# DD Area Scribe Users Manual

## Table of Contents



# 1. Introduction

DD Area Scribe or Scribe is a tool for producing formatted area files for the Dragons Domain Envy-style MUD. DD area files retain the Envy MUD format, with changes to the `#AREA` header information and the addition of new variables and flag values to other sections.

Scribe is a script that reads a simplified text source file containing all area information, validates the information, and if no serious errors are found, outputs a formatted area file. Scribe is written in Perl and is run from the command line (see Section 2).

Scribe was written as an alternative to writing area files by hand or by using the existing Windows tool MZF. The advantages and disadvantages of each of these systems might be summarised:

*Writing areas by hand:*

- Laborious; lots of redundant variables to include. Unsafe; no error or format checking.

*Using the MZF tool:*

- Easy to use; convenient and simple user interface.
- Format and variable checking provides some safety.
- Buggy; contains known bugs and bizarre behaviours.
- Incompatible with DD; MZF does not contain DD's special flags and values; output requires hand-editing.
- Win16/32 executable only.

*Using the Scribe tool:*

- Fairly laborious; all data are handwritten, although in a friendly, human-readable format.
- Safe; variable checking and correct formatting.
- Compatible; contains all DD special variables and flags.
- Requires Perl 5 for compilation.

Scribe meets writing areas by hand and using MZF midway, providing full DD-server compatibility and error checking, but not including a simple to use graphical interface.

# 2. Usage

Producing an area file with Scribe involves the following steps: 1. Source file is written The area source file contains all of the relevant information to be included in the final, correctly formatted area file. It is a text file with a simple, human-readable format that is easy to edit. Information can be added to the source file in any order. 2. Scribe is used to check the source file and write the area Scribe is used to process the area source file. It reads through the source file checking for obvious formatting errors and then validates this information, filling in any missing variables where it can. If there are serious errors then Scribe will indicate where they occur in the source file and then exit. If there are no serious errors, then the area is formatted and written to file. Scribe is a Perl script that is run from the command: scribe.pl <source file> <destination file> For example, scribe.pl kahsis.src kahsis.are Area files traditionally have the .are extension. Both source and destination files must be indicated for the script to run. Scribe will create a backup copy of any existing destination file.

1. What's Perl? If you are not able to run Perl but would like to use this script to produce your areas, you can write your source files and send them to one of the DD Immortals, who will test the file and inform you of any errors.
2. Producing source files The area source file consists of discrete blocks of data that each define area information (area name, author, etc), one mob (mobile, creature), one room, one object, one mobile reset, one object reset, one shop or one help file entry. Some reset information (e.g. shutting doors and mob special functions) are included in the above blocks. Blocks may occur in any order within the source file. A block consists of a header indicating the type of data it defines, followed by a list of field and value pairs. Fields are short keywords representing room, object, mobile or area variables. A block is terminated by the start of another block or the end of the source file. Headers are single words preceded by a dollar sign $ and occupy one line. Field keywords and values are separated at least one space or tab (one tab is nice). Field/value pairs occupy one line unless they define multiple-line text blocks, in which case they are terminated by a line containing a single tilde ~. You should not terminate single line strings of text with a tilde: these will be automatically added by Scribe. Blocks have the following format: $header field1 value1 multiple_line_field <lines of text> ~ field2 value2 … fieldN valueN

For example, the following block defines the name, short name, long name, description, sex, affect flags, vnum, level and act flags variables for a mobile: $mob nm octopus sh an octopus lo An octopus crawls along the coral. de The octopus' slimy body is purple with bright red blotches. It crawls over the coral, its tentacles writhing horribly. ~ sx female aff hide sneak vn 1 lv 5 act scavenger wimpy stay_area

(Field/value pairs for each type of block are discussed in detail in Section 5 onwards.) Field/value pairs can be arranged in any order within the block. Multiple definition of the same field can be made, although only the last instance will be used. The dollar sign $ at the beginning of a line indicates the beginning of a new block, so should be avoided. Your source file will consist of a series of blocks of any type in any order. Use your favourite text editor to produce the source file. All text before the first block header is ignored, so you can add comments if you wish. Your source file must contain an $area block in order to be compiled without error. Source file format: Any number of comments at the start of the file. $area <area header information> $… <mobile, room, object, etc blocks as desired> You may add comments at any point in your area file by prefixing them with a

non-alphanumeric symbol that is not a dollar sign $, e.g. #, ; or whatever symbol you prefer. . This is a comment # So is this ; The line below is a header $mob . This is a comment within a $mob block

Note Make sure that you hard-wrap lines of text in multi-line text blocks to under 80 columns, i.e. supply a carriage-return at the end of every line. Make sure your text editor isn't soft-wrapping your text!

1. Area header and base vnum definition The area header block defines the #AREA section of the final area file. It describes the name and author of the area, and character access information. One definition is required, or your source file will produce errors. Note on Vnums Vnums, or virtual numbers, are the unique identification numbers for the mobiles, rooms and objects in your area. No blocks of the same type (mobs versus rooms versus objects) may share the same vnum; blocks of different types may share a particular vnum. Vnums are indicated as relative numbers in the area source file (usually beginning at zero). Scribe uses relative vnums: the vnums used in the source file are numbered from 0 upwards. A base vnum is defined in the area header block and is used to calculate absolute vnums from the relative values:

```
    Final vnum in area file = base vnum + relative vnum
If you have been allocated a range of vnums from the DD Immortals for use in your
area (e.g. 2600-2799), use the lowest vnum as your base (2600), and number your
mobs, rooms and objects from zero upwards.  If you haven't been given a range of
vnums, you should still number mobs, rooms and objects from zero up, and just use
any value for your base value.  The use of a base value allows you to easily
renumber your area if necessary, e.g. when you are finally given some to use by
those lazy Imms.
If you need to use vnums for rooms, mobiles or objects that are not defined in the
same source file (i.e. are defined in other areas), you need to use temporary
holding values and manually edit the formatted area file produced by Scribe.
```

