TABLE OF CONTENTS

GENERAL LESSONS

AREA SUBMISSION PROCESS

You should have gotten approval to make the area in the first place. Once you have gotten your area as complete as possible, to the point where it needs testing you should notify the builder's admins. You should areacheck your area (see Building Tools below) and make sure it passes. This is very important because if your area does not pass the check program, then it will crash the testport. Once your area passes area check send your area to areas@forgottenkingdoms.com. Please note that this is not the address to send questions to. Mystra is the only recipient of the areas emails and its sole purpose is to take areas and load them up into the game. If you have questions, you must send it to builders@forgottenkingdoms.com

At this point your area will be allocated vnums, put up on the testport and sent back to you. You must use that version of the area that has been sent to you. You will be given a god character on the testport to test your area with. You will then test your area, and you will find bugs. You will fix the bugs and send back the fixed version of your area to be uploaded. You will be notified by return email (unless you are on the tport at the same time as the builders admins) that your area has been updated to the testport. You will then test your area again, find new bugs and send it back in. All areas have to be sent up more than once. In fact most will get sent up many times. In order to be eligible for testing, an area must have mobiles, objects, rooms and resets completed.

Area Review Process

When you think your area has been tested fully and is ready for the real game it will undergo a review process. Your area will be first sent to Dalvyn who will go over it very carefully looking for problems with the area and making sure that the area meets the building standards. He will then send back an email with a list of changes he wants done. Often he will have suggestions on how to better do something or do something differently, and he will ask questions about what is going on in your area. You will then answer questions and edit your area file, making the changes asked for. In general these changes are not negotiable. We have a set of standards in place to try and keep the game balanced. You will then test your area file, and when you are satisfied with that version you will send it back in for the next stage of review.

Your area is then sent to Tyr, who is one of the coders. He tends to look at areas differently than Dalvyn or Mystra and give a different perspective on what is wrong in the area. He often has suggestions for streamlining your programs. He will then send an email stating the things he wants to see changed. You will make the changes, send up the area for testing, test it, and then send the area back when it is ready for the next stage of review.

The area then comes to Mystra. She will look over the area and make any changes herself, unless they are large ones that take too much time. If that is the case the area will be sent back to you asking for the changes to be made. If they are minor, which is the hope after the other two reviewers have reviewed the area, then Mystra will do all the changes herself. Sometimes Mystra will ask questions of you and suggest better ways to do things. You will recieve an email that will state the changes that were made, and a current version of your area. Mystra will then put the area up onto the real game. From this point you cannot make any changes to your area. The area may get edited over time by the admins to make it conform to new standards or new code. Players may find bugs missed by you and the review process and they may need to be fixed

as well. So the version you have and the version in game will end up being different.

Please note that the review process will take time. The reviewers are all busy people and going over an area to check for problems is time consuming.

Revisiting an area

You might decide sometime later that you want to do a major revision of the area. When making your area in the first place you must do so not with the intent to add to it later, but with the intent to submit a fully completed area that you will not need to rework later. However, sometimes a quest is thought up, additions are thought up after the area has been in the game for a while. If the additions are minor, the area administrators will most likely be the ones to do the editing. If it is major, then your area will be sent to you (you must get current version from the game), the thread about it moved back to the New Areas TODO on the forums (providing there is not too much IC information in the thread), and you will be allowed to edit it. Once complete your area must go through the review process above again. This process will take time to do, just the same as a new area review.

AREA SAMPLE

The following sample area is an extract taken from a real area in the game. It has sample mobiles, with programs, objects and rooms. It also shows how these things are dealt with in resets. You will notice that the justice system does not have matching vnums. This is because this area is part of a city and therefore is protected by the justice system of its parent area.

```
#AREA Roseportal House~
#AUTHOR Dalvyn~
#JUSTICE
CourtRoom 4081
Dungeon 4085
Judge 4070
Crime CRIME_HIGH_MURDER PUNISHMENT_DEATH
Crime CRIME_LOW_MURDER PUNISHMENT_SEVER
Crime CRIME_ASSAULT PUNISHMENT_JAIL
Crime CRIME_MUGGING PUNISHMENT_RANDOM_ITEM
$
#RANGES
0 65 0 65
$
#RESETMSG {D0}You hear priests chanting prayers of hope for the next morning.~
#FLAGS
0 0
#ECONOMY 0 80000
#WEATHER 5 5
#MOBILES
#16201
cleric guard~
{70}a cleric guard~
{70}A cleric guard of the temple stands here.~
This tall and strong human wears a steel breastplate decorated
with the symbol of Lathander. It is one of the warrior-priests
who have sworn to defend the temple and its inhabitants.
~
U 45 CLASS_LATHANDER RACE_HUMAN SEX_MALE POS_STANDING DEITY_LATHANDER
ACT_SENTINEL|ACT_CITIZEN
0
ARMOR_TYPE_FULL_PLATE MATERIAL_STEEL
d8+2 650
13 13 13 13 13 13 13
0 0 0 0 0
```

```
LANG_COMMON
LANG_COMMON
0 0 0
>death_prog 100~
mpjunk all
~
|
#16202
fighter guard~
{70}a figher guard~
{70}A fighter guard of the temple stands here.~
This tall and strong human wears a steel breastplate decorated
with the symbol of Lathander. It is one of the warrior-priests
who have sworn to defend the temple and its inhabitants.
~
U 45 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_LATHANDER
ACT_SENTINEL|ACT_CITIZEN
0
ARMOR_TYPE_FULL_PLATE MATERIAL_STEEL
d10+2 650
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
0 0 0
>death_prog 100~
mpjunk all
~
|
#16203
paladin guard~
{70}a paladin guard~
{70}A paladin guard of the temple stands here.~
This tall and strong human wears a steel breastplate decorated
with the symbol of Lathander. It is one of the warrior-priests
who have sworn to defend the temple and its inhabitants.
~
U 45 CLASS_PALADINS RACE_HUMAN SEX_MALE POS_STANDING DEITY_LATHANDER
ACT_SENTINEL|ACT_CITIZEN
0
ARMOR_TYPE_FULL_PLATE MATERIAL_STEEL
d10+3 1000
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
```

```
0 0 0
>death_prog 100~
mpjunk all
~
|
#16204
gardener~
{30}the gardener~
{30}The temple gardener stands here, seeing that nobody walks on the flowers.~
This rather old man wear a simple brown robe. He is wandering the
gardens of the temple, making sure that nobody walks on the flowers
or on the herbs.
~
S 30 CLASS_FIGHTER RACE_HUMAN SEX_MALE POS_STANDING DEITY_LATHANDER
ACT_SENTINEL|ACT_CITIZEN
%14 1 dig~
>fight_prog 20~
yell Help! I am being attacked by $N on $b in the temple of Lathander!
mpat 4000 yell Help! I am being attacked by $N on $b in the temple of Lathander!
if quest(0,2,self) == 0
  mpecho A guard comes to help $I
  mpmload 16201
  mpforce m16201 mpoload 16239
  mpforce m16201 wield i16239
  mpforce m16201 mpkill $n
  mpmadd self quest 0 2 1
else
  if quest(0,2,self) == 1
    mpecho A guard comes to help $I
    mpmload 16202
    mpforce m16202 mpoload 16235
    mpforce m16202 wield i16235
    mpforce m16202 mpkill $n
    mpmadd self quest 0 2 1
  else
    if quest(0,2,self) == 0
      mpecho A guard comes to help $I
      mpmload 16203
      mpforce m16202 mpoload 16240
      mpforce m16202 wield i16240
      mpforce m16202 mpkill $n
      mpmadd self quest 0 2 1
    endif
  endif
endif
~
```

```
>death_prog 100~
mpmset self quest 0 2 0
mplog WITNESS: $n has killed $I on $b in the temple of Lathander (Berdusk).
mpjunk all
~
|
#16208
tailor temple~
{B0}the temple tailor~
{B0}The temple tailor is weaving some clothes here.~
This old human is clad in a wonderfully woven yellow silk robe. He
seems to be very good at his art, managing to weave perfect clothes
very quickly.
~
S 35 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_LATHANDER
ACT_SENTINEL|ACT_CITIZEN
%15 1 staves~
>fight_prog 20~
yell Help! I am being attacked by $N on $b in the temple of Lathander!
mpat 4000 yell Help! I am being attacked by $N on $b in the temple of Lathander!
if quest(0,2,self) == 0
  mpecho A guard comes to help $I
  mpmload 16201
  mpforce m16201 mpoload 16239
  mpforce m16201 wield i16239
  mpforce m16201 mpkill $n
  mpmadd self quest 0 2 1
else
  if quest(0,2,self) == 1
    mpecho A guard comes to help $I
    mpmload 16202
    mpforce m16202 mpoload 16235
    mpforce m16202 wield i16235
    mpforce m16202 mpkill $n
    mpmadd self quest 0 2 1
  else
    if quest(0,2,self) == 0
      mpecho A guard comes to help $I
      mpmload 16202
      mpforce m16202 mpoload 16240
      mpforce m16202 wield i16240
      mpforce m16202 mpkill $n
      mpmadd self quest 0 2 1
    endif
  endif
endif
```

```
~
>intercept_prog buy~
if deity($n) == Lathander
  mpunintercept
else
  sayto $n I only trade with followers of Lathander.
endif
~
>death_prog 100~
mpmset self quest 0 2 0
mplog WITNESS: $n has killed $I on $b in the temple of Lathander (Berdusk).
mpjunk all
~
|
#0
#OBJECTS
#16200
long yellow robe pink trim~
{B0}a long yellow robe with {D0}pink trim~
{B0}A long yellow robe with {D0}pink trim lies on the ground.~
~
ITEM_TYPE_ARMOR
0
ITEM_WEAR_TAKE|ITEM_WEAR_BODY
ITEM_QUALITY_SUPERIOR MATERIAL_SILK COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_CLOTH 0 0
E
robe yellow pink trim lathander long~
This brightly coloured cloth is made of fine quality silk.
~
>give_prog 100~
mplog GIVEAWAY: $n has given i16200 (robe of Lathander) to $t.
~
|
#16201
yellow silk cap pink trim~
{B0}a yellow silk cap with {D0}pink trim~
{B0}A yellow silk cap with {D0}pink trim lies on the ground.~
~
ITEM_TYPE_ARMOR
0
ITEM_WEAR_TAKE|ITEM_WEAR_HEAD
ITEM_QUALITY_SUPERIOR MATERIAL_SILK COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_CLOTH 0 0
E
yellow silk cap pink trim~
```

```
This brightly coloured cloth is made of fine quality silk.
~
#16235
broadsword sword pink handled steel~
{D0}a pink handled {70}broad sword~
{D0}A pink handled {70}broad sword lies on the ground here.~
~
ITEM_TYPE_WEAPON
0
ITEM_WEAR_TAKE|ITEM_WEAR_HOLD
ITEM_QUALITY_HIGH MATERIAL_STEEL COND_PERFECT SIZE_MEDIUM
0 0 0 WEAPON_TYPE_BROAD_SWORD 0 0
E
broadsword sword pink handled steel~
The blade of this weapon is made of steel. Its handle is
covered with pink paint.
~
#16239
flail pink handled steel~
{D0}a pink handled {70}steel flail~
{D0}A pink handled {70}steel flail lies on the ground here.~
~
ITEM_TYPE_WEAPON
0
ITEM_WEAR_TAKE|ITEM_WEAR_HOLD
ITEM_QUALITY_HIGH MATERIAL_STEEL COND_PERFECT SIZE_MEDIUM
0 0 0 WEAPON_TYPE_FLAIL 0 0
E
flail pink handled steel~
This weapon is made of steel. Its handle is
covered with pink paint.
~
#16240
morningstar star pink handled steel~
{D0}a pink handled {70}steel morningstar~
{D0}A pink handled {70}steel morningstar lies on the ground here.~
~
ITEM_TYPE_WEAPON
0
ITEM_WEAR_TAKE|ITEM_WEAR_HOLD
ITEM_QUALITY_HIGH MATERIAL_STEEL COND_PERFECT SIZE_MEDIUM
0 0 0 WEAPON_TYPE_MORNING_STAR 0 0
E
morningstar star pink handled steel~
This weapon is made of steel. Its handle is
covered with pink paint.
```

```
~
#0
#ROOMS
#16201
Stone corridor~
{B0}An arched hallway circles around the small garden in the center
of this temple. Brightly coloured paintings decorate the walls. There
are many circular holes in the ceiling, to allow the rays of the sun
to light this place.
~
0 ROOM_INDOORS SECT_INSIDE 0 0 0
DDIR_EAST
~
~
0 -1 16200 1
DDIR_NORTH
~
~
0 -1 16202 1
DDIR_SOUTH
~
~
0 -1 16214 1
DDIR_WEST
~
~
0 -1 16241 1
E
paintings brightly coloured~
{90}These paintings depict the Morninglord, Lathander.
~
S
#16202
Stone corridor~
{B0}This arched hallway circles around the large garden in the center
of the temple. The floor is made of small white and yellow marble
squares that reflect the light of the sun coming in through the many
circular holes in the ceiling.
~
0 ROOM_INDOORS SECT_INSIDE 0 0 0
DDIR_NORTH
~
~
0 -1 16203 1
DDIR_SOUTH
~
```

```
~
0 -1 16201 1
S
#16203
Stone corridor~
{B0}The floor of this arched hallway is made of small yellow and white
marble squares. At dawn, when the rays of the sun comes through the
many circular holes in the ceiling, these marble squares reflect the
light and light the place.
~
0 ROOM_INDOORS SECT_INSIDE 0 0 0
DDIR_SOUTH
~
~
0 -1 16202 1
DDIR_NORTH
~
~
0 -1 16223 1
DDIR_WEST
~
~
0 -1 16204 1
S
#16219
Rose Garden~
{A0}Several roses grow in this part of the gardens. Yellow, pink,
red, and white roses move slightly with each breath of the wind,
emitting a very nice perfume. A gravel path circles around the
patches of flowers, to allow visitors and worshippers to wander
around without walking on the flowers.
~
0 0 SECT_FIELD 0 0 0
DDIR_SOUTH
~
~
0 -1 16242 1
DDIR_EAST
~
~
0 -1 16220 1
S
#16220
Garden~
{A0}In this part of the gardens, the healers of Lathander grow special
herbs that are used to brew healing potions. You can see herbs of
```

```
every colours and every sizes here. A gravel path circles around the
patches of herbs, to allow visitors and worshippers to wander
around without walking on the herbs.
~
0 0 SECT_FIELD 0 0 0
DDIR_WEST
~
~
0 -1 16219 1
DDIR_SOUTH
~
~
0 -1 16241 1
S
#16223
Temple Tailor~
{70}This small room is barely decorated. Several clothes hanging
on the wall are displayed for those who would like to buy them.
Most of them have been dyed yellow, pink, or red, the colours of
Lathander.
~
0 ROOM_INDOORS SECT_INSIDE 0 0 0
DDIR_SOUTH
~
~
0 -1 16203 1
S
#16241
Gravel Path~
{70}This small gravel path leads to the main church of Lathander to
the west and to the entrance of the temple to the east. It circles
around several patches of brightly coloured flowers and bushes. The
ground is coverd with several tiny yellow and black stones.
~
0 0 SECT_ROAD 0 0 0
DDIR_EAST
~
~
0 -1 16201 1
DDIR_NORTH
~
~
0 -1 16220 1
DDIR_SOUTH
~
~
```

```
0 -1 16222 1
DDIR_WEST
~
~
0 -1 16242 1
S
#0
#RESETS
M 0 16204 1 16219; gardener in gardens
 G 0 8512 50; sells carrot
 G 0 8521 50; sells red apple
 G 0 8030 100; sells shovel
M 0 16208 1 16223; tailor in tailor shop
 G 0 16200 30; sells yellow robe with pink trim
 G 0 16201 30; sells yellow silk cap with pink trim
S
#SHOPS
16204 ITEM_TYPE_NONE ITEM_TYPE_NONE ITEM_TYPE_NONE ITEM_TYPE_NONE
ITEM_TYPE_NONE 120 70 5 22 ; gardener buys nothing
16208 ITEM_TYPE_ARMOR ITEM_TYPE_NONE ITEM_TYPE_NONE ITEM_TYPE_NONE
ITEM_TYPE_NONE 120 70 5 22 ; tailor buys armor
0
#REPAIRS
16208 ITEM_TYPE_ARMOR ITEM_TYPE_CONTAINER MATERIAL_TYPE_LEATHER
100 1 5 22 ; Tailor repairs armours and containers made of silk/leather
0
#SPECIALS
M 16201 spec_guard
S
#$
```

