martial
3=bow
4=crossbow

VALUE 6: damage type

0=bludgeoning
1=piercing
2=slashing

VALUE 7: critical damage

20 minus this number = minimum critical hit (roll of 1d20). For example, if this number is 1, then 20-1 = 19. They need a 19 or a 20 for an automatic critical hit.

To enter the data values for an item:

CHANGE_OBJECT_DATA1: <item> <value1> <value2> <value3> etc
CHANGE_OBJECT_DATA2: <item> <value1> <value2> <value3> etc