Header $area Fields Field Description Type au author text ti title text ls lower level suggested number us upper level suggested number le lower level enforced number ue upper level enforced number bv base vnum number Description au Author Line of text The person or people responsible for writing the area. ti Title Line of text The name of the area. Don't make it too long (25 characters maximum). ls us Suggested level range Number: 0 or higher The suggested level range for travelling to your area, as shown in the online AREAS command. Use these values to indicate what level of character would profit from visiting your area; don't use them to indicate the lowest and highest level mobs (1, 100 is fairly unhelpful). le ue Enforced level range Number: 0 or higher The level range for permitted entry to your area. Characters outside this range may not access the area and get the 'God prevents you from entering there' message when they try. Note: only the lower level limit is currently enforced. bv Base vnum Number: 0 or higher The base value used to calculate absolute vnums from relative vnums. Note that if your area links to other areas in the MUD and you want to be able to have these links active while you build, it may be best to set this to 0 and use absolute vnums in your .src file, for convenience's sake. Example

$area ti The Planet Vulcan au Mr Spock ls 75 us 90 le 0 ue 100 bv 2600

1. Recall header The recall header block defines the #RECALL section of the final area file. It provides the author with an opportunity to override the default recall room location for a

player who is adventuring in the area. Header $recall Fields Field Description Type rl recall location number

Description rl Recall location Number: vnum of a room in the MUD The vnum of a room a character in this area should recall to by default. Does not have to be in the current area. If the character has multiple recalls available and is using a non-default one, they will recall there, and not to the #RECALL location. Example

$recall rl 27347

1. Area special header The area special block defines the #AREA_SPECIAL section of the area file. This section can be used to add supported area-wide features.

Header $special

Fields Field Description Type af area flags keyword list xp experience modifier number Description af area flags Line of text Flags that produce area-wide effects. They include: school area is a MUD SCHOOL (new character beginning area) no_quest no mobs in this area may be auto-quest targets hidden the “areas” command will not show this area, and DD’s mapmaker will not generate maps for it safe pkilling cannot happen in this area no_teleport a player cannot teleport into this area no_magic magic cannot be used in this area

```
E.g.    af  hidden no_teleport safe
```

xp experience modifier Number: 0 or higher an experience point modifier that will be applied to all mobs killed in this area. 100 == no change, 50 == halved, 200 == doubled, etc

E.g. xp 125

1. Mobiles Mobiles are the creatures that populate your area. They are defined in the #MOBILES section of the final area file. You don’t have to define any mobs in your area for it to be valid. Mobs are individually defined in single blocks. You can have as many mobile blocks as you wish. Mobiles may not share the same vnum: this will produce an error when you run Scribe over your source file. Remember to format your descriptive text fields to fit within an 80-column screen!

Block header $mob or $mobile Fields Field Description Type nm name (keywords) text sh short description text lo long description text de description multi-line text block vn vnum number lv level number al alignment number sx sex keyword bf body form keyword list act act flags keyword list aff affect flags keyword list sp special function keyword mp mob program multi-line text block te teacher skill text Description nm Name (keywords) Line of text The keywords that can be used to indicate the mob. E.g. nm wraith hazy shadow sh Short description Line of text The short name of the mob, used whenever an action is performed involving the mob. Don’t capitalise any leading ‘a’, ‘the’, ‘an’, etc: the DD server does automatically where necessary. E.g. sh an alligator sh the Gatekeeper sh Mycroft lo Long description Line of text The description of the mob as it appears in room after the LOOK command is issued. Capitalise the initial letter and don’t make the description too long if the mob will have many affect flag labels like (White Aura) and (Flaming). Remember full stops etc. E.g. lo A snake lurks in the grass. de Description Multiple-line text block The description of the mob as it appears after the LOOK <mobile> command is issued. Descriptions can