Make sure that there is an extra line at the end of the file, after the #$ symbol. The area checker will not report it as a bug if there is no such line, but that will make the mud crash.

AREA TEMPLATE

The following is a template of all the headers and sections in an area file. It can be handy to use this when starting your area and filling in the blanks so to speak. What each flag and section means is explained in more detail in the rest of the lessons.

```
#AREA   Areaname~
#AUTHOR Authorname~
#RANGES
0 65 0 65
$
#RESETMSG Reset message to be heard on repop.~
#FLAGS
0 0
#ECONOMY 0 1600293
#JUSTICE
CourtRoom 0
Dungeon 0
Judge 0
Crime CRIME_HIGH_MURDER PUNISHMENT_NONE
Crime CRIME_LOW_MURDER PUNISHMENT_NONE
Crime CRIME_ASSAULT PUNISHMENT_NONE
Crime CRIME_MUGGING PUNISHMENT_NONE
$
#WEATHER 5 5
#MOBILES
#0
#OBJECTS
#0
#ROOMS
#0
#RESETS
S
#SHOPS
0
#REPAIRS
0
#SPECIALS
S
#$
```

HEADER FLAGS LESSON

The typical area file has the following main headers in it. I am using the Waterdeep area as an example. These are the headers that come before the #MOBILES section of the area file. They set alot of the overall and general settings for the area file.

```
#AREA Waterdeep~
#AUTHOR Blythe~
#RANGES
1 65 0 65
$
#RESETMSG {A0}You hear the marching of soldiers patroling the streets.~
#FLAGS
0 0
#ECONOMY 100000 100000
#WEATHER 5 5
#JUSTICE
CourtRoom 8859
Dungeon 8115
Judge 8210
Crime CRIME_HIGH_MURDER PUNISHMENT_DEATH
Crime CRIME_LOW_MURDER PUNISHMENT_SEVER
Crime CRIME_ASSAULT PUNISHMENT_JAIL
Crime CRIME_MUGGING PUNISHMENT_RANDOM_ITEM
$
#MINING MATERIAL_TIN
#LOGGING MATERIAL_WOOD
```

Lets break this up line by line in order to understand what each one means a bit better.

`#AREA Waterdeep~` The area name can be any text. This is the name your area will be known as. When someone types 'areas' on the game this is what they will see as the name of the area. You must finish the area name with a tilda ~.

`#AUTHOR Blythe~` This is the name you choose as the builder of the area. If the area has more than one builder you can put them in as follows:

`#AUTHOR Blythe Jaxom~` - Depending on the length of the names both names can appear on the area listing. If they are too wide then it will make the listing go out of balance. When you are testing your area, be sure to type area to make sure that your author and area name fits into the allotted space in the area listing in game.

```
#RANGES
1 65 0
65
$
```

The first two numbers are the recommended level ranges of the area. There are 50 mortal levels on FKmud. 51 is the hero level and the rest are for immortals. The second two numbers are the enforced levels. For instance the newbie training area is limited up to level 9 by typing 1 9. You must end the #RANGES section with a $.

`#RESETMSG {A0}You hear the marching of soldiers patroling the streets.~` This is the echo heard whenever the area resets. If none is put in then the default messages for each sector are echoed to the room. Note that you can colourise this message.

```
#FLAGS
0 0
```

The first number isnt to be used by regular builders and should be left at 0. (It is used for the wilderness areas). The second number is how long in minutes it takes an area to reset. An area like the newbie temple that resets frequently is set at 3. If left at 0 the game decides on the reset time which is quite a long period of time. A typical value is 15. Areas that are set to -1 never reset (until mud reboots).

`#ECONOMY 100000 100000` The economy of an area is the total amount of copper that all the mobs in the area have available to buy goods from characters (via a player's sell command). Buying goods from a character will add the cost of the object to the economy and selling an object to a mob will subtract the cost from the economy. Small villages and guild/temple type areas should be in the range of 5k-20k. Medium sized towns and cities should range from 20k to 40k. Large cities might range from 40k to 100k.

`#WEATHER 5 5` With these two fields you can set the weather for your area. The first number is the humidity of the area. Humidity is defined in a range from 1 - 10, where 1 is desert and 10 is rain forest. The second number is temperature. Temperature like humidity is defined in a range from 1 - 10, where 1 is frozen and 10 is hot. The default is 5 5.

`#JUSTICE` This denotes the beginning of the justice section of the area file. It consists of 9 lines all up.

- `Courtroom` - This is the vnum of the room in which crime listings and reporting are done.
- `Dungeon` - This is the vnum of the dungeon where offenders are placed when doing jail time.
- `Judge` - This is the mobile who is in the courtroom who take crime reports. This mob also lets characters out of the dungeon when they have served their time.
- `Crime CRIME_HIGH_MURDER PUNISHMENT_DEATH` - The crime is killing a PC, and the punishment is death when caught by a mobile that is flagged spec guard in the specials section of the area file.
- `Crime CRIME_LOW_MURDER PUNISHMENT_SEVER` - The crime is killing a mobile/NPC and the punishment is the severing of a limb when the PC is caught by a mobile that is flagged spec guard in the specials section of the area file.
- `Crime CRIME_ASSAULT PUNISHMENT_JAIL` - The crime is stealing from a PC, and the punishment is jail time when caught by a mobile that is flagged spec_guard in the specials section of the area file.
- `Crime CRIME_MUGGING PUNISHMENT_RANDOM_ITEM` - The crime is stealing from a mobile/NPC and the punishment is the confiscation of a random item from the PC by a mobile that is flagged spec guard in the

specials section of the area file.

Note that you can change the punishment to fit the crime of the area. More information about how to set up a justice system can be found here.

`#MINING MATERIAL_TIN` This sets the material that can be mined in an area. This header does not always need to be used. Its purpose is for underground mining areas where a set metal can be found.

`#LOGGING MATERIAL_WOOD` This sets the material that can be logged in an area. This header does not always need to be used. Its purpose is for forested areas where a set type of tree can be logged.

BUILDING IN FORGOTTEN KINGDOMS

- On line builders lessons
- Builders Discussion Forum
- Current Projects Listing Forum
- Area Check Program

In order to be an area builder on Forgotten Kingdoms one should approach the builders admin team - Mystra and Dalvyn. All the areas on Forgotten Kingdoms are to keep within a Forgotten Realms theme. If you really want to create you should find out what needs creating first. There is a list on our forums (see the link above). We want all areas to fit in together rather than running off on a tangent of their own. Do NOT make an area without consulting us first and then expect it to be automatically accepted. We have very set ideas on how we want our world to look. There is room for creativity but we value the integrity of our world. Also note we do give OLC rights to new builders and being a builder does NOT entitle you to an immortal character in the game. Deity characters and builders are two totally seperate things on Forgotten Kingdoms. OLC is available to builders who have submitted an area and understand how to build offline. OLC only builds an area so far, and the builder needs to be able to finish their area off line. It is suggested that first areas should be small areas, so as to not overwhelm the builder. Bigger areas are best done in OLC by builders who have OLC rights.

Current teams and projects are listed on our discussion board. If you are looking for an idea of what to apply for to build, please take the time to look here. See the links above. These pages are kept up to date. When we get a new builders project happening, we add it to this forum. When we have an idea for a new area that we want in the game we add them to the Areas needing builders forum. There is plenty of variety to choose from.

We have available an area check program that runs under windows, and one that runs in a dos window (though the dos window is less current). Checkout the Area Check page for downloading and instructions. Whenever new hard code is added to the game, the area check program changes. Make sure that you have a current version. The date of current release will be on the area check web page.

Our builders documentation is in the form of lessons on web pages. We do not provide one document. The main reason for this is our code is constantly changing as is our information. It is easier to update an individual lesson for me that constantly uploading an already out of date document. See the link to the lessons above. Lessons are currently being up dated to conform with new code written over the last year or so by the hard coders.

Also you may find it handy to obtain Forgotten Realms campaign material to work with. Reading Forgotten Realms novels gives a good feel for the world as well. *Amazon.com* and *Wizards of the Coast Store* are good places to purchase these from. For out of print items, ebay.com is a good source for them.

In order to keep up with new code updates it is a good idea to check back on a regular basis to see if the lessons pages have been updated. Also announcements are posted on the Forgotten Kingdom's discussion board on the builders and new code releases forum. Active builders are placed in the builder's group which allows them to access the builders discussion

```
d15+15 1000
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
>in_file_prog waterdeepfight.prg~
|
```

Only one program per infile prog. You cannot combine more than one program. When naming an in file prog, make sure that the area name is included in the name so that it is easy to figure out which area the prog is associated with. There are some progs that are available for all areas.

- `>in_file_prog poop.prg~` - This prog can be put on horses so that they will generate poop. It can only go on horses as it generates horse manure.
- `>in_file_prog horsehair.prg~` - This prog can be put on horses that a PC can pluck a horse hair out of the tail for a spell component.

JUSTICE SYSTEM

For a complete justice system, you need five things:

- a judge, who will list all the crime reports
- a courtroom, where this judge shall be
- a dungeon room, that has a locked door and a key to unlock it if need be
- guards that will (attempt to) carry out punishment on crime doers
- witnesses who can report crimes

The last point is quite important. If all your mobs are sentinel, with only one mob per room, it is very likely that the crimes will never be reported. For an effective justice system, you will need wandering citizens and guards. Mobs who report crimes will have to walk to the courtroom to see the judge, so they need a clear path to this room (they cannot open closed doors). Your judge should be in the courtroom.

In the first part of the area file, you should have a block like the following one.

```
#JUSTICE
CourtRoom 4081
Dungeon 4085
Judge 4070
Guard 4021
Crime CRIME_HIGH_MURDER PUNISHMENT_DEATH
Crime CRIME_LOW_MURDER PUNISHMENT_SEVER
Crime CRIME_ASSAULT PUNISHMENT_JAIL
Crime CRIME_MUGGING PUNISHMENT_RANDOM_ITEM
$
```

The first four lines after #JUSTICE list the room vnum for the courtroom, the room vnum for the dungeon, the mob vnum for the judge and the vnum of the guards who wander the area and will come to the aid of any citizen being attacked..

The courtroom is needed for all areas that are to have a justice system. It does not have to be in the actual area. For instance, areas that are associated with Waterdeep like temples and taverns, will use the Waterdeep justice system. This is the room where the command "crime" works.

A judge resides in the courtroom vnum. This is whom the mobiles come to to report the crime. Again the judge does not have to be in the actual area if your area is part of a bigger areas justice system.

The dungeon is not actually needed if you do not use any of the punishments for jailing someone. If you do not use a

dungeon, then this would be set to 0. For a dungeon you will need to have a locked room that is not easily escaped from. PC's who escape before their time is served, will be placed back in the dungeon by spec_guard mobs when caught. A PC could actually get out of the area and never return to the area and avoid serving time if they so wished.

The guards from the guard vnum must be actively in the area. These guards when a citizen is attacked will run from the place where they are to aid the victim. They cannot get through locked doors. The number of guards running to the aid of the victim depends on how many are in the area. It is ideal that there are several guards of this vnum in the area, and that they are not flagged sentinel. Note if the citizen has a fight prog on them they will NOT call for help.