span multiple lines; the de text block is terminated by a line containing a single tilde ~. Text on the same line after the de keyword is ignored. It is best to justify your text hard up against the left margin. Leading space before the first character in the block is removed by the DD server. E.g. de The alligator is immense and ferocious, thrashing its tail in the water and baring its razor-like teeth. ~ vn Vnum Number: 0 or higher The virtual number of the mob: its unique identifier. Remember this is a relative value, and will most likely begin at 0 (see Section 5). lv Level Number: 0 or higher The level of the mob. al Alignment Number: ¬1000 to 1000 Default: 0 How good or evil the mob is. 1000 is absolutely evil, 0 is true neutral, 1000 is absolutely good. Defaults to zero if not indicated. sx Sex Keyword Default: neuter The sex of the mob. Defaults to neuter (sexless) if not present. Indicated by a single keyword: neuter male female E.g. sx female bf Body form Keyword list Default: none Describes the morphology (physical structure) of the mob. The default value of none describes a humanoid of normal size that is capable of speech and is made of flesh and blood. Body form controls carnage and corpse production upon the mobile's death, and affects what combat manoeuvres the mob may use or have used against it. Body form is set using a list of any of the following keywords: none default value no_head has no head no_eyes has no eyes no_arms has no arms no_legs has no legs no_heart has no heart no_speech cannot speak the common language no_corpse does not produce a corpse: body disappears upon death and loot falls to the ground huge enormous in size inorganic not made of flesh and blood has_tail has a tail E.g. bf no_arms no_speech bf no_heart inorganic no_speech huge act Act flags Keyword list Default: none Act flags define how the mobile behaves within the MUD world. If not included, a default value of none is used, describing a non-aggressive, non-wimpy creature that wanders between rooms and has no special interactions with players. The following flags may be used: none, zero no flags sentinel stays in one place scavenger picks up objects from ground questmaster can give players random quests aggressive aggressive: attacks players within sensible range stay_area does not leave the area wimpy flees from combat if hurt no_quest will not be selected as a target for a quest practice can train players regenerator heals much faster than normal no_charm cannot be charmed healer is a healing mob famous fame rewarded if killed lose_fame fame subtracted if killed wizinvis undetectable by players mount can be mounted tinker repair damaged items for money banker runs a bank for players identify identifies objects for players die_if_master_gone will die if its master is not in the same room clan_guard guards a clan HQ no_summon cannot be magically summoned no_experience does not give experience if killed no_heal cannot heal damage inflicted on it cannot_fight won't fight back when attacked objectlike for mobs you want to behave like destructible objects invulnerable cannot be physically damaged unkillable will not die, no matter how much damage is done to it

```
    E.g.    act aggro scavenger
                act no_charm no_quest lose_fame
                act mount
aff Affect flags
Keyword list
Default: none
Affect flags define any special abilities or magical/supernatural effects that the mob
is affected by.  If not included, a default value of none is used.  Note some of these
are not really meant to be applied to mobs during mob creation, but rather during
gameplay.  However, they are all included below:
none, zero  no flags
blind   cannot see
sneak   movement not reported
```

```
hide    cannot be seen if still
passdoor    may move through closed doors
invis   is invisible (normal invis)
infrared    has infrared vision
det_evil    can detect evil mobs or players
det_invis   can detect invis
det_magic   can detect magic
det_hidden  can detect hidden mobs or players
det_good    can detect good mobs or players
det_traps   can detect traps
det_sneak   can detect sneaking mobs or players
hold    is trapped, cannot move
sanctuary   has sanctuary spell
globe   has globe spell
protection  has protection spell
faerie_fire has faerie fire spell
flaming has fireshield spell
meditate    is meditating
fly is flying
cursed  can't recall, attacked by mobs with det_curse
poison  is poisoned
sleep   is asleep
charmed is charmed
battle_aura has battle aura (damage reduction)
deter   affected by deter spell
swim    is swimming
plague  is affected by the plague prayer
non_corporeal   does not interact with the world much
swallowed   has been swallowed by a large creature
no_recall      can't recall (but not cursed!)
DOT takes damage every tick update
prone   can't use skills, can still cast
dazed   can't do anything
slow    has been slowed (many negative effects)
```

```
            E.g.    aff sneak hide invis infrared
                aff globe sanctuary flaming fly
                aff poison
sp  Special function
Keyword
Special functions give extra behaviours to mobs either during combat or outside of
combat.  If you don't wish to give your mob a special function, do not include the sp
field at all; if you include an sp field and leave it blank you will produce an error.
Only one special function may be granted per mob:
spec_breath_acid    breathes acid [combat]
spec_breath_fire    breathes fire [combat]
spec_breath_frost   breathes frost [combat]
spec_breath_gas breathes gas [combat]
spec_breath_lightning   breathes lightning [combat]
spec_breath_steam   breathes steam [combat]
```

```
spec_breath_any breathes any of the above at random
spec_buddha random breath weapons and cleric
                                                spells [combat]
spec_guard  attacks killers, thieves or evil
                                                players
spec_kungfu_poison  poison-palm technique [combat]
spec_warrior    warrior skills [combat]
spec_vampire    vampire skills [combat]
spec_mast_vampire   powerful vampire skills [combat]
```

spec_bloodsucker sucks blood [combat] spec_clan_guard guards clan entrance spec_cast_adept healer spec_cast_hooker sexy healer ;) spec_cast_druid casts druid spells [combat] spec_cast_water_sprite casts cleric/mage/psionic spells

spec_cast_cleric casts cleric spells [combat] spec_cast_ghost undead ghost; appears only during night spec_cast_judge fires explosive bullets (a la Dredd)

spec_cast_mage casts mage spells [combat] spec_cast_psionicist casts psionic spells [combat] spec_cast_undead casts undead spells [combat] spec_cast_orb powerful healer spec_cast_archmage casts powerful mage spells

spec_cast_priestess casts powerful cleric spells

spec_cast_chill casts spell chill touch [combat] spec_executioner attacks thieves and killers spec_fido eats corpses spec_guard cityguard spec_janitor gathers rubbish from ground spec_poison poisonous bite [combat] spec_repairman repairs broken doors spec_thief steals coins spec_assassin assassin skills [combat] spec_bounty old grail spec (deprecated) spec_grail teleporting healer and thief/killer assassin spec_scavenger gets objects from ground spec_cleaner gathers rubbish from ground spec_spectral_minion teleporting mob spec for bastion.are—not for general use spec_celestial_repairman a better broken door repairman (teleports) spec_sahuagin sahuagin-based skills/spells

spec_evil_evil_gezhp give to mobs you don’t want players messing with spec_demon demon/infernal spells [combat] spec_cast_electric electricity-based attacks [combat] spec_small_whale flukeslaps [combat] spec_large_whale flukeslaps and swallows [combat] spec_kappa kappa skills/spells [combat] spec_aboleth aboleth skills/spells [combat] spec_laghathti laghathti skills/spells [combat] spec_superwimpy tries very hard to escape combat spec_uzollru uzollru skills/spells [combat] spec_sahuagin_baron sahuagin baron skills/spells

spec_sahuagin_prince sahuagin prince skills/spells

spec_green_grung green grung skills/spells

spec_sahuagin_infantry sahuagin infantry skills/spells

spec_sahuagin_cavalry sahuagin cavalry skills/spells

spec_sahuagin_guard sahuagin guard skills/spells

spec_sahuagin_lieutenant sahuagin lieutenant skills/spells

spec_sahuagin_cleric sahuagin cleric skills/spells [combat] spec_sahuagin_high_cleric sahuagin high priest/shaman skills/spells [combat] spec_red_grung red grung (mage) skills/spells

```
            E.g.    sp  spec_cast_adept
mp  Mob program
Multi-line text block
Mob programs (mob progs, mprogs) are scripts that add special functionality to mobs.
What mob progs are available and what syntax they use is not discussed here, but up to
date documentation is distributed with the MUD□s source, and can be read online here.
```

A single mob prog is defined by each mp tag; you can have as many mp tags per mobile as you like. Mob progs are defined using the following format: mp <mob prog name and arguments> <mob prog code> ~ E.g. mp death_prog 100 mpecho The water begins to thrash! mpmload 2601 mpmload 2601 ~ te Teacher skill Line of text Some mobs are able to train particular skills for players. The mob must have the practice act flag set or these fields will be ignored. You can define as many teacher fields as you like per mob. Use the following format: te <percentage> <skill name> Percentage must be a number that is 0 or higher. Do not quote skill names if they contain multiple words. E.g. te 100 divine magiks te 75 flamestrike Examples

$mob nm imp horrible sh an imp lo A horrible imp prances about the room. de The imp looks horrible, its hairless body a dirty brown colour and its eyes a mucky yellow. ~ vn 0 lv 2

$mob nm gezhp mighty warrior dwarf dwarven sh Gezhp lo Gezhp the mighty dwarven warrior stands afore! de What a fearsome yet attractive fellow this dwarven warrior is... such a magnificent beard, etc. ~ vn 1 lv 150 act sentinel famous no_charm practice aff sanctuary globe flaming det_evil bf no_heart sx male te 100 headbutt te 100 charm te 100 dwarven wrestling al 750 mp rand_prog 10 say Dwarvish? You're not wrong! ~ mp death_prog 100 shout AIEE! I'm done for! ~ sp spec_warrior

1. Objects Objects are the items found in your area; they are either carried or equipped by mobiles or are placed in rooms or container objects; they can be carried and worn by players or be immovable fixtures or features of a room; they can also be trapped (or be traps themselves). They are defined in the #OBJECTS section of the final area file. You don't have to define any objects in your area for it to be valid. Objects are individually defined in single blocks. You can have as many object blocks as you wish. Objects may not share the same vnum: this will produce an error when you run Scribe over your source file.

Block header $obj or $object Fields Field Description Type nm name (keywords) text sh short description text lo long description text vn vnum number ty type keyword v0 value0 variable v1 value1 variable v2 value2 variable v3 value3 variable wg weight number ex extra flags keyword list we wear flags keyword list ed extra description multi-line text block ap apply effect text trt trap trigger keyword list trd trap damage type keyword trc trap charges number Description nm Name (keywords) Line of text The keywords that can be used to indicate the object for manipulation. E.g. nm potion red bubbling sh Short description Line of text The description of the object as it appears when manipulated or in a character's inventory. Don't capitalise any leading 'a', 'the', 'an' etc. E.g. sh a bubbling red potion lo Long description Line of text The description of the object as it appears in the room after a LOOK command. Capitalise and terminate with a full stop, etc. E.g. lo A bubbling red potion lies here. vn Vnum Number: 0 or higher Relative vnum of the object. The first object would usually be zero. ty Type Keyword The object's type; one of the following: light light source

scroll recite for spells wand zap for spells paint smear for spells staff brandish for spells potion quaff for spells pill eat for spells smokeable smoke for spells weapon armour money coins treasure valuables (not coins) furniture trash container holds other items drink_container holds liquids key food boat npc_corpse can be used as a container fountain water fountain climbing_eq for scaling walls, cliffs anvil used for refining armour auction_ticket allows participation in an auction clan clan healing item portal portal to other location poison_powder for poisoning weapons lockpick for picking locks instrument for singing songs (Bards) armourers_hammer for forging armour mithril for crafting a bladethirst weapon whetstone for sharpening weapons craft bonus to crafting skills if object is in room spellcraft bonus to spellcrafting skills if object is in room turret_module for engineer skills forge for smithy skills arrestor_unit for engineer skills driver_unit for engineer skills reflector_unit for engineer skills shield_unit for engineer skills turret for engineer skills defensive_turret_module for engineer skills combat_pulse for engineer skills defensive_pulse for engineer skills pipe to use smokeable substances pipe_cleaner for cleaning pipes remains left behind by objectlike mobs when they are destroyed (similar function to NPC corpse)

```
                                          E.g. ty  potion
        v0-v3   Values
Numbers or text
The four value fields v0, v1, v2 and v3 are used by some types of object.  Some object
types do not use any value fields, and you will not have to include them in the
$object block.  Other types expect certain value definitions and Scribe will report
errors if they are absent or invalid.  You must supply the relevant value fields for
the following objects:
Lights
    v2  Hours of light provided
        Number
        A value below zero indicates infinite hours of light.
    E.g.    ty  light
        v2  -1
Scrolls, potions, paints and pills
    v0  Level of spell(s)
        Number: 1 or higher
    v1-v3   Name of spell(s)
        Text
        You should indicate between 1 and 3 spells.
    E.g.    ty  potion
        v0  10
        v1  heal
        v2  cure poison
Wands and staves
    v0  Level of spell
        Number: 1 or higher
    v1  Maximum charges
        Number: 0 or higher
    v2  Current charges
        Number: 0 or higher
    v3  Name of spell
        Text
    E.g.    ty  wand
```

```
        v0  30
        v1  5
        v2  3
        v3  combat mind
Weapons
    v3  Attack type
        Keyword
        One of the following:
        hit     slice       stab
        slash       whip        claw
        blast       pound       crush
        grep        bite        pierce
        suction chop        rake
        swipe       sting       scoop
mash         hack
    E.g.    ty  weapon
        v3  pound
Containers
    v0  Capacity
        Number: 0 or higher
        How much weight (in pounds) the container can hold.
    v1  Lid flags
        Keyword list
        Whether the container has a lid, and whether the lid is closed or locked:
        none        open, no lid
        closeable   has a lid
        pickproof   lock can't be picked
        closed  lid is closed
        locked  lid is locked
        Weirdness can result if the flags are incorrectly set (e.g. locked but with no
lid).
    v2  Key
        Number: -1 or higher
        The relative vnum of any key.  Use -1 to indicate no key exists.
    E.g.    ty  container
        v0  50
        v1  closeable closed locked
        v2  4
Drink containers
    v0  Capacity
        Number: 0 or higher
        The maximum number of draughts the container can hold.
    v1  Current capacity
        Number: 0 or higher
        Current number of draughts in the container.
    v2  Liquid type
        Keyword
        One of the following:
        water       beer        wine
        ale     dark_ale    whisky
        lemonade    firebreather    local
```

```
        slime_mould milk        tea
        coffee  blood       salt_water
        cola
    v3  Poison
        Number
        zero        not poisoned
        non-zero    poisoned
    E.g.    ty  drink_container
        v0  5
        v1  4
        v2  blood
              v3    0
```

Fountains

```
v2  Liquid type
    Keyword
    One of the following:
    water       beer        wine
    ale     dark_ale    whisky
    lemonade    firebreather    local
    slime_mould milk        tea
    coffee  blood       salt_water
    cola
v3  Poison
    Number
    zero        not poisoned
    non-zero    poisoned
E.g.    ty  fountain
    v2  milk
          v3    1
```

Key v0 Vnum of room/container unlocked Number: 0 or higher Default: 0 Only a convention; not required.