Next it lists the punishment associated to the four types of crime:

- `CRIME_HIGH_MURDER` - is murdering another PC
- `CRIME_LOW_MURDER` - is killing a mob
- `CRIME_ASSAULT` - is attacking (but not killing) a PC/mob
- `CRIME_MUGGING` - is a failed pickpocket/steal attempt

The available sentences are the following ones:

- `PUNISHMENT_NOT_ENFORCED` - no punishment
- `PUNISHMENT_DEATH` - death
- `PUNISHMENT_RANDOM_ITEM` - random item is confiscated
- `PUNISHMENT_SEVER` - random limb is severed
- `PUNISHMENT_JAIL` - 1 hour real time in the dungeon room
- `PUNISHMENT_EXILE` - NOT CODED yet

Setting a mob as a guard is done in the #SPECIAL section of the area file.

```
#SPECIALS
M 16201 spec_guard
M 16202 spec_guard
S
```

Please note that mobs cannot open locked or closed unlocked doors when they are enroute to report a crime. So try and make sure the way is clear for mobs to report most crimes.

MOBILES

SIMPLE OR UNIQUE MOBILES?

There are two choices that a builder has when making a mobile. He/she can choose simple or unique. Simple mobiles have less information and draw alot of their settings from their race. Each race has a special file of settings in the game. It determines things like hit points, number of limbs, what sorts of affects that they might have, what sectors they can move into and so on. When you have earned an immortal character on the testport you can type `showrace` to get a list of the available races in the game. If you want to see more information on that race you would type `showrace racename` (i.e. `showrace elf`).

An unique mobile allows you to set more of your mobiles characteristics. The mobile no longer draws from the race file for most of its information. It will only draw information that is not set in either simple or unique mobiles by the builder. If you make a mobile unique you will also have to give the mobile coins in resets. See the lesson about that in the resets lessons section. DO NOT give coins to a simple mob in resets. The game determines the coinage based on race. You will also need to give the mobile the special affects and things that the mobile has naturally because of its race, for instance, a unique drow mobile will need to have the infravision affect set. Unless, you do not want the drow to have this ability for some reason. It might be one of the reasons why you have chosen to make the mobile unique.

Most mobs that you make will be simple mobs, a unique mob is only needed when you need to set something like the mobiles alignment and resistances.

MOBILE DESCRIPTIONS

The descriptions that a mobile has, has several key elements. A mobile needs to have all of them set. If they are left blank then the tilda's still must be used there as a place marker. Make sure you do put in the tildas ~. The placement of them is very important to the game. Forgetting one will cause your area not to work and the game to crash. Putting in extras does not help the game either. And putting them in the wrong place, will make your area work funny or crash the game. Below is a sample describing the key elements only.

```
#QQ00
key words~
{00}short description~
{00}long description.~
{00}description.
~
```

Lets go over each of those elements:

`#QQ00` Number mobiles from 00, using the QQ method. Later when you are allocated vnums the QQ will be changed to the area vnum. It is important that you start at 00 and not 01. The vnums allocated to you when you get a 100 room vnum block will go from 00 to 99 and for 50 room blocks go from 00 to 49 or 50 to 99. If you start at 01 you do not make full use of the vnums available to you.

`keywords` - The keywords are the words that can be used in commands to identify a mobile. ie `look monster` or `kill monster`. Make sure you pick appropriate keywords as there is nothing more frustrating for a player than trying to figure out how to look at a mobile with inadequete keywords.

`short description` - The short description is what you see when a mobile peforms actions. For instance A monster says, or you hit a monster. Make sure you put in articles such as "a, an or the".

`long description` - This is what a character sees when they walk into the room and see the mob. ie A tall smelly monster stands here. Try if you can to keep this to one line in length. Not all telnet applications word wrap and see anything after the first line.

`description` - This is the longest description. This is what is seen when a character explicitly looks at a mobile. Try and keep this description between 4 and 10 lines in length. Try to avoid putting actions into the description. This is better done with mob programs. For instance saying that the monster lunges at you in the description is best done as a mob program. It is fine for mobiles to put equipment in descriptions (unlike PC characters), because it would be too unbalancing for the game to equip every location on a mobile, we tend to only put one or two items on a mobile. So therefore we encourage builders to describe what the mobile is wearing as well.

{00} - This is the colour code. All mobiles in the game should be colour coded. See the lesson on colour coding areas for more information on our standards and how to do the colour coding. Make sure to NOT colourise the keywords.

A sample mobile description follows.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks.  He is wearing mithril chainmail armour.
~
```

When working on descriptions for the mud, whether they are rooms or mobiles physically put in the enters for an 80 character line. The mud doesnt word wrap and with some clients the player will miss out on the information. So type in descriptions like you are on an old manual type writer. For windows users you might find using dosedit a good way to figure out the screen width to press enter to. A program that I use for offline building is TextPad. It is very good as it shows line numbers and so on.

SIMPLE MOBILES

A simple mobile will likely be the type of mobile you make the most. You only set some of the things possible to be set on a mobile, and the rest is drawn from the mobiles race file. Their coins are worked out from their race file as well, so you do not need to give them any in resets.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. He is dressed in mithril chainmail.
~
S 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE ACT_SENTINEL|ACT_CITIZEN
```

As you can see there is only one line of code (apart from the descriptions) for Simple Mobiles. Here is what each signifies:

`S` - Denotes to the game that this mob is a Simple Mob.

`25` - This is the level of the mobile. With simple mobs if this level is outside the level min/max set in the race file then your mobile will be set up or down in level accordingly. For instance goblins are set to be from level 1 to 10, and if this mobile was a goblin the level 25 would be over-ruled by the game and brought down to level 10. We have done this to set standards in levels of mobs in the game. We do not want to see level 1 dragons and level 50 goblins. Maximum mobile level is 50. We want to keep things realistic. If you have a builder character you can do showrace on the race in question to check the level ranges we have set up for it. If not, you can always ask on the builders forum.

`CLASS_WARRIOR` - This denotes the class of the mobile. This helps determine what sort of attacks the mobile makes in combat. Please refer to the lesson on Mobile Classes.

`RACE_DWARF` - This is the race of the mobile. Now this list is often being added to as builders request new races. The best way to see a current list is to do `showrace` with your immortal character on the builders port. There is a listing of current races but it is only as current as the date stamped on the bottom of the page. There may have been new races made since this lesson.

`SEX_MALE` - There are only three choices of sex. `SEX_MALE`, `SEX_FEMALE` and `SEX_NEUTRAL`.

`POS_STANDING` - This is the default position that the mobile is in. The common one would be standing. Refer to the Postions listing for the full list of positions. Often it is best to use `POS_STANDING` even when a mobile is said to be sleeping or resting. This is because programs will still trigger when the mobile is standing, and you can set the long description to however you like, where as other positions override your long description.

`DEITY_NONE` - A common deity for a dwarf would be Moradin. However, do not put deities on leveling mobiles. This would mean someone could max out their favour too easily. For a full list of deity flags refer to the mobile deity listings.

`ACT_SENTINEL|ACT_CITIZEN` - These are the flags that denote what the mobile acts like. There can be more than one of these flags seperated by the `|` (pipe). Note that any mobiles that are something that a lawful character would not kill they should be flagged `ACT_CITIZEN`. For a list of all the act flags and what they mean refer to the ACT flags listing.

STANDARDS FOR COLOURISING MOBILES

Refer to the lesson on colourising an area for the syntax on putting colour into an area.

Every mobile in the game should be colourised, even if you think that the mobile will be the default colour. There is more than one default colour as players get to choose a colour system on account creation. So while you think a mobile might be one colour for you, it will be another for another player. When the colour code is manually set on the mobile it over-rides all defaults. Below is a sample of a mobile that has colour codes:

```
#QQ00
danilothannmob thann~
{70}Danilo~
{70}Danilo Thann gives you a wicked grin.
~
{70}He is a handsome man, dressed to the hilt in the finest of clothes.
He is from one of the most wealthy families of Waterdeep.
Actually he looks like a bit of a dandy.
~
U 45 CLASS_BARDS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_LEATHER MATERIAL_LEATHER
d15+15 1000
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON|LANG_ELVEN
LANG_COMMON|LANG_ELVEN
RIS_NONE RIS_NONE RIS_NONE
```

In general mobiles will be one colour. Do not colourise the keywords. For one they are not seen by the players, and secondly if you put a colour code in front of a keyword, the keyword will no longer work. Below is a sample of a mobile that has an acceptable use of more than one colour:

```
#QQ01
small brown black rat~
{30}a small brown and {80}black rat~
{30}A small brown and {80}black rat {30}hides here.
~
{30}The body of this small rat is mostly covered with brown hair,
while {80}the hair of its head is much darker.
~
S 5 CLASS_WARRIOR RACE_RAT SEX_MALE POS_STANDING DEITY_NONE
ACT_WIMPY|ACT_NOASSIST
```

The above rat is two coloured, and to give that impression the builder has divided up the short and long descriptions into two colours. Note that he included his articles in the colourisation. Do not make your articles a seperate colour, as that would not be consistent with the rest of the game.

MOBILE CLASSES

Below is a listing of all the possible classes that a mobile can be. The numbers next to each one are the bit vector numbers. You could use those instead of the words but it would make your area file less readable. When a mobile is set to a specific class it will act in that classes special way. Some of these things happen all the time, and some things only in battle. For instance a mobile set to `CLASS_FIGHTER` will kick and punch in battle, whereas a mobile set to `CLASS_MAGES` will cast spells in battle.

As you can see each deity has a class. This is used mostly by priests mobiles of their faith. These mobs will use skills and spells inherant to the priests of that diety. In most instances you should set the mobiles deity to be the same.

Please use CLASS_MONSTER for those mobiles that are monsters and are the kinds of NPC's who have not trained at all in the art of battle.

Warrior Classes	Wizard Classes	Rogue classes
CLASS_WARRIOR 0	CLASS_WIZARD 1	CLASS_ROGUE 3
CLASS_RANGERS 6	CLASS_ILLUSIONISTS 4	CLASS_THIEVES 8
CLASS_FIGHTERS 12	CLASS_INVOKERS 15	CLASS_BARDS 9
CLASS_PALADINS 13	CLASS_MAGES 11	
	CLASS_NECROMANCERS 7	
	CLASS_TRANSMUTERS 14	
	CLASS_ABJURERS	
	CLASS_CONJURERS	
Priest classes	**Priest Classes**	**Other classes**
CLASS_PRIEST 2	CLASS_MASK 18	CLASS_VAMPIRE 42
CLASS_BANE 46	CLASS_MIELIKKI 21	CLASS_MONSTER 43
CLASS_BESHABA 32	CLASS_MORADIN 36	CLASS_NONE -1
CLASS_CHAUNTEA 5	CLASS_MYSTRA 16	
CLASS_CORELLON 35	CLASS_OGHMA 33	
CLASS_CYRIC 23	CLASS_SELUNE 27	
CLASS_GARL 40	CLASS_SHAR 41	
CLASS_GOND 26	CLASS_SUNE 20	
CLASS_GRUUMSH 37	CLASS_TALONA 47	
CLASS_HELM 30	CLASS_TALOS 31	
CLASS_ILMATER 17	CLASS_TEMPUS 22	
CLASS_KELEMVOR 19	CLASS_TORM 38	
CLASS_LATHANDER 24	CLASS_TYMORA 28	
CLASS_LLOTH 34	CLASS_TYR 10	
CLASS_LOVIATAR 29	CLASS_UMBERLEE 44	
CLASS_MALAR 25	CLASS_WAUKEEN 45	
	CLASS_YONDALLA 39	

MOBILE RACES

This race list is current as at the 12th of March 2004. Please note for a current listing you can type `showrace` with your immortal character on the test mud. If there is a race that you think should be there because you cannot find a similar one for your mobile then you can request a new race be made on the builders forum.

Race #	Race Name	Category	Race #	Race Name	Category	Race #	Race Name	Category
0	human	humanoid	41	fish	humanoid	82	ankheg	humanoid
1	elf	humanoid	42	griffon	humanoid	83	air_elemental	humanoid
2	dwarf	humanoid	43	illithid	humanoid	84	fire_elemental	humanoid
3	halfling	humanoid	44	beholder	humanoid	85	earth_elemental	humanoid
4	halfelf	humanoid	45	centaur	humanoid	86	water_elemental	humanoid
5	gnome	humanoid	46	bat	humanoid	87	minotaur	humanoid
6	drow	humanoid	47	spider	humanoid	88	naga	humanoid
7	orc	humanoid	48	ogre	humanoid	89	owlbear	humanoid
8	troll	humanoid	49	ettercap	humanoid	90	rakshasa	humanoid
9	gremlin	humanoid	50	gibberling	humanoid	91	yuanti	humanoid
10	vampire	humanoid	51	gnoll	humanoid	92	scorpion	humanoid
11	goblin	humanoid	52	hobgoblin	humanoid	93	abishai	humanoid
12	lizardman	humanoid	53	remorhaz	humanoid	94	bulette	humanoid
13	bugbear	humanoid	54	fishman	humanoid	95	gith	humanoid
14	kobold	humanoid	55	skeleton	humanoid	96	pixie	humanoid
15	harpy	humanoid	56	stirge	humanoid	97	treant	humanoid
16	giant	humanoid	57	thrikreen	humanoid	98	pegasus	humanoid
17	plant	humanoid	58	wyvern	humanoid	99	faeriedragon	humanoid
18	demon	humanoid	59	lamia	humanoid	100	black_dragon	humanoid
19	undead	humanoid	60	dummy	humanoid	101	blue_dragon	humanoid
20	insect	humanoid	61	ghoul	humanoid	102	green_dragon	humanoid
21	ape	humanoid	62	wight	humanoid	103	red_dragon	humanoid
22	dog	humanoid	63	wraith	humanoid	104	white_dragon	humanoid
23	bird	humanoid	64	mummy	humanoid	105	brass_dragon	humanoid
24	bear	humanoid	65	ghost	humanoid	106	bronze_dragon	humanoid
25	horse	humanoid	66	lich	humanoid	107	copper_dragon	humanoid
26	spirit	humanoid	67	duergar	humanoid	108	gold_dragon	humanoid
27	snake	humanoid	68	hookhorror	humanoid	109	silver_dragon	humanoid
28	slime	humanoid	69	umberhulk	humanoid	110	brown_dragon	humanoid
29	cat	humanoid	70	halforc	humanoid	111	deep_dragon	humanoid
30	lizard	humanoid	71	displacer_beast	humanoid			
31	dragon	humanoid	72	quaggoth	humanoid			
32	gelatin	humanoid	73	myconoid	humanoid			
33	golem	humanoid	74	drider	humanoid			
34	beetle	humanoid	75	rabbit	humanoid			
35	shapeshifter	humanoid	76	zombie	humanoid			
36	pig	humanoid	77	ghast	humanoid			
37	rat	humanoid	78	xorn	humanoid			
38	frog	humanoid	79	panther	humanoid			
39	crab	humanoid	80	wolf	humanoid			
40	magical	humanoid	81	crawling_claw	humanoid			