Food v0 Hours of nourishment Number: 0 or higher v3 Poison Number zero not poisoned non-zero poisoned E.g. ty food v0 0 v3 -1

Crafting item v0 Crafting bonus Number: 0 or higher Will be the percentage bonus giving to crafting that takes place in the same room, i.e. “5” == +5% bonus.

```
E.g.    ty  craft
    v0  15
```

Spellcrafting item v0 Spellcrafting bonus Number: 0 or higher Will be the percentage bonus giving to relevant spells that are cast in the same room, i.e. “5” == +5% bonus. E.g. ty spellcraft v0 25

Money

```
All coin amounts are fuzzy unless the pure flag is also applied to the money item.
v0  Copper coins
    Number: 0 or higher
    Default: 0
v1  Silver coins
    Number: 0 or higher
    Default: 0
v2  Gold coins
    Number: 0 or higher
    Default: 0
v3  Platinum coins
    Number: 0 or higher
    Default: 0
E.g.    ty  money
    v2  50
    v1  200
```

Turret Module v0 Lower damage range Number: 0 or higher

```
v1  Upper damage range
    Number: 0 or higher
```

```
v2  Current charges
    Number: 0 or higher
```

```
v3  Maximum charges
    Number: 0 or higher
```

```
E.g.    ty  turret_module
    V0  50
    v1  200
    v2  7
    v3  9
```

Driver unit v0 Number of uses Number: 0 or higher Default: 0 E.g. ty driver_unit v0 79

Shield unit v0 Number of uses Number: 0 or higher Default: 0 E.g. ty shield_unit v0 7 Defensive turret module v2 Module capacity Number: 0 or higher

```
E.g.    ty  defensive_turret_module
    v2  7
```

Combat pulse v0 Current charges Number: 0 or higher v1 Max charges Number: 0 or higher

```
v2  Spell level
    Number: 0 or higher
```

```
v3  Spell
    Text
```

```
    E.g.    ty  combat_pulse
        v0  5
        v1  7
v2  40
v3  fireball
```

Defensive pulse v0 Current charges Number: 0 or higher v1 Max charges Number: 0 or higher

```
v2  Spell level
    Number: 0 or higher
```

```
v3  Spell
    Text
```

```
    E.g.    ty  defensive_pulse
        v0  5
        v1  7
v2  40
v3  protection
```

Pipe v0 Current benefit Number: 0 or higher How much positive or negative benefit your pipe applies to the smokeable when used. v1 Max benefit Number: 0 or higher The maximum positive or negative benefit your pipe can apply to the smokeable (if in top condition).

```
    v2  Effect on thirst
        Number: 0 or higher
How thirsty smoking the pipe makes you.  Lower number is better.
```

```
    v3  Speed
        Number: 0 or higher
The speed of the pipe—how fast you can smoke with it.  Lower is better.
```

```
    E.g.    ty  pipe
        v0  90
```

```
        v1  95
v2  170
v3  30
```

Pipe cleaner v0 Current uses Number: 0 or higher v1 Maximum uses Number: 0 or higher

```
    v2  Current effectiveness
        Number: 0 or higher
How good the cleaner is at cleaning pipes.
```

```
    v3  Maximum effectiveness
        Number: 0 or higher
The best the pipe cleaner can be at cleaning pipes.
```

```
    E.g.    ty  pipe
        v0  17
        v1  18
v2  41
v3  52
```

Portal v0 Room vnum low Number: 0 or higher The lowest vnum the portal can travel to. If v1 is 0, this will be the destination. If < 1 the portal is nonfunctional. v1 Room vnum high Number: 0 or higher Default: 0 The highest vnum the portal can travel to. If > 0 the portal will go to a random room between v0 and v1. v2 Lower use limit Number: 0 or higher Default: 0 The lowest level a character can be to use the portal. v3 Upper use limit Number: 0 or higher Default: 0 The highest level a character can be to use the portal. E.g. ty portal v0 2570 v1 5055 v2 2 v3 70

Smokeable By convention, a smokeable has 3 spells on it, and the third one is harmful to the smoker. This is not server-enforced. v0 Uses remaining Number: 0 or higher v1 Spell Text

```
v2  Spell
    Text
```

```
v3  Spell
    Text
```

```
    E.g.    ty  smokeable
        v0  7
        v1  armor
v2  mental barrier
v3  poison
```

Remains v0 Weight capacity Number: 0 or higher v1 “Lid” flags Keyword list Default: 0 Whether the remains can be closed, and if they are closed: closeable can be closed closed are closed

```
E.g.    ty  remains
    v0  5
    v1  closeable
```

All types not listed above do not require any values to be defined; all relevant variables are calculated by the DD server based on the item’s level. wg Weight Number: 0 or higher The weight of the object in pounds. ex Extra flags Keyword list Any special properties of the object are indicated using the following keywords: none, zero no extra flags glow glows (visual effect) hum hums (visual effect) ego an ego item evil is evil invis is invisible magic is magical trapped is trapped donated has been donated no_drop cannot be dropped no_remove cannot be removed blessed has been blessed (weapon) anti_good cannot be worn by good players anti_neutral cannot be worn by neutral players anti_evil cannot be worn by evil players inventory is a shopkeeper’s inventory item poison is poisoned (extra damage if an item is a weapon) anti_mage cannot be used by mages, warlocks or necromancers anti_cleric cannot be used by clerics, templars or druids anti_thief cannot be used by thieves, bounty hunters or ninjas anti_warrior cannot be used by warriors, thugs or knights anti_psionic cannot be used by psionicists, witches or satanists anti_ranger cannot be used by rangers, barbarians or bards anti_brawler cannot be used by brawlers, monks or martial artists anti_shifter cannot be used by shapeshifters, vampires or werewolves anti_smithy cannot be used by smithies vorpal can be used to decapitate with sharp has been sharpened (weapon) bladethirst is thirsty (weapon) forged has been forged (armour) body_part is a body part; may not be disarmed in combat lance can be used to joust with bow can be used to shoot with deployed item has been deployed (engineer) rune item has had a rune inscribed on it (runesmith) pure item is not randomised at all by server steady item is only weakly randomised by server cursed item is cursed; mobs with det_curse will attack holder

E.g. ex glow magic evil ex poison anti_cleric anti_good we Wear flags Keyword list Wear location information for the object. Any number of the below locations can be given, but to avoid weirdness, choose one of the following combinations: none cannot be picked up or worn; take can be picked up, cannot be worn; take <pos> can be taken and worn in position pos (only one position is given). The following keywords can be used: none, zero cannot be taken or worn take can be taken finger neck body usually heavy, e.g. armour about_body usually light, e.g. shirt head legs feet hands arms shield waist wrist wield weapons except for lances and bows ranged for lances and bows (required for joust and shoot to work) hold held in hand float orbits about head pouch belt-pouch E.g. sh a huge cast iron stove we none sh a sharp dagger we take weapon sh a longbow we take ranged sh a small potted plant we take hold ed Extra description Multi-line text block Extra descriptions are descriptive text seen by characters who examine the item. Extra descriptions consist of a list of keywords and a text block. The text block is printed when a character enters LOOK <keyword> when the item is visible to her. They have the following format: ed <keyword list> <lines of text> ~ The terminating tilde is required. You can give multiple extra descriptions to items. E.g. ed pot plant cactus A small ceramic pot contains a squat, prickly cactus. A large pink flower blooms from its spiny crown. ~ ed pink flower The flower sprouting from the top of the cactus is pretty and fragrant. ~ ap Apply effect Line of text Applied effects are bonuses, penalties or special enhancements given to characters when they wear an item. They have the following format: ap <apply type> <modifier> Apply type is one keyword from the list below; modifier is any number.

Note some of these (marked with a red *) are unimplemented, so applying them to an object is currently purely decorative (will show up on identify, for example). str strength int intelligence wis wisdom dex dexterity con constitution sex * class * level * age * height * weight * gold * exp * hp hit point maximum mana mana point maximum move movement point maximum ac armour class hitroll to-hit modifier damroll damage bonus save_para currently just sums with save_spell save_rod currently just sums with save_spell save_petri currently just sums with save_spell save_breath currently just sums with save_spell save_spell save versus spell save_breath save versus breath fly flight * sneak move undetected * pass_door pass through doors * invis invisibility * det_invis detect invis mobs or players * det_hidden detect hidden mobs or players * flaming fireshield spell * protect protection spell * globe globe spell * sanc sanctuary spell * dragon_aura dragon aura spell * resist_heat resist heat spell * resist_cold resist cold spell * resist_lightning resist lightning spell * resist_acid resist acid spell * breathe_water breathe water spell * balance for smithy skill set_uncommon object set related set_rare object set related set_epic object set related set_legendary object set related strengthen for smithy skill engraved for smithy skill serrated for smithy skill incribed for smithy skill crit increases chance of critical hit swiftness increases chance of bonus attack

(*) The value of the modifier for these applies is not important; 1 is usually used. You may give an item as many applies as you wish. E.g. ap hitroll -4 ap int 5 ap fly 1 trt Trap trigger Keyword list

The event which will trigger a trap installed in the object. This field will be ignored if the trap extra flag has not been given to the object. Use one of the following triggers: room trap will affect everyone in room move movement in any direction triggers trap north movement north triggers trap south movement south triggers trap east movement east triggers trap west movement west triggers trap up movement up triggers trap down movement down triggers trap object Trap triggered on GET <object> or PUT <object> open Trap triggered on OPEN <object> E.g. trt room open trd Trap damage type Single keyword

The type of effect the trap produces after it is triggered. sleep victim sleeps teleport teleports victim away poison poisons victim fire cold acid energy damage inflicted for these types blunt pierce slash trc Trap charges Number: 0 or higher

Number of charges left in the trap. Examples:

$obj vn 0 ty armour nm wooden shield sh a wooden shield lo A wooden shield has been left here. we take shield wg 20

$obj vn 1 nm jewellery box golden sh a golden jewellery box lo A golden jewellery box rests on the floor. we take hold wg 10 ty container ex trap v0 8 v1 closeable closed locked v2 3 trt open trc 1 trd poison

$obj vn 2 ty staff nm staff serpent golden snake sh the Staff of the Serpent lo You see a long golden staff fashioned as a snake. we take hold wg 35 ex glow magic anti_good anti_evil v0 75 v1 5 v2 5 v3 gas breath ed staff serpent golden snake The staff is made of solid gold with small emeralds inset along the shaft. The top end has been fashioned into a beautiful cobra’s head, with a gaping jaw and long protruding fangs. Large emeralds serve as the staff’s eyes, and the instrument glows softly. ~ ap int 5 ap wis 5 ap hp –100 10. Object sets Goes here.

1. Rooms Rooms are distinct locations within your area, and need not be rooms in a literal sense

(inside space with walls, ceiling, floor and doors). They are defined in the #ROOMS section of the final area file. You don't have to define any rooms in your area for it to be valid, although you'll probably want rooms if you want people to adventure in your area! Rooms are individually defined in single blocks. You can have as many room blocks as you wish. Rooms may not share the same vnum: this will produce an error when you run Scribe over your source file.