Race #	Race Name	Category		Race #	Race Name	Category	
112	owl	animal	bird	135	halfdrow	humanoid	halfdrow
113	tabaxi	humanoid	tabaxi	136	halfelfaquatic	humanoid	halfelf
114	troglodyte	humanoid	reptilian	137	gray orc	humanoid	orc
115	corporeal	undead	corporeal	138	mountain orc	humanoid	orc
116	incorporeal	undead	incorporeal	139	orcorog	humanoid	orc
117	goblinoid	humanoid	goblinoid	140	halflingghostwise	humanoid	halfling
118	reptilian	humanoid	reptilian	141	lightfoot halfling	humanoid	halfling
119	lycanthrope	humanoid	lycanthrope	142	halflingstrongheart	humanoid	halfling
120	dwarfarctic	humanoid	dwarf	171	aasimar	humanoid	human
121	dwarfgold	humanoid	dwarf	172	gensaiair	humanoid	human
122	efreeti	outsider	efreeti	173	gensaiearth	humanoid	human
123	shield dwarf	humanoid	dwarf	174	gensaifire	humanoid	human
124	dwarfurdunnir	humanoid	dwarf	175	gensaiwater	humanoid	human
125	dwarfwild	humanoid	dwarf	176	feyri	outsider	feyri
126	elfaquatic	humanoid	elf	177	tanarukk	outsider	orc
127	elfavariel	humanoid	elf	178	tiefling	humanoid	human
128	moon elf	humanoid	elf	179	aarakocra	humanoid	aarakocra
129	sun elf	humanoid	elf	180	kirlanan	outsider	kirlanan
130	elfwild	humanoid	elf	181	shade	humanoid	shade
131	elfwood	humanoid	elf	182	wemic monsterous	humanoid	wemic
132	gnomedeep	humanoid	gnome	183	wormpurple	animal	reptilian
133	gnomeforest	humanoid	gnome	184	slaad	outsider	slaad
134	rock gnome	humanoid	gnome	185	slaaddeath	outsider	slaad

MOBILE POSITIONS

In most instances you will want to use POS_STANDING, even if you want it to appear that your mobile is sleeping or resting. Sleeping mobiles might not trigger quests and so on. Also the position over-rides any long description that you might have. So if you want to have the mobile to look to be doing something while sleeping, it is best to simulate it with the long description, than set the position.

POS_DEAD	0	POS_MORTAL	1
POS_INCAP	2	POS_STUNNED	3
POS_SLEEPING	4	POS_MEDITATING	5
POS_RESTING	6	POS_KNEELING	7
POS_FIGHTING	8	POS_STANDING	9
POS_MOUNTED	10	POS_SHOVE	11
POS_DRAG	12		

MOBILE DEITIES

It is best to not set deities on mobs that are used for leveling. It could be abused as a quick means of gaining favour. In general if you have set the class to be that of a priest of a deity, then set the deity as well.

DEITY_NONE	0	DEITY_CHAUNTEA	5	DEITY_TYR	10
DEITY_MYSTRA	16	DEITY_ILMATER	17	DEITY_MASK	18
DEITY_KELEMVOR	19	DEITY_SUNE	20	DEITY_MIELIKKI	21
DEITY_TEMPUS	22	DEITY_CYRIC	23	DEITY_LATHANDER	24
DEITY_MALAR	25	DEITY_GOND	26	DEITY_SELUNE	27
DEITY_TYMORA	28	DEITY_LOVIATAR	29	DEITY_HELM	30
DEITY_TALOS	31	DEITY_BESHABA	32	DEITY_OGHMA	33
DEITY_LLOTH	34	DEITY_CORELLON	35	DEITY_MORADIN	36
DEITY_GRUUMSH	37	DEITY_TORM	38	DEITY_YONDALLA	39
DEITY_GARL	41	DEITY_SHAR	42	DEITY_UMBERLEE	43
DEITY_WAUKEEN	44	DEITY_BANE	45	DEITY_TALONA	46

MOBILE ACT FLAGS

Act flags define how the mobile acts. How they act while fighting and act in the way they go about their business of being a mobile. There are more flags than are listed here, but these are the only ones you will use as a builder.

ACT_SENTINEL	The mobile stays in one room
ACT_SCAVENGER	The mobile picks up objects on the ground
ACT_IS_HEALER	Mobile is a healer using the "heal" command
ACT_AGGRESSIVE	The mobile attacks PC's when they enter the room the mob is in
ACT_STAY_AREA	The mobile will not leave the area it has been loaded into. If this is not set the mobile can wander into other areas unless stopped by a nomob flag on a room or an exit.
ACT_WIMPY	The mobile flees when hurt. Many pets and mounts will be flagged with this, unless they are bred for war. In addition mobiles with this flag, will submit to the demand command.
ACT_PET	For mobiles that are to be pets and animals that can be claimed. Do not use on horses or anything that should be considered a mount.
ACT_UNDEAD	For undead mobiles. Make sure to use this on undead mobs as it is used by the favour system.
ACT_NOSHOVE	Mobile cannot be shoved. This is important for mobiles who should always be found in a certain spot.
ACT_NOFIGHT	Mobile will not fight back. Part of the killmode code.
ACT_BANK	Mobile is a banker
ACT_NOWANDER	Doesn't wander outside the sector type it is loaded in
ACT_MOUNTABLE	The mobile can be mounted.
ACT_SECRETIVE	The mobile's actions are not seen.
ACT_CITIZEN	Mobile is a citizen and affects a character's lawful status if killed. This is also used in the games justice system.

ACT_MOBINVIS	Mobile's cannot be seen by mortals even with detect invis. It is like an immortals wizinvis. Can only be seen by immortals.
ACT_NOASSIST	Does not assist other mobiles or characters in battle.
ACT_REQUEST	The armour that the mobile wears can be requested by good aligned with the request command. Do not use on shop-keepers.
ACT_NOCORPSE	The mobile drops no corpse on death. Ideal for dummies.

UNIQUE MOBILES

An unique mobile sets all the things that a simple mobile does. So everything that has gone before in previous lessons applies here, except instead of an S, a U is used to denote that the mobile is unique and not simple. In addition to what the simple mobile has, the unique mobile has more lines of code setting more features of the mobile. A unique mobile draws less information from the race file than a simple one does. Because of this you may need to set things for that mobile that its race should have, like infravision for underground races.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. He wears mithril chainmail.
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
```

Unique mobiles have the same first two lines of code as simple mobs except it starts with a U to denote that it is a unique mobile. As you can see there are several more lines of code setting information about the mobile. Lets go through each line:

AFF_BERSERK - This is the affect flags section. It determines what spells and so on the mobile is affected by. A mobile can be affected by more than one affect. Seperate each one with a | (pipe). Some simple mobs may be affected by spells if it is in their race characteristics. For instance simple mobile elves are affected by sneak. You can see what a race is affected by with the showrace command on the test port. For a complete list of affect flags look at here.

ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL - Every mobile in the game has what is called *Virtual Armour*. This armour is worn on the body and helps equal up the fight between NPC and PC. This virtual armour does not fall from

the mobile on death. For simple mobs what type and material the virtual armour has is determined by the race and class file. The dwarf in our sample mobile is wearing mithril platemail virtual armour. If you decide that they should wear a piece of armour that players can get, the armour you put on them in resets will over-ride the virtual armour. You should still set the virtual armour to what you want so that if your mobile loads without the object for some reason they are wearing appropriate armour. For a list of armour types refer to here and for a list of material types refer to here.

`d10+2 500` - The d10+2 refers to hit point dice. The formula is level x (1-10 + 2) hit points or whatever numbers you prefer to use. Hit dice size depends on the class and race of the mobile.

- Wizards use d4
- Rogues use d6
- Priests use d8
- Warriors use d10

For mobs where the race is more important than the class (generally all mobs whose race is not a player character race), we use the guidelines of Dungeons and Dragons - 3rd edition:

- d6 for small fey creatures
- d10 for beasts ("animals" that do not exist on Earth), constructs (golems, automata), magical beasts ("animals" with magical abilities), oozes (slimes, jellies)
- d12 for dragons and undead creatures
- d8 for nearly all the other creatures (aberrations - "alien" creatures that do not resemble animals, animals, elementals, giants, humanoids with no specific class, monstrous humanoids like centaurs, minotaurs, kuo-toan, yuan-ti, creatures from other planes, plants, and vermin)

The second number (2 in the example) represents the relative toughness of the mobile in comparison with "normal" mobiles (a dwarven guard would be tougher than a dwarven cook for example). Each mobile also gets bonus hit points for high constitution, in addition to the numbers given here.

The 500 refers to the mobiles alignment. For a list of alignments and what numbers they are refer to here. Note this field should ALWAYS be numeric.

`13 13 13 13 13 13 13` - This line sets the statistics in the following order: str int wis dex con cha lck. You can choose values between 3 and 22. They affect many things that a mobile can do, like what they can wield and carry. A safe set of values is one that is all 13's. This gives the mobile average statistics. Take care not to go overboard with these values. Our dwarf has average statistics. Dexterity affects the number of objects that BOTH NPC and PC's can carry. A mobile who is going to be a shopkeeper does NOT have to have a high dexterity in order to carry all the items. It has been hardcoded that shopkeepers bypass this requirement.

`LANG_COMMON|LANG_DWARVEN` - This is the languages that the mobile SPEAKS. This is what he UNDERSTANDS. This is very important in questing mobiles and shop keepers. Mobiles can know more than one language. They need to be seperate by the `|` character. Refer to here for a list of languages.

`LANG_COMMON|LANG_DWARVEN` - This second line is the languages that the mobile is SPEAKING. If you use sayto $n in your mobile progs it will determine what language the PC is speaking and then speak in that language IF it knows it. If you just use say on its own in mobile progs it will speak the first language in its list reguardless of whether the PC knows it or not.

`RIS_MAGIC RIS_NONE RIS_NONE` - The first RIS_NONE refers to the resistances that the mobile has, the second are the mobs immunities, and the third its susceptabilities. Our sample dwarf is resistant to magic. For a complete list of possibile resistances/immunities/susceptabilities refer to here.

MOBILE AFFECT FLAGS

Affect flags define more attributes of the mobile like the spells that affect them.

Affect Flag	Description
AFF_BLIND	Mobile is affected by blindness
AFF_INVIS	Mobile is invisible
AFF_DETECT_EVIL	Mobile can detect evil
AFF_DETECT_INVIS	Mobile can detect magic
AFF_DETECT_MAGIC	Mobile can detect magic
AFF_DETECT_HIDDEN	Mobile can detect hidden creatures
AFF_DETECT_BURIED	Mobile can detect buried
AFF_SANCTUARY	Mobile is protected by sanctuary
AFF_FAERIE_FIRE	Mobile is affected by faerie fire
AFF_INFRARED	Mobile has infravision
AFF_CURSE	Mobile is cursed
AFF_POISON	Mobile is poisoned
AFF_PROTECT	Mobile is protected by protection
AFF_PARALYSIS	Mobile is paralyzed
AFF_SNEAK	Mobile is sneaking
AFF_HIDE	Mobile is hiding
AFF_SLEEP	Mobile is affected by sleep spell
AFF_CHARM	Mobile is charmed
AFF_FLYING	Mobile is is flying
AFF_PASS_DOOR	Mobile can pass door
AFF_FLOATING	Mobile is floating
AFF_TRUE_SIGHT	Mobile is affected by truesight
AFF_FIRESHIELD	Mobile is affected by fireshield
AFF_SHOCKSHIELD	Mobile is affected by shockshield
AFF_ICESHIELD	Mobile is affected by iceshield
AFF_BERSERK	Mobile is berserk when fighting
AFF_WATER_BREATHING	Mobile is affected by water breathing
AFF_GUARDIAN	Mobile wakes you if someone walks in while you sleep

MOBILE ALIGNMENT

You must use the numeric value in the Mobile Code.

ALIGNMENT	VALUE	RANGE
Lawful Good	1000	775 to 1000
Neutral Good	650	550 to 774
Chaotic Good	450	325 to 549
Lawful Neutral	200	100 to 324
True Neutral	0	-99 to 99
Chaotic Neutral	-200	-324 to -100
Lawful Evil	-450	-549 to -325
Neutral Evil	-650	-774 to -550
Chaotic Evil	-1000	-1000 to -775

Older characters that were created in the first year of mud operation, will not have the exact numbers from the first coloum. This is because when the game first started, actions affected actual alignment. Later on we created a hidden alignment, and chosen alignment was not changed by coded events. It now requires an immortal to decide if a character is not acting their alignment by looking at the hidden alignments. However, those older characters will be within the range for their alignments, and if you check the exact number, you may exclude these older characters from the if check. Many of these older characters are still being played.

When checking for a PC's alignment you have several options available to you. You can use the `if align($n) == 0` check or you can use the `if isgood($n)` checks. There is more information about this in the lessons on if checks.

MOBILE RESISTANCES

RIS_MAGIC RIS_NONE RIS_NONE - The first RIS_NONE refers to the resistances that the mobile has, the second are the mobs immunities, and the third its susceptabilities. Below is the list of resistances available.

RIS_FIRE	Spells that are firebased, such as fireball burning hands etc
RIS_COLD	Spells that are cold based, such as chill touch etc
RIS_ELECTRICITY	Spells that are electricity based, such as call lightning etc
RIS_ENERGY	Spells that are energy based, such as disintergrate etc.
RIS_BLUNT	Resistant or suseptible to blunt weapons, such as maces etc.
RIS_PIERCE	Resistant or suseptible to piercing weapons, such as daggers etc
RIS_SLASH	Resistant or suseptible to slashing weapons, such as swords etc
RIS_ACID	Spells that are acid based such as, acid blast etc.
RIS_POISON	Resistant or suseptible to poison, including potions, poisoned foods and drinks, poisoned weapons and poison based spells
RIS_DRAIN	Spells that a are necromantic based, such as vampiric touch etc.
RIS_SLEEP	Spells that are sleep based, such as the sleep spell etc.
RIS_CHARM	Spells and skills that are charm based, such as charm person, charm spell, influence etc.
RIS_HOLD	Spells that are hold based, such as hold, web, paralysis
RIS_NONMAGIC	Resistant or suseptible to non magical weapons
RIS_SUMMON	Resistant or suseptible to being summoned
RIS_HOLY	Resistant or suseptible to healing based spells.
RIS_MENTAL	Resistant or suseptible to mental based spells, revive, mind wrench etc

RIS_DROWNING	Resistant or suseptible to physical attacks like earthquake, drowning
RIS_LIGHT	Resistant or suseptible to light based spells such as sunray etc
RIS_SOUND	Resistant or suseptible to sound based spells such as, sonic resonance etc
RIS_MAGIC	Resistant or suseptible to magical weapons
RIS_WOOD	Resistant or suseptible to non magical weapons made of wood.
RIS_SILVER	Resistant or suseptible to non magical weapons made of silver
RIS_IRON	Resistant or suseptible to non magical weapons made of iron

MOBILE SIZE AND GOLD

Size - Size is not determined by the builder. The size of your mobile is determined by the race of the mobile. That way all goblins will be small, and all giants large. Use the `showrace` command on the testport to see what size a particular race is.

Gold - As a builder you do not set the coins that the mobile carrys in the actual mobile code. For simple mobiles coin is calculated automatically depending on what the settings are for coins in the race file. For unique mobiles it is set with resets. See the lesson about Coin Resets for more information on how to give mobiles coins. Please be reasonable in the amount of coin given to your unique mobiles. If you are unsure as to the amount to give, then you can type `showrace racename` on the test mud to see what the standard coinage is for that race. As a rule of thumb the coinage amount is based on 10 copper per level. Dragon type mobs obviously will have more than that.

MOBILE PROGRAMS

Mobile programs are commonly termed "mob progs". This lesson does not go into great detail on what mob progs do, more information on that can be found on the section on the lessons pages devoted to that. This lesson focuses on the placement of the program in the mobile's information. Mobile programs work on BOTH simple and unique mobiles.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. Be sure to hold your nose as he stinks!
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
>fight_prog 15~
say By Moradin's Hammer I will defeat you!
~
|
```

Our dwarf has a fight program that has a 15% chance of triggering in which he says 'By Moradin's Hammer I will defeat you!'. Possible triggers for mobile programs can be found on the Mob Progs Listing.

MOBILE SKILLS AND SPELLS TRAINERS

On Forgotten Kingdoms we don't use ACT_TRAIN and ACT_PRACTICE. We are of the belief that no one person knows all skills. So our trainers are set up to know a few skills in a specialised area. For instance a weapons trainer might teach some weapon skills but not wizard spells. As a rule of thumb a mobile will know no more than 3 skills (we dropped this down from the original 6 now that the game has grown to the point where trainers are numerous). If someone is not of the right guild/class for a particular skill when they type train in the game that skill will not be listed. Please make sure you give your mobile appropriate things to train. For instance it does not make sense the a dwarven warrior would train magic missile or charisma.

No area should be a one stop training point for any class. So lets say you were building a thieves guild, you cant go and put all of the thieves skills in your area. The game has been designed that characters should seek trainers from all over, not just in one spot. Builders doing their first area often have each mob training several skills and its pretty much a one stop shop, and the area gets sent back to builder, or the designer told, thats too many skills etc.

There are some skills/spells/feats that are limited or have trainers that are hard to get to. I am not going to list them individually but here is the general stuff.

- Trades are not put on a trainer like other skills. Trades are only learnt by a quest. See the lessons on trade skills and writing trade quests for more information.
- Transport spell trainers must be a quest to learn from or find. Or put on a book or scroll that can be written into the spellbook that was a quest reward.
- Brew and scribe are now quests to learn. Do not put these skills on your mobs.

And if you are not sure, email the builders admins and ask. If you know a skill or feat to be hard to find, ASK first.

The skills known to a mob are added to the end of their mobile information before any mobile programs. Lets give our dwarf a few skills that he can train.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. He is wearing mithril chainmail armour.
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
%10 1 dig~
%10 2 short axes~
>fight_prog 15~
say By Moradin's Hammer I will defeat you!
~
|
```

```
%10 1 dig~
%10 2 short axes~
```

Our dwarf can train dig and short axes. Every skill listing goes at the end of the general mobile information before the mobile program. It is preceded by a %. The 10 in this case means he can train the skill up to a skill level of 10. Max skill level is 25. Trainers with the ability to train to 25 should be very very rare if at all. Most trainers would train below 12.

The next figure (1 for dig and 2 for short axes) determines the price. The actual price of a skill is dependant on the level of the skill with 50 copper being the cheapest price. The 2 in short axes denotes that he charges twice the price. So if short axes is a level 1 skill which is 50 copper or 1 silver, the trainer doubles it to charge 100 copper or 1 gold. This ability to charge more is very handy if you want the trainer to be able to train the skill to a high level. For instance if a trainer can bring a skill up to a skill level of 12 then he might charge 3 times the normal price. It means that players might want to seek out cheaper trainers. Also if a skill is high level in the game we dont want the skill to be that cheap to train so upping the price works in that situation too.

Spells use the same syntax as the skills. Our dwarf is not a mage or a cleric and therefore he does not cast spells, and he should not train them either.

SPELLS LISTING

Below is a list of all the available spells in alphabetical order. Next to each spell in () are two stats. The first stat is the main one that affects the success of the spell, and the second affects the spell but not as much as the first.

You can find out what guilds can learn what spells by typing `showguild guildname` on the testport. Or there is an up to date listing in the guilds section of the web site.

Spell Name	Spell Name
acid arrow (Int,Dex)	cure critical (Wis,Lck)
acid blast (Int,Dex)	cure light (Wis,Lck)
acid breath (Int,Lck)	cure poison (Wis,Lck)
alertness (Int,Lck)	cure serious (Wis,Dex)
animate dead (Wis,Int)	curse (Int,Wis)
animate object (Wis,Lck)	delayed blast (Int,Lck)
antimagic shell (Int,Lck)	detect buried (Int,Lck)
armor (Int,Lck)	detect evil (Wis,Lck)
astral walk (Int,Lck)	detect hidden (Int,Wis)
barkskin (Wis,Lck)	detect invis (Int,Wis)
blazebane (Int,Lck)	detect magic (Int,Wis)
bless (Wis,Lck)	detect poison (Wis,Int)
blindness (Int,Lck)	disintegrate (Int,Lck)
blood of cyric (Wis,Lck)	disjunction (Int,Lck)
bulls strength (Int,Lck)	dispel evil (Wis,Lck)
burning hands (Int,Dex)	dispel magic (Wis,Int)
call lightning (Wis,Lck)	divinity (Wis,Lck)
cause critical (Wis,Dex)	dragonskin (Wis,Con)
cause light (Wis,Dex)	dream (Int,Lck)
cause serious (Wis,Dex)	earthquake (Wis,Lck)
chain lightning (Int,Lck)	eltsacs fear (Int,Lck)
change sex (Int,Dex)	enchant armor (Int,Lck)
charged beacon (Int,Dex)	enchant weapon (Int,Lck)
chariot of the sun (Wis,Lck)	energy drain (Int,Lck)
charm monster (Int,Cha)	entangle (Wis,Lck)
charm person (Int,Cha)	ethereal flyer (Int,Lck)
chill touch (Int,Dex)	faerie fire (Wis,Lck)
clairvoyance (Int,Lck)	faerie fog (Wis,Lck)
color spray (Int,Lck)	farheal (Wis,Lck)
comprehend languages (Int,Lck)	fatigue (Int,Lck)
cone of cold (Int,Dex)	feeblemind (Int,Lck)
conjure elemental (Wis,Lck)	find familiar (Int,Lck)
continual light (Wis,Int)	find traps (Wis,Lck)
control undead (Wis,Lck)	fire breath (Int,Dex)
control weather (Wis,Int)	fireball (Int,Lck)
create food (Wis,Lck)	fireshield (Int,Lck)
create object (Wis,Lck)	flame arrow (Int,Lck)
create spring (Wis,Lck)	flame jaws (Int,Lck)
create water (Wis,Lck)	flamestrike (Wis,Lck)
cure blindness (Wis,Lck)	fly (Int,Lck)

Spell Name	Spell Name
freedom of movement (Wis,Dex)	monster summon (Int,Lck)
friends (Int,Cha)	moonbeam (Wis,Lck)
frost breath (Dex,Lck)	nondetection (Wis,Lck)
fumble (Int,Lck)	null sphere (Int,Lck)
gas breath (Dex,Lck)	pass door (Int,Lck)
gate (Int,Lck)	pass plant (Wis,Lck)
globe of invulnerability (Int,Lck)	phantasmal killer (Int,Dex)
good fortune (Wis,Lck)	phoenix claw (Int,Lck)
hand of chaos (Int,Dex)	poison (Wis,Lck)
harm (Wis,Dex)	polymorph (Wis,Lck)
heal (Wis,Lck)	possess (Int,Lck)
heroism (Int,Lck)	produce flame (Wis,Lck)
hold monster (Int,Lck)	protection (Wis,Lck)
hold person (Int,Lck)	quantum spike (Int,Lck)
holy sanctity (Wis,Lck)	rainbow pattern (Int,Lck)
holy symbol (Wis,Lck)	raise dead (Wis,Lck)
ice storm (Int,Lck)	razorbait (Int,Lck)
iceshield (Int,Lck)	recharge (Int,Lck)
identify (Int,Lck)	refresh (Wis,Lck)
ill fortune (Wis,Lck)	regenerate (Wis,Lck)
ilmaters bless (Wis,Con)	remove curse (Wis,Lck)
infravision (Int,Wis)	remove trap (Dex,Lck)
invis (Int,Lck)	resilience (Int,Lck)
invisibility purge (Int,Lck)	resist cold (Wis,Con)
knock (Int,Dex)	resist electricity (Wis,Con)
know alignment (Wis,Lck)	resist fire (Wis,Con)
levitate (Int,Lck)	restoration (Wis,Int)
lightning bolt (Int,Lck)	restore mana (Int,Wis)
lightning breath (Dex,Lck)	resurrection (Wis,Lck)
llanthyrs mend (Int,Dex)	revive (Wis,Lck)
locate object (Int,Lck)	sagacity (Wis,Lck)
magic mirror (Int,Lck)	sanctuary (Wis,Lck)
magic missile (Int,Lck)	scorching surge (Int,Lck)
magnetic thrus (Int,Lck)	sentry of helm (Int,Con)
mass invis (Int,Lck)	shadow conjuration (Int,Dex)
mind blank (Wis,Con)	shadow door (Int,Lck)
mind wrack (Int,Int)	shadow fist (Int,Lck)
mind wrench (Int,Int)	shadow funnel (Int,Lck)
mirror image (Int,Lck)	shadow walk (Wis,Lck)
mnemonic enhan (Int,Lck)	shield (Int,Lck)

Spell Name	
shocking grasp (Int,Dex)	touch of justice (Wis,Lck)
shockshield (Int,Lck)	transport (Int,Lck)
silence (Wis,Lck)	trollish vigor (Int,Lck)
sleep (Int,Lck)	true sight (Wis,Int)
slink (Int,Lck)	turn undead (Wis,Lck)
sound burst (Int,Lck)	valiance (Wis,Lck)
speak with dead (Wis,Lck)	vampiric touch (Int,Dex)
spectral fist (Int,Dex)	ventriloquism (Int,Lck)
spectral hand (Int,Dex)	warhorse (Wis,Cha)
spectral light (Int,Lck)	water breathing (Wis,Lck)
stone skin (Int,Lck)	water to wine (Int,Lck)
stone walk (Wis,Lck)	weaken (Wis,Lck)
sulfurous spray (Int,Lck)	web (Int,Lck)
summon (Int,Lck)	wind walk (Wis,Lck)
sunray (Wis,Lck)	winter mist (Int,Lck)
swordbait (Int,Lck)	witch light (Int,Lck)
tayzas acidshield (Int,Lck)	word of recall (Wis,Lck)
teleport (Int,Lck)	wraithform (Int,Lck)

SKILLS LISTING

Below is a list of all the available skills in alphabetical order. Next to each skill in () are two stats. The first stat is the main one that affects the success of the skill, and the second affects the skill but not as much as the first.

You can find out what guilds can learn what spells by typing `showguild guildname` on the testport. Or there is an up to date listing in the guilds section of the web site.

Skill Name	Skill Name
aid (Wis,Lck)	listen (Int,Con)
bash (Str,Lck)	parry (Dex,Str)
circle stab (Dex,Lck)	pick lock (Dex,Lck)
companion (Wis,Lck)	rage (Con,Lck)
detrap (Dex,Lck)	sap (Str,Dex)
disarm (Dex,Str)	second attack (Str,Dex)
dodge (Dex,Lck)	shapechange (Wis,Con)
dual wield (Dex,Lck)	sneak (Dex,Lck)
fifth attack (Dex,Str)	sting (Lck,Lck)
grip (Dex,Str)	tail (Lck,Lck)
hide (Dex,Lck)	third parry (Str,Dex)
influence (Cha,Lck)	use magic device (Int,Wis)
limber (Lck,Lck)	backstab (Dex,Lck)
mount (Dex,Wis)	brew (Int,Lck)
peek (Dex,Lck)	climb (Dex,Str)
punch (Str,Dex)	cook (Wis,Int)
riposte (Dex,Str)	dirtkick (Dex,Lck)
search (Int,Lck)	disguise (Lck,Lck)
second parry (Str,Dex)	dual backstab (Str,Dex)
slice (Wis,Dex)	feed (Lck,Lck)
steal (Dex,Lck)	gouge (Str,Dex)
swim (Con,Str)	handle animal (Cha,Wis)
third dodge (Dex,Lck)	ignite (Wis,Int)
trip (Str,Dex)	layonhands (Wis,Lck)
animal empathy (Cha,Wis)	meditate (Int,Wis)
bite (Lck,Lck)	pathfinding (Wis,Con)
claw (Lck,Lck)	poison weapon (Dex,Lck)
concentration (Int,Wis)	rescue (Con,Lck)
dig (Str,Lck)	scribe (Int,Dex)
discern (Wis,Lck)	second dodge (Dex,Lck)
doorbash (Str,Lck)	sing (Cha,Lck)
enhanced damage (Str,Dex)	spellcraft (Int,Wis)
fourth attack (Dex,Str)	stun (Str,Dex)
haggle (Cha,Wis)	third attack (Dex,Str)
hitall (Str,Dex)	track (Wis,Lck)
kick (Str,Dex)	

WEAPONS LISTING

Below is a list of all the available weapon skills in alphabetical order. Next to each weapon skill in () are two stats. The first stat is the main one that affects the success of the weapon usage, and the second affects the weapon usage but not as much as the first.

You can find out what guilds can learn what weapon skills by typing `showguild guildname` on the testport. Or there is an up to date listing in the guilds section of the web site.

bows	(Str,Dex)	brawling	(Str,Dex)
chains	(Str,Dex)	clubs	(Str,Dex)
crossbows	(Str,Dex)	double-edged blades	(Str,Dex)
great blades	(Str,Dex)	great chains	(Str,Dex)
lines	(Str,Dex)	long axes	(Str,Dex)
long spikes	(Str,Dex)	mounted polearms	(Str,Dex)
polearms	(Str,Dex)	rope weapons	(Str,Dex)
shieldwork	(Str,Dex)	short axes	(Str,Dex)
short blades	(Str,Dex)	short spikes	(Str,Dex)
single-edged blades	(Str,Dex)	slings	(Str,Dex)
staves	(Str,Dex)	thrown projectiles	(Str,Dex)
thrusting blades	(Str,Dex)	whips	(Str,Dex)

MOBILE LEVEL TRAINERS

Leveling on Forgotten Kingdoms is not automatic. A player needs to find a trainer and train level. We will give our dwarf the ability to train level.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. Be sure to hold your nose as he stinks!
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
%20 2 level~
>fight_prog 15~
say By Moradin's Hammer I will defeat you!
~
|
```

`%20 2 level~` - Levels work similar to skills. The first figure in this case however is the level that the mobile can train the PC up to. In this case he can only train till level 20. High level trainers are few and far between in the game. Maximum training level is 50. The second figure is the price factor. This trainer charges twice the base price.

MOBILE STATISTIC TRAINERS

On Forgotten Kingdoms a player gets 1 stat point every 5 levels. These points can be used to improve statistics or learn languages. Lets make our dwarf train constitution. Please note that no trainer should train luck. Game policy dictates that luck cannot be gained with the train command.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. Be sure to hold your nose as he stinks!
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
%17 2 constitution~
```

`%17 2 constitution~` - Again training a statistic is much like any other skill. Our dwarf trains constitution up to 17. There are now no limits as to how high a statistic can be trained, however, trainers that train statistics over 18 are very rare and very expensive and often a quest to get to or train with. Generally mobiles in lower level areas will only be able to train up to 15. The cost of training is determined by the game and can be increase by a factor up to 5 as determined by the builder. Our dwarf charges twice the going rate.

The possible statistics to be trained are:

- Constitution
- Strength
- Wisdom
- Intelligence
- Charisma
- Luck
- Dexterity

MOBILE LANGUAGE TRAINERS

The code for making a language trainer is the same as any other skills. It costs a player gold and experience to learn a language. If their skill in the language is low they have a small percentage chance of understanding someone speaking the language. If a mobile teaches a language it will show up with the "train" command.

We can make our dwarf teach dwarven now. Refer to the Languages Listing for a list of languages that can be taught by mobiles.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. Be sure to hold your nose as he stinks!
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
%12 2 dwarven~
>fight_prog 15~
say By Moradin's Hammer I will defeat you!
~
|
```

He teaches dwarven up to a skill level of 12 for twice the going rate.

LANGUAGES LISTING

Below is a list of the languages that can be learned. Some guilds cannot learn certain languages. All languages rely on intelligence to be used, and to some degree luck.

ancient	(Int,Lck)	grandmaster	animal	(Int,Lck)	grandmaster
aquan	(Int,Lck)	grandmaster	thieves cant	(Int,Lck)	grandmaster
auran	(Int,Lck)	grandmaster	common	(Int,Lck)	grandmaster
darkspeak	(Int,Lck)	grandmaster	draconic	(Int,Lck)	grandmaster
dwarven	(Int,Lck)	grandmaster	elven	(Int,Lck)	grandmaster
giant	(Int,Lck)	grandmaster	gith	(Int,Lck)	grandmaster
gnoll	(Int,Lck)	grandmaster	gnome	(Int,Lck)	grandmaster
goblin	(Int,Lck)	grandmaster	halfling	(Int,Lck)	grandmaster
ignan	(Int,Lck)	grandmaster	insectoid	(Int,Lck)	grandmaster
magical	(Int,Lck)	grandmaster	orcish	(Int,Lck)	grandmaster
sylvan	(Int,Lck)	grandmaster	terran	(Int,Lck)	grandmaster
abyssal	(Int,Lck)	grandmaster	celestial	(Int,Lck)	grandmaster

LANG_COMMON	1
LANG_ELVEN	2
LANG_DWARVEN	4
LANG_SYLVAN	8
LANG_DARKSPEAK	16
LANG_ORCISH	32
LANG_ABYSSAL	64
LANG_AQUAN	128
LANG_INSECTOID	256
LANG_AURAN	512

LANG_GIANT	1024
LANG_DRACONIC	2048
LANG_THIEVES	4096
LANG_MAGICAL	8192
LANG_GOBLIN	16384
LANG_GOD	32768
LANG_ANCIENT	65536
LANG_HALFLING	131072
LANG_CLAN	262144
LANG_GNOLL	524288

LANG_GITH	1048576
LANG_GNOME	2097152
LANG_ANIMAL	4194304
LANG_CELESTIAL	8388608
LANG_IGNAN	16777216
LANG_INFERNAL	33554432
LANG_TERRAN	67108864

MOBILE FEAT TRAINERS

The code for making a feats trainer is similar to the code for other skills. But instead of costing gold and experience, it costs gold and a feat point. Characters recieve 1 feat point every 5 levels. If a mobile teaches a feat it will show up with the "train" command. Most feats can only be trained once, as in they use one feat point. So in general, the first number is going to be 1. There are a couple of exceptions to this. Like toughness. The character can train this feat more than once. So you can use a number up to 5 for the number of times the mobile is prepared to train this feat with the character. The second number like with other skills etc, is the price factor.

Some feats are regionally restricted or restricted in other ways. Read the help file of a feat before putting it on a mob to make sure they should be able to train it.

We can make our dwarf teach improved bash. Refer to the Feats Listing for a list of feats that can be taught by mobiles. It is important to check the help files for each feat before using it on a trainer. Some feats have restrictions on who can learn them. For instance some feats are only good for characters from specific locations and you will want to not train those feats if your mobile is not from those locations.

```
#QQ00
smelly dwarf short~
{30}a short smelly dwarf~
{30}A short smelly dwarf is drinking an ale here.~
{30}He is very short for a dwarf with dark hair and a dark beard.  He has a
stubby nose and beady eyes.  His hands are large and calloused from
many hours of work at the forge.  He is of a very sturdy build with arms
and legs like small tree trunks. Be sure to hold your nose as he stinks!
~
U 25 CLASS_WARRIOR RACE_DWARF SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_CITIZEN
AFF_BERSERK
ARMOR_TYPE_PLATE_MAIL MATERIAL_MITHRIL
d10+2 500
13 13 13 13 13 13 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_MAGIC RIS_NONE RIS_NONE
%1 5 improved bash~
>fight_prog 15~
say By Moradin's Hammer I will defeat you!
~
|
```

FEATS LISTING

More information about each feat can be found either by typing help featname in the game or by doing showfeat featname with your builder character on the testport.

awareness	improved critical	silent spell
armor proficiency	improved disarm	silver palm
blind-fight	improved familiar	skill focus
blooded	improved initiative	smooth talk
bloodline of fire	improved trip	spellcasting prodigy
bullheaded	improved brawling	spell focus
cleave	luck of heroes	spell mastery
combat casting	maximize spell	spell penetration
courteous magocracy	militia	spirited charge
daylight adaptation	mind over body	stealthy
deflect arrows	mounted archery	still spell
empower spell	mounted combat	strong arm
endurance	persistent spell	strong soul
enlarge spell	point blank shot	sunder
evasion	poison resist	teacher
expertise	power attack	thug
extend spell	precise shot	toughness
extra turning	quick draw	trample
far shot	quicken spell	twin spell
foe hunter	rapid shot	twin sword style
great cleave	run	weapon focus
heighten spell	saddleback	weapon proficiency
improved bash	shadow weave magic	
improved counterspell		

MOBILE TRADE TRAINERS

Trades are not trained like any other skill in the game. Our policy is that they require a quest to learn. And the reward from the quest is a couple of points or so in the trade skill. Often a character can do more than one quest in the same trade and thereby improve his or her skill by doing the quest and not just by actually working the trade. Please note that the code for the trades skills are within mobile programs, not in the skills listing of what the mobile trains.

The command to give the character a skill point in the trade they are doing or have done the quest in is as follows:

```
mppractice $n 5 tradename
```

For example `mppractice $n 5 mining` will give one point in the trade and only if the PC has no more than 5 skill points in the trade already. The low number is ideal for low level quests expected to be the first quest that a PC will complete in that trade. That number will have to be raised for quests that characters can do in addition to other quests in that trade. It also depends on what kind of abilities the mobile has. If the mobile is not a master in their trade, then they should not be able to teach those are better in the skill than they are. If you want to give more than one point in the trade, then you will have to repeat the command in the program.

For a listing on available trades see the trades list. Also if you want to put a trades quest in your area you should run it past the Area Administrators FIRST.

TRADES LISTING

Please make sure to speak with the Area Administrators before adding any trade quests to your area. The first statistic in the listing is the statisit that most affects that trade, and the second statistic affects the trade, but not to the same degree as the first.

appraise	(Int,Wis)	armorsmithing	(Str,Dex)
fletching	(Dex,Str)	lapidary	(Dex,Lck)
leathermaking	(Dex,Str)	logging	(Lck,Str)
mining	(Lck,Str)	smelting	(Str,Con)
tanning	(Dex,Str)	weaponsmithing	(Str,Dex)
woodworking	(Str,Dex)	clothmaking	(Dex,Str)
herbalism	(Wis,Dex)	tailoring	(Dex,Wis)

BARDSONG TRAINERS

Bardsong is a skill that only bards can use. When they use the sing command they can elect to sing a specific song that will have affects on the bard and possibly the bard's party. Bardsong can only be learned via quests. The quest should only be directed at bards. If you wish to put a quest in your area to teach a bardsong, you will need to get approval from the area administrators first. The command to put in the quest programming to teach bardsong to the PC is as follows:

```
mpsetsong $n 'songname' #
```

- `mpmsetsong` - This is the command to teach the bardsong to the PC.
- `$n` - This is the target. In general it should always be $n.
- `'songname'` - This is the name of the song being taught.
- `#` - This is the number of lines of the song that the PC is being taught. In general teach one verse at a time.

  Verses are generally 4 lines in length. So to teach the first verse this number would be 4, to teach the second this number would be 8.

Bard characters are encourage to write bardsongs. If they are interested they should email the publications team. For a list of current bardsongs available see here.

Make sure to add the following mplogs

- `mplog QUESTCOMPLETE: $n has learned the first verse of Song of Bardsongname.`
- `mplog BARDSONG: $n has learned the first verse of Song of Bardsongname.`

BARDSONG LIST

A current list of available songs can be found on the testport by typing `showsong` with your builder character. If you wish to put a quest in your area to teach a bardsong, you will need to get approval from the area administrators first.

song of heroism
shadows cloak
song of selune
helms watch
elemental compassion

MOBILE KNOWLEDGE SKILLS TRAINERS

Knowledge skills are taught differently to other skills. They are generally given as a part of a quest reward, with the exception of knowledge geography. The syntax for giving one point in a knowledge skill is as follows:

```
mppractice $n 25 knowledge-geography
```

You also should add the following logging condition:

```
mplog KNOWLEDGE: $n has gained 1 point in knowledge geography for traversing the Sea of Moving.
```

Unlike other skills, knowledge skills do not improve with usage. They are more of a measure of what the character has done and learned. The knowledge skills are primarily used by the prestige classes to make sure that the character meets certain criteria in what they know.

`knowledge-history` - This knowledge skill should be given out with quests that teach some sort of history about the kingdoms or forgotten realms in general.

`knowledge-nature` - This knowledge skill should be given out with quests that teach something about nature. For instance herbalism etc.

`knowledge-planes` - This knowledge skill will be harder to obtain. Any quest that has something to do with the planes should give out a point in knowledge planes.

`knowledge-religion` - Any quest that teaches information about faiths in general or a faith should give out a point in knowledge religion. If you are building a temple you are strongly encouraged to have the mobiles there teach about the god and its dogma.

`knowledge-arcana` - This is for quests that teach about magic in general.

Knowledge Geography Mobile Sample

Unlike other knowledge's Knowledge Geography is not a quest reward. It is a measure of how much the character has travelled the kingdoms. Scattered throughout the wilderness are mobinvis (unable to be seen by mortals even with detect invis) mobiles. These mobiles are set up so they wander only in one sector type by adding the flag `ACT_NOWANDER`. The number of passes past the mobile before the point is given depends on the intelligence of the character. The lower the intelligence, the more passes it will take before it is considered they have learned the region. Quest bits need to be set, so that the character only gets one point from each region. Right now in the game there is more than 25 of these mobiles

around the game in the wilderness and underground areas. This means that it is possible to become a grandmaster easily in this skill just by wandering around and learning the wilderness. Most areas will not teach this skill, unless it is a large section of tunnels that underground races will traverse. Check with the builders admins before adding knowledge geography mobiles to your area.

```
#QQ00
sea moving knowledge geography mob~
the Sea of Moving knowledge geography mob~
The Sea of Moving Knowledge Geography mob stands here.~
This mob raises the PC in one point in knowledge geography when they
meet the conditions for the number of times passed and the intelligence
required.
~
U 1 CLASS_MONSTER RACE_HUMAN SEX_NEUTRAL POS_STANDING DEITY_NONE
ACT_NOWANDER|ACT_SECRETIVE|ACT_MOBINVIS|ACT_NOASSIST
AFF_DETECT_INVIS|AFF_DETECT_HIDDEN|AFF_INFRARED|AFF_TRUESIGHT
ARMOR_TYPE_CLOTH MATERIAL_CLOTH
d1+1 0
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
>greet_prog 100~
if questr(50000,0,2,$n) < 3
  if int($n) > 17
    mpmset $n questr 50000 0 2 3
  else
    if int($n) >= 15
      if questr(50000,0,2,$n) > 0
        mpmset $n questr 50000 0 2 3
      endif
    else
      if questr(50000,0,2,$n) > 1
        mpmset $n questr 50000 0 2 3
      endif
    endif
  endif
  if questr(50000,0,2,$n) == 3
    mpechoat $n As you move about the Sea of Moving you feel your geographical knowledge
of the re$
    mppractice $n 25 knowledge-geography
    mplog KNOWLEDGE: $n has gained 1 point in knowledge geography for traversing the Sea
of Moving.
  else
```

```
    mpmadd $n questr 50000 0 2 1
  endif
endif
~
|
```

KNOWLEDGE SKILLS LISTING

Please refer to the lesson on Knowledge Skill Trainers, as knowledge skills are not taught like normal skills.

knowledge-geography	(Int,Wis)	knowledge-history	(Int,Wis)
knowledge-nature	(Int,Wis)	knowledge-planes	(Int,Wis)
knowledge-religion	(Int,Wis)	knowledge-arcana	(Int,Wis)

OBJECTS

OBJECTS - BASIC INFORMATION

Below is a sample object, and then following that is an explanation of each field in the object.

```
#QQ00
ring pink ice~
{D0}a pink ice ring~
{D0}A pink ice ring is lying here.~
~
ITEM_TYPE_TREASURE
FLAG_GLOW|FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_FINGER
QUALITY_AVERAGE MATERIAL_GOLD CONDITION_PERFECT SIZE_MEDIUM
0 0 0 0 LAYER_ALL 0
E
pink ice ring~
{B0}It is golden ring with {D0}a pretty pink gem in it.
~
A APPLY_STR 1
>wear_prog 100~
mpechoat $n The ring feels cold on your finger
~
|
```

Lets look at the above item, a ring line by line. A note to builders who have been with us for a while. Some of the syntax for the flags have changed. The old syntax will still work, but the lessons have been updated to the new syntax. OLC will print out your area in the new syntax.

`#QQ00` - This is the VNUM of the object. As with rooms and mobiles start at 00. You can only have as many objects in your area as you have vnums allocated. ie If you have a 100 room area then you will be limited to 100 objects.

`ring pink ice~` - These are the keywords of the object. This is the word that is used in order to interact with the object. ie get ring or look pink or drop ice. Make sure you have adequete keywords from your short and long descriptions of the object. There is nothing more frustrating for a player who sees a pile of gold fabric on the ground which is a celestial robe when worn but you have neglected to add gold or fabric to the key words and they don't know what to pick up.

`{D0}a pink ice ring~` - This is the short description that a players sees when they type inventory or they use the object. The first character should be lower case as it is often used in the middle of sentences. Make sure you use articles where applicable (ie a, then, an). Note that the short description is colourised. The {D0} makes the item pink in colour. All

objects should be colourised. See the lesson on object colourisation for more information on our colourisation standards.

`{D0}A pink ice ring is lying here.~` - This is the long description of the object. It is what the player sees when they enter the room and the object is visable. The long description of the object should always be colourised.

`~` - This is the action description. It is not used by FKMud code so leave it blank and just put in the tilda ~.

`ITEM_TYPE_TREASURE` - This determines item type. See the list of item types for all possible types. You must give your objects appropriate types. For instance rings are treasure and not armour.

`FLAG_GLOW|FLAG_MAGIC` - This is for extra attributes of the object. You can define more than one flag by seperating them with a pipe `|`. If an item has any sort of magical apply or program that makes it act magical you MUST flag it `FLAG_MAGIC`. See here for a list of flags. Note to older builders, this is one of the fields that has had a syntax change. The ITEM part of the flag has been removed.

`CAN_WEAR_TAKE|CAN_WEAR_FINGER` - The `CAN_WEAR_TAKE` flag defines if the object can be picked up. Most objects in the game will need to have this flag. For items like fountains you wouldn't have this flag. The next defines the wear locations. You can have more than one location. However more than two locations is strongly discouraged except in special situations. If there is no wear location for an object just put a 0 in for field. For a list of wear locations look here. Note to old builder, the syntax of the wear flags has changed and the ITEM has been replaced with CAN.

`QUALITY_AVERAGE MATERIAL_GOLD CONDITION_PERFECT SIZE_MEDIUM` This line states the quality of the object, the material it is made from, its condition and its size. Refer to here and here for more information. Note to older builders, the QUALITY flag no longer has the word ITEM in it.

`0 0 0 0 LAYER_ALL 0` - These are the values of the object. See the table here for more information. Each different object has different uses for those values. Many types leave them unused or some of the unused. Refer to the Layer Lesson for more about Layers. Each of the zeros has a field name, the first is VALUE0, the second VALUE1, and it continues to the last one being VALUE5. VALUE5 is not used by any of the objects. Builders are free to use it in object programs to check the status of an object.

```
E
pink ice ring~
It is golden ring with a pretty pink gem in it.
~
```

This is the extra description. The `E` denotes that there is an extra description. You can leave off all of these 4 lines if you wish to have no extra description. It is encouraged on Forgotten Kingdoms that you do have extra descriptions as there is nothing worse than typing look ring and seeing you see nothing special about a pink ice ring. The second line of the extra description is the key words. A player would type look keyword to see the extra description. The third line is the actual description. Place the tilda on the line below it as if you dont the description looks cramped on the game. An item can

have more than one description with different key words. For instance you could refer to runes in the first description, and then have an extra description with the keyword of runes that would show more information to the player. An object can have more than one extra description. The keywords need to be different for each each description.

`A APPLY_STR 1` This is where you can put special attributes of an item. This ring adds 1 strength. For a list of applies see here. Note that we ask you keep your applies to a minimum and do not apply damroll or hitroll to armour items. Damroll and hitroll are only for weapons. There is the rare exception to this. Any exception should be run by the builder admins. Magic items should not be very easy to get. They should involve an area or immortal driven quest to obtain.

```
>wear_prog 100~
mpechoat $n The ring feels cold on your finger.
~
|
```

This is where you would put in the mud program on the object. See the lesson on object programs for a listing of possible programs on objects.

Please note that builders do not set the weight, cost and level of the objects. These things are determined by the hard code looking at things like item type, material, condition and so on. If you want to change the weight and value of an item from what the hard code sets it to you will need to use applies. There must be good IC reason for to change these attributes of an object with applies.

COLOURISING OBJECTS

Refer to the lesson on colourising an area for the syntax on putting colour into an area.

The best way to get a feel for how objects should be colourised, is look at objects in the game. Look at your characters. The main thing to note is that we do not colourise the articles a different colour. We tend to make the object all one colour, unless there is reason for more than one colour. Below are some samples for colourising objects in your area:

```
#QQ00
sign waterdeep~
{30}a sign~
{30}A sign is here. ~
~
ITEM_TYPE_TRASH
0
0
QUALITY_AVERAGE MATERIAL_WOOD COND_VERY_GOOD SIZE_MEDIUM
0 0 0 0 0 0
E
sign waterdeep~
{00}.{30}               ------------------------------.
{00}.{30}              |    Welcome to Waterdeep      |
{00}.{30}              |                              |
{00}.{30}              |       Created by Blythe      |
{00}.{30}              |                              |
{00}.{30}              .------------------------------.
{00}.{30}                  |  |               |  |
{00}.{30}
~
```

In the object above, note the use of the 00 colour. This allows an invisible character to be placed at the start of the line. If just spaces were put in, the game would ignore them and take the first non space character it finds to the start of the line. The object below has more than one colour. The words that have to deal with the embossing on the armour are in yellow {B0}. This helps to give the impression of more than one colour on an object. None of the articles have been left a different colour.

```
#QQ20
black scale mail armour gold crescent embossed~
{B0}embossed {80}black scale armour~
```

```
{B0}Gold crescent embossed {80}black scale mail armour lies here. ~
~
ITEM_TYPE_ARMOR
0
ITEM_WEAR_TAKE|ITEM_WEAR_BODY
ITEM_QUALITY_AVERAGE MATERIAL_METAL COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_SCALE_MAIL 0 0
E
black scale mail armour gold crescent embossed~
{80}It is a black scale mail armour {B0}embossed with an upturned gold
crescent moon {70}surrounded by 9 silver stars.
~
```

OBJECT TYPES

The 'item-type' is the type of the item (weapon, armor, potion, etc). Depending on the item type, value0 through value5 will have different meanings. Any value that is not used is set to 0. EX. for a light value2 is the number of hours left until the light burns out. Value5 on all objects is unused by hard code and can be used by builders in object programs to set and check the status on an object. As a result Value5 is not shown on this table.

A spell number of zero or negative value means 'no spell'.

Bit Vector/Item Type	Value0	Value1	Value2	Value3	Value4
1 ITEM_TYPE_LIGHT	unused	unused	hours left, 0 is dead, -1 is infinite. Infinite lights are to be rare magical items.	unused	unused
2 ITEM_TYPE_SCROLL	level of spell/s *	spell number 1	spell number 2	spell number3	unused
3 ITEM_TYPE_WAND	level of spell	max charges	charges left	spell number	unused
4 ITEM_TYPE_STAFF	level of spell	max charges	charges left	spell number	unused
5 ITEM_TYPE_WEAPON	unused	weapon flag	weapon flag modifiers	Weapon Type	unused
7 ITEM_TYPE_SHEATH	capacity in pounds	container flags	key vnum	unused	layers
8 ITEM_TYPE_TREASURE	unused	unused	unused	unused	layers
9 ITEM_TYPE_ARMOR	unused	unused	layers	Armor type	unused
10 ITEM_TYPE_POTION	level of spells	spell number 1	spell number 2	spell number 3	unused
12 ITEM_TYPE_FURNITURE	unused	unused	Furniture State	unused	unused

13 ITEM_TYPE_TRASH	unused	unused	unused	unused	unused
15 ITEM_TYPE_CONTAINER	capacity in pounds	container flags	key vnum	unused	layers
17 ITEM_TYPE_DRINKCON	total amount of drinks	current amount of drinks	liquid #	component/herb value	junks on use or not
18 ITEM_TYPE_KEY	unused	unused	unused	unused	unused
19 ITEM_TYPE_FOOD	nourishment value	decay timer	FOOD_RAW or FOOD_COOKED, 0 is raw	component/herb value	unused
20 ITEM_TYPE_MONEY	# of coins	coin type	unused	unused	unused
21 ITEM_TYPE_PEN	amount of ink	unused	unused	unused	unused
23 ITEM_TYPE_CORPSE_NPC	unused	unused	decomposition timer	25 * Race Size	unused
24 ITEM_TYPE_CORPSE_PC	unused	unused	unused	unused	unused
25 ITEM_TYPE_FOUNTAIN	unused	Amount of drinks	Liquid Type	unused	unused
26 ITEM_TYPE_PILL	level of spells	spell number 1	spell number 2	spell number 3	unused
27 ITEM_TYPE_BLOOD	unused	quantity	decay timer	unused	unused
28 ITEM_TYPE_BLOODSTAIN	unused	unused	decay timer	unused	unused
29 ITEM_TYPE_SCRAPS	unused	unused	decay timer	unused	unused
30 ITEM_TYPE_PIPE	maximum capacity of	amount of herb in the	herb type	pipe flags	unused

	pipe	pipe			
34 ITEM_TYPE_FIRE	unused	unused	hours left, 0 is dead, -1 is infinite	unused	unused
35 ITEM_TYPE_BOOK	unused	spell number	unused	Language	Skill Level (From 1 to 25)
37 ITEM_TYPE_LEVER	lever trigger flag	vnum of teleport room or spell number or start room or room to be randomised	room to load the mob or object into	object or mob to be loaded	unused
39 ITEM_TYPE_BUTTON	lever trigger flag	vnum of teleport room or spell number	unused	unused	unused
44 ITEM_TYPE_TRAP	trap type	number of reloads	trap trigger	unused	unused
45 ITEM_TYPE_MAP	unused	low room vnum	high room vnum	unused	unused
46 ITEM_TYPE_PORTAL	unused	unused	unused	unused	unused
47 ITEM_TYPE_PAPER	text status	subject status	to status	language number	language skill level
57 ITEM_TYPE_PROJECTILE	unused	weapon flag	weapon flag modifiers	Weapon Type	unused
58 ITEM_TYPE_QUIVER	capacity in pounds	container flags	key vnum	unused	layers
59 ITEM_TYPE_SHOVEL	unused	unused	unused	unused	unused

60 ITEM_TYPE_SALVE	level	Number of uses	unused	herb type	spell slot number
61 ITEM_TYPE_SYMBOL	NO. spell component uses	unused	unused	unused	unused
62 ITEM_TYPE_TRADEGOODS	unused	unused	unused	unused	unused
63 ITEM_TYPE_INSTRUMENT	level of spell	max charges	charges left	spell number	unused
64 ITEM_TYPE_HIDE	unused	unused	unused	mob vnum	race number
65 ITEM_TYPE_CART	capacity	container flags	key vnum	unused	unused
66 ITEM_TYPE_COMPONENT	number of uses for the component and amount of herb	unused	herb type	unused	unused

Scroll Notes

The level of the spell determines the cost. For scrolls that are sold make the spell level high. For scrolls that are found make the spell level low.

Symbol Notes

Builders are not to set any of their objects as type symbol. Type symbol objects have already been set up in the game.

Corpse Notes

Builders should never use the ITEM_TYPE_CORPSE_PC type in objects.

Portal Notes

Builders are not to use ITEM_TYPE_PORTAL in objects. It is used by hard code in the gate spell.

Unused Types Notes

The following types can be found on the FKBIT.LST but are no longer used by the game. Builders should not use them at all. They may eventually be replaced with new types.

```
ITEM_TYPE_FIREWEAPON 6
ITEM_TYPE_WORN 11
ITEM_TYPE_OLDTRAP 14
ITEM_TYPE_NOTE 16
ITEM_TYPE_BOAT 22
ITEM_TYPE_HERB_CON 31
ITEM_TYPE_HERB 32 - Merged with ITEM_TYPE_COMPONENT
ITEM_TYPE_INCENSE 33
ITEM_TYPE_SWITCH 36
ITEM_TYPE_PULLCHAIN 38
ITEM_TYPE_DIAL 40
ITEM_TYPE_RUNE 41
ITEM_TYPE_RUNEPOUCH 42
ITEM_TYPE_MATCH 43
ITEM_TYPE_TINDER 48
ITEM_TYPE_LOCKPICK 49
ITEM_TYPE_SPIKE 50
ITEM_TYPE_DISEASE 51
ITEM_TYPE_OIL 52
ITEM_TYPE_FUEL 53
ITEM_TYPE_SHORT_BOW 54
ITEM_TYPE_LONG_BOW 55
ITEM_TYPE_CROSSBOW 56
```

ITEM FLAGS

These extra flags describe more attributes of an object.

FLAG_GLOW	Item description has (Glowing), and item provides some light
FLAG_HUM	Item description has (Humming)
FLAG_DARK	This flag has no affect
FLAG_LOYAL	Weapon does not drop on floor if disarmed
FLAG_EVIL	Item has an evil aura
FLAG_INVIS	Item is invisible
FLAG_MAGIC	Item object has affects or program.
FLAG_NODROP	PC cannot drop object. Item is cursed
FLAG_RESIZE	Armour will resize when worn
FLAG_ANTI_LAWFUL	Item zaps good chars
FLAG_ANTI_CHAOTIC	Item zaps evil chars
FLAG_ANTI_UNCONCERNED	Item zaps neutral chars
FLAG_NOREMOVE	Item cannot be removed. Item is cursed.
FLAG_INVENTORY	Item cannot be put into containers and is more resistant to damage.
FLAG_ANTI_WIZARD	Item cannot be used by wizards
FLAG_ANTI_ROGUE	Item cannot be used by rogues
FLAG_ANTI_WARRIOR	Item cannot be used by warriors
FLAG_ANTI_PRIEST	Item cannot be used by priests

FLAG_NOSCRY	Item cannot be scryed for with spells.
FLAG_SHOPKEEPER	Used in hard code. Not for use by builders.
FLAG_METAL	No longer in use.
FLAG_CONCEALED	Only used on holy symbols with the conceal command
FLAG_DONATION	Do not use.
FLAG_POISONED	1/4 more damage
FLAG_COVERING	For containers "look under"
FLAG_DEATHROT	Item disappears from corpse when the PC or moble dies
FLAG_PROTOTYPE	Used in OLC. Not to be used for offline building.
FLAG_BURIED	Item is underground
FLAG_PERMANENT	Item stays on the PC through death.
FLAG_TRANSPARENT	Items worn under this layer can be seen. Used for cloaks etc.
FLAG_UNIQUE	Prevents the PC from wearing more than one of a specific object.

WEAR LOCATIONS

Some locations can be layered. See the lesson on layering for more information on how to layer. The table below lists the possible wear locations and if they can be layered or not. Note that we do not have a wield or shield location anymore. Shields and weapons should be wear hold. The game automatically determines if a weapon should be worn in both hands or not, depending on the size of the PC and the size of the weapon. For old builders please note that the ITEM part of the wear location flag was replaced with CAN.

CAN_WEAR_TAKE	This allows the item to be picked up by the PC.
CAN_WEAR_FINGER	There are two finger locations and they are not layerable.
CAN_WEAR_NECK	There are two neck locations and they are not layerable.
CAN_WEAR_BODY	There is one body location and it is layerable.
CAN_WEAR_HEAD	There is one head location and it is layerable.
CAN_WEAR_LEGS	There is one legs location and it is layerable.
CAN_WEAR_FEET	There is one feet location and it is layerable.
CAN_WEAR_HANDS	There is one hands location and it is layerable.
CAN_WEAR_ARMS	There is one arms location and it is layerable.
CAN_WEAR_WAIST	There is one waist location and it is not layerable.
CAN_WEAR_BELT	There is one belt location and it is layerable.
CAN_WEAR_WRIST	There are two wrist location and they are not layerable.
CAN_WEAR_HOLD	There are two hold locations and they are not layerable.
CAN_WEAR_BOTH_HANDS	There is one both hands location and it is not layerable.
CAN_WEAR_EARS	There is one ears location and it is not layerable.
CAN_WEAR_FACE	There is one face location and it is not layerable.

CAN_WEAR_FLOATING	There is one floating location and it is not layerable.
CAN_WEAR_SYMBOL	There is one feet location and it is layerable. (This one is not to be used in normal areas. It is for god symbols only).
CAN_WEAR_SADDLE	There is one saddle location and it is layerable. This can only be used by mobiles and PC's of a certain body type.
CAN_WEAR_ARMOR	This wear location is only for mobiles, set in hard code. It is not for use by builders.

OBJECT LAYERS

Various wear locations can be layered. That means that the character can wear up to three layers in the one location. Not all locations can be layered. For a list of locations that can be layered, refer to the locations listing. To set the layer of the object refer to the table in the Item Types Listing to see which value is reserved for the items layers.

LAYER_ALL	10	Nothing else can be worn with the item in the same locations.
LAYER_UNDER	11	Worn under armour. Such as padding and under clothes.
LAYER_ARMOR	12	Actual armor. It can have something worn under and over it.
LAYER_OVER	13	Can be worn over armour. Like robes.

Values CANNOT any combination of the above. You can only choose one of the above layers.

When choosing a material for an item at a certain layer level, make sure it makes sense. For instance layer under items will tend to be made of cloth, and not steel or leather. Layer over items will also tend to be cloth. Because of game balance, layer over and layer under items that are made of material stronger than cloth will be rare. Anything else will only be allowed to be made with area administrator approval.

OBJECT APPLIES

An 'A' section ('apply') contains an apply type and an apply value. When a character uses this object as equipment (holds, wields, or wears it), then the value of 'apply-value' is added to the character attribute identified by 'apply-type'. If the value put in is a negative it will deduct from that attribute. Older builders please note that several apply flags are now defunct and been replaced. Some of the old ones have been removed from this list altogether as they are no longer on the current bit list. Only those that are still on the bit list are on this table. There are also many new ones.

BIT VECTOR and APPLY	COMMENT
1 A APPLY_STR	Adds or takes away strength.
2 A APPLY_DEX	Adds or takes away from dexterity
3 A APPLY_INT	Adds or takes away from intelligence
4 A APPLY_WIS	Adds or takes away from wisdom
5 A APPLY_CON	Adds or takes away from constitution
6 A APPLY_SEX	Changes the sex of the PC by the value
7 A APPLY_CLASS	Do not use
8 A APPLY_LEVEL	Do not use
9 A APPLY_AGE	Do not use
10 A APPLY_HEIGHT	Adds to or takes away from the height of the character
11 A APPLY_WEIGHT	Adds to or takes away from the characters weight. (Not the weight carried)
12 A APPLY_MANA	Adds to or takes away from the character's total mana
13 A APPLY_HIT	Adds to or takes away from the total hitpoints of the character
14 A APPLY_MOVE	Adds to or takes away from the total stamina/move of the character

15 A APPLY_VALUE	Adds to or takes value from an object. This is measured in Copper.
16 A APPLY_EXP	Do not use
17 A APPLY_AC	Affects the character/s armor class. Negative value improves armour class, a postitive value degrades armour class
18 A APPLY_HITROLL	Adds or takes away hitroll to/from a weapon
19 A APPLY_DAMROLL	Adds or takes away dammroll from a weapon
20 A APPLY_RANGE	Allows the character to shoot or throw further or less, by the number of rooms specified
21 A APPLY_BOWS	Adds or takes away from the characters bow skill
22 A APPLY_SAP	Adds or takes away from the characters sap skill
23 A APPLY_BRAWLING	Adds or takes away from the characters brawling skill
24 A APPLY_APPRAISE	Adds or takes away from the characters appraise skill
25 A APPLY_CHA	Adds to or takes away from a PC's charisma
26 A APPLY_AFFECT	Used to apply AFF_ flags. Character remains affected while the object is worn.
27 A APPLY_RESISTANT	No longer in use. Do not use.
28 A APPLY_IMMUNE	No longer in use. Do not use.
29 A APPLY_SUSCEPTIBLE	No longer in use. Do not use.
30 A APPLY_WEAPONSPELL	Casts a spell when hitting use SPELL_ 100 % of the time.
31 A APPLY_LCK	Adds to or takes away from luck
32 A APPLY_BACKSTAB	Adds to or takes away from the backstab skill
33 A APPLY_PICK	Adds to or takes away from the pick locks skill

34 A APPLY_TRACK	Adds to or takes away from the track skill
35 A APPLY_STEAL	Adds to or takes away from the steal skill
36 A APPLY_SNEAK	Adds to or takes away from the sneak skill
37 A APPLY_HIDE	Adds to or takes away from the hide skill
38 A APPLY_PALM	Not coded. Do not use.
39 A APPLY_DETRAP	Adds to or takes away from the detrap skill
40 A APPLY_DODGE	Adds to or takes away from the dodge skill
41 A APPLY_PEEK	Adds to or takes away from the peek skill
42 A APPLY_SCAN	No longer used
43 A APPLY_GOUGE	Adds to or takes away from the gouge skill
44 A APPLY_SEARCH	Adds to or takes away from the search skill
45 A APPLY_MOUNT	Adds to or takes away from the mount skill
46 A APPLY_DISARM	Adds to or takes away from the disarm skill
47 A APPLY_KICK	Adds to or takes away from the kick skill
48 A APPLY_PARRY	Adds to or takes away from the parry skill
49 A APPLY_BASH	Adds to or takes away from the bash skill
50 A APPLY_STUN	Adds to or takes away from the stun skill
51 A APPLY_PUNCH	Adds to or takes away from the punch skill
52A APPLY_CLIMB	Adds to or takes away from the climb skill
53 A APPLY_GRIP	Adds to or takes away from the grip skill
54 A APPLY_SCRIBE	Adds to or takes away from the scribe skill

55 A APPLY_BREW	Adds to or takes away from the brew potions skill
56 A APPLY_WEARSPELL	Used to apply SPELL_ spell affects. Spell is applied to wearer when object is worn, and will wear off like a normal spell.
57 A APPLY_REMOVESPELL	When object is removed, the SPELL_ affects the character.
58 A APPLY_EMOTION	Adds to or takes away from a PC's emotional state
59 A APPLY_MENTALSTATE	Adds to or takes away from a PC's mental state
60 A APPLY_STRIPSN	Use SPELL_here. Wearing of object forces spell to wear-off.
61 A APPLY_REMOVE	Use AFF_flags here. Removes the affect upon wearing the object.
62 A APPLY_DIG	Adds to or takes away from a PC's dig skill
63 A APPLY_FULL	Affects the hours until the PC is hungry
64 A APPLY_THIRST	Affects the hours until the PC is thirsty
65 A APPLY_DRUNK	Affects the hours until the PC is sober
66 A APPLY_BLOOD	Do not use.
67 A APPLY_HAGGLE	Increases or decreases the characters haggle skill.
68 A APPLY_OBJWEIGHT	Increases or decreases the weight of the object.
69 A APPLY_RESIST_MAGIC	Wearing of the object affects the characters resistance to magic
70 A APPLY_RESIST_FIRE	Wearing of the object affects the characters resistance to fire
71 A APPLY_RESIST_COLD	Wearing of the object affects the characters resistance to cold
72 A APPLY_RESIST_ELECTRICITY	Wearing of the object affects the characters resistance to electricity
73 A APPLY_RESIST_ENERGY	Wearing of the object affects the characters resistance to energy
74 A APPLY_RESIST_ACID	Wearing of the object affects the characters resistance to acid

75 A APPLY_RESIST_POISON	Wearing of the object affects the characters resistance to poison
76 A APPLY_RESIST_DRAIN	Wearing of the object affects the characters resistance to drain
77 A APPLY_RESIST_HOLD	Wearing of the object affects the characters resistance to hold spells
78 A APPLY_RESIST_PHYSICAL	Wearing of the object affects the characters resistance to physical attacks
79 A APPLY_RESIST_HEALING	Wearing of the object affects the characters resistance to healing
80 A APPLY_RESIST_MIND	Wearing of the object affects the characters resistance to mind spells and attacks
81 A APPLY_RESIST_BASH	Wearing of the object affects the characters resistance to bash
82 A APPLY_RESIST_PIERCE	Wearing