Block header $room Fields Field Description Type vn vnum number nm name (title) text de description multi-line text block st sector type keyword rf room flags keyword list ed extra description multi-line text block rnd random exits keyword Exit fields In the following list, the symbol ? is replaced by n, s, e, w, u or d for north, south, east, west, upwards or downwards exits. Field Description Type ? exit number ?nm exit name (keywords) text ?de exit description multi-line text block ?lo exit locks keyword list ?ke exit keys number ?ds exit door state keyword Description nm Name (title) Line of text The title of the room as it appears after the LOOK command is issued. Don't use terminating punctuation. Capitalise as desired (although make the initial letter a capital). E.g. nm The Dark Gate nm A narrow, overgrown forest path de Description Multi-line text block The descriptive blurb that is shown after the LOOK command is given. st Sector type Single keyword Default: inside Describes the type of terrain that room has; used to calculate movement penalties, among other effects. Use one keyword from the following list: inside (You probably want to set the indoors room flag too) city field forest hills mountain water_swim Don't need boat/flight to enter water_no_swim Need boat/flight to enter underwater Will drown unless you can breathe underwater air Need flight to enter desert swamp underwater_ground As underwater, but some attacks possible that are not in regular underwater sectors

```
        rf  Room flags
Keyword list
Default: none
Defines any special properties of the room.  Use a list of any of the following
keywords:
none, zero  no room flags
dark    need light source
no_mob  mobs may not enter
                  indoors  sheltered from weather, sunlight
                                      vault             player□s vault can be
manipulated
                  craft             bonus to crafting
                  spellcraft        bonus to spellcrafting
private space for only two creatures
safe    can't pkill
solitary    space for only one creature
pet_shop    pet store
no_recall   can't recall
silence can't cast spells
arena   anyone can pkill, without penalty
healing accelerated healing within room
freezing    players take cold damage per tick
                  burning         players take heat damage per tick
                  no_mount          room may not be entered while mounted
                  toxic             healing slowed, chance of being poisoned
no_drop           objects may not be dropped in room
```

```
            E.g.    rf  safe healing no_mob
        ed  Extra description
        Multi-line text block
Extra descriptions are descriptive text seen by characters who examine the room.
Extra descriptions consist of a list of keywords and a text block.  The text block is
printed when a character enters LOOK <keyword> within the room.  They have the
following format:
            ed  <keyword list>
            <lines of text>
            ~
            E.g.    ed  writing wall
                You read the writing on the wall:
                "Dwarves do it standing up."
                ~
        rnd Random exits
        Keyword
Randomise the exits in the room, so that the room becomes a maze.  Use one of the
following     keywords:
2d  two-dimensional maze (north, south, east and west exist scrambled)
3d  three-dimensional maze (north, south, east, west, up and down exits)
            E.g.    rnd 2d
This field will also accept numbers from 0 to 6 if you require (although it is
recommended you use the 2d or 3d keywords).  You should avoid using random exits if
you are going to include     door resets (see below); these resets may produce
unexpected results.
Exits   There are six possible exits from each room; each room may contain between 0
and 6 exits.
Exit code   Exit direction
n       north
s       south
e       east
w       west
u       upwards
d       downwards
To create an exit simply add the following to your $room block:
<exit code>   <destination room relative vnum>
E.g.
n   0
    u   23
If you don't want a room to have an exit in a particular direction, just leave out the
relevant field.  If you do not add any further information for your exit it will be
valid; however, it won't have a door or any form of description.
The following fields can be used to further define exits.  They should be used in the
following manner:
<exit code><field>  <data>
There is no space between exit code and field.
        ?nm Exit name (keywords)
        Line of text
A list of keywords that describe the exit.  Usually used to indicate a door.  The
first name on the list will be used for generating messages by the MUD server.
            E.g.    wnm door iron reinforced
```

```
                unm reinforced iron door
                dnm path overgrown
Regarding the first two examples above, the first is preferred to the second, as any
messages from the MUD look more natural:
            Cf. "Crash!  You bash open the door!"
                "Crash!  You bash open the reinforced!"
        ?de Exit description
        Multi-line text block
The description of the exit given after the LOOK <direction> or LOOK <exit keyword> is
given.
            E.g.    wde
                The small trail wanders west into the
                heavy forest.
                ~
        ?lo Exit locks
        Keyword
This value is used to indicate whether an exit is a door, and whether the door can be
forced open if locked or passed through if closed.  It can also be used to indicate
whether an exit is a wall or if it is secret (doesn't show up on SCAN etc).  Due to
the way the locks value is read by the server, the keywords used by this field are
rather awkward.  You may enter one of the following keywords; you can also use the
relevant number between 0 and 12.
0   none        no door
1   door        door
2   pick        pick-proof door
3   bash        bash-proof door
4   pick_bash       pick-, bash-proof door
5   pass        pass-proof door
6   pick_pass       pick-, pass-proof door
7   bash_pass       bash-, pass-proof door
8   pick_bash_pass  pick-, bash-, pass-proof  door
9   wall        wall, able to be scaled using climb
10  door_secret     hidden door
11  door_secret_pbp pick-, bash-, pass-proof hidden door
12  secret      hidden exit (not a door)
Door resets will be ignored if you fail to indicate that the exit is a door with one
of these flags (all are doors except for none, wall and secret).
        ?ke Exit key
        Number: -1 or higher
        Default: -1
The relative vnum of the object that can be used to unlock the door.  Use -1 if you
wish there to be no key in existence.  Any value above -1 indicates a relative vnum
(including 0).
            E.g.    eke -1
                nke 4
        ?ds Door state
        Keyword
This field produces a door reset, i.e. will update the position of the door every time
your area is reset.  Indicate one of the following actions:
open    open and unlocked
close   closed and unlocked
```

```
lock    closed and locked
The reset will only be used if the relevant exit has been defined and that exit is a
door.
            E.g.    e   2
                elo door
                eds close
Unless otherwise desired (e.g. one-way doors), use the same door reset in both rooms
sharing a door for consistency.
Examples
```

```
$room
vn  1
nm  The Void
de
You float in the inky darkness of the Void.
~
```

```
$room
vn  10
nm  A small, empty room
de
This tiny room is entirely empty except for a few pieces
of litter against the walls.  You may return to the main
corridor through the southern archway.
~
st  inside
rf  indoors private dark
s   9
```

```
$room
vn  11
nm  At the base of the Outpost Tower
de
You cross the courtyard to the base of the northern wall.
The lean stone tower rises above you; a sturdy wooden door
in the centre of the wall leads into its heart.
     You may head north through the door into the tower,
or head west, south or east back across the courtyard.
~
ed  tower outpost
The tower rises high into the air; it is at least half
a dozen stories high.  Flashes of light flicker from its
peak every now and then.
~
st  city
n   12
nlo pick_bash
```

```
nke 23
nds lock
nnm door sturdy wooden
nde
The door leading into the tower is reinforced with heavy
iron bands.  It looks very solid, and is covered in sharp
studs.  You see a large keyhole in its centre.
~
w   9
e   8
s   6
```

1. Mobile resets A reset refers to an action performed by the MUD to manipulate and update the world's areas. Mobile resets determine how mobiles are loaded into your area: what rooms they appear in, how many may appear and what objects they are wearing or carrying. The $addmob block is used to describe a single mobile reset. You may have as many $addmob blocks as you like in your area. You do not need to have any mobile resets for your area to be valid. You may use any particular rooms and mobs as many times as you wish. Block header $addmob or $addmobile Fields Field Description Type mb mobile vnum number rm room vnum number num maximum mobile number number inv item in inventory number
    - equip item number
    - See below for list of equip fields. Description mb Mobile vnum Number: 0 or higher The relative vnum of the mobile you wish to load. A mob using this vnum must be defined otherwise Scribe will indicate an error. rm Room vnum Number: 0 or higher The relative number of the room you wish to load the mobile into. A room using this vnum must also be defined otherwise an error will result. You can add as many rm fields as you like to a single $addmob block, and identical mobiles carrying and wearing all equipment specified will be loaded into each room (up until the limit specified by the num field). num Maximum mobile number Number: 1 or higher Default: 1 The maximum number of mobiles with this vnum that can be loaded into the world (all areas) at any one time. inv Item in inventory Number: 0 or higher The relative vnum of any object you wish the mob to carry. An object using this must be defined otherwise an error will result. You do not have to include any inv fields in your reset; you may give a reset as many inv fields as you like.
    - Equip an item Number: 0 or higher Equip an object on the mob. The following format is used: <position> <relative vnum> Position is any of the following: light light source finger1 finger2 neck1 neck2 body on body head legs feet hands arms shield about about body waist wrist1 wrist2 wield primary weapon dual secondary weapon ranged ranged weapon (lance, bow) hold held in hand float orbiting head pouch belt pouch You may equip as many items as you wish; if you indicate the same position more than once, the last instance defined will be used. Examples

Load a single naked mob to a room:

```
$addmob
mb  1
rm  3
```

Load a naked mob to rooms 30, 32 and 33 (relative vnums); keep adding a mobile to these rooms every time the area resets until there are 6 mobs in the MUD:

```
$addmob
mb  4
rm  30
rm  32
rm  33
num 6
```

Load and equip a mob:

```
$addmob
rm  3
mb  3
wield   7
inv 6
shield  9
body    8
inv 5
inv 5
```

1. Object resets Object resets determine how objects other than those given to mobiles are loaded into your area: objects that appear on the ground in rooms, and objects that are placed inside other objects. The $addobj block is used to describe a single object reset. You may have as many $addobj blocks as you like in your area. You do not need to have any object resets for your area to be valid. You may use any particular rooms and objects as many times as you wish. Block header $addobj or $addobject Fields Field Description Type ob object vnum number rm room vnum number lv level of object number con container object vnum number Description ob object vnum Number: 0 or higher The relative vnum of the object you wish to load. rm Room vnum Number: 0 or higher The relative number of a room you wish to load the object into. A room using this vnum must also be defined otherwise an error will result. lv Level of object Number: 0 or higher The level you wish the object to be. Only applies if the object is reset to the ground and NOT to a mob or container.

   ```
           con Container vnum
   Number: 0 or higher
   The relative number of a container object you wish to place the item in.  A
   container object with this vnum must be defined for the reset to be valid.
   Only one of each kind of object may occupy any single container.  The object is
   loaded to the most recently loaded container with the specified vnum.  For best
   results, only load one of each container object into your area.
   You need to have at least one rm or one con fields for the $addobj block to be
   valid.
   Examples
   ```

```
$addobj
ob  4
rm  12
```

```
$addobj
ob  2
rm  6
rm  7
con 12
```

2. Games Games stuff here.
3. Helps You can define help file entries in your area using the $help block. You don't have to include any $help blocks in your area for it to be valid. Block header $help Fields Field Description Type he Help text Multi-line text block lv Level restriction number Description he Help text Multi-line text block Define the help entry keywords and body text using the he field: he <keywords or phrases> <help text> ~ The keywords or phrases string is automatically capitalised by Scribe. E.g. he fly levitate This is the text displayed whenever the commands HELP FLY or HELP LEVITATE are entered. ~

```
                he  'aura of fear'
                This is the text displayed whenever the
                command HELP 'AURA OF FEAR' is issued.
                ~
        lv  Level restriction
Number: -1 or higher
The minimum level a character has to be in order to access the help entry.  0 is
used for general help entries.  Use 1 if you want the help entry keyword header to
be hidden.
```

4. Shops Mobiles can be made to run shops: the inventory of the shop is the inventory of the mobile. So long as the mobile is alive, players can attempt to buy and sell items from the shop. Block header $shop Fields Field Description Type vn Vnum of shopkeeper number t1 Traded item type 1 keyword t2 Traded item type 2 keyword t3 Traded item type 3 keyword t4 Traded item type 4 keyword t5 Traded item type 5 keyword ps Profit-sell number pb Profit-buy number oh Opening hour number ch Closing hour number Description vn Vnum of shopkeeper Number: 0 or higher The relative vnum of the mobile that will run the shop. t1-t5 Traded item types Keyword The item types that the shopkeeper will be prepared to buy from players. You may indicate up to five types; you don't have to specify any types for the shop to be valid. Use the same keywords used to define item types. E.g. t1 weapon t2 armour ps Profit-sell Number: 0 or higher The percentage markdown on items sold to the shopkeeper. 100 is the intrinsic value of the item, 75 is a 25% markdown, etc. This value should be at most 100%. pb Profit-buy Number: 0 or higher The percentage mark-up on items bought from the shopkeeper. 100 is the intrinsic value of the item, 150 is a 50% mark-up, etc. This value should be at least 100%. oh ch Opening and closing hours Number: 0  23 The times when the store opens and closes (0 = midnight). Example

```
$shop
vn  0
t1  wand
t2  staff
t3  potion
t4  scroll
oh  7
ch  18
ps  85
pb  120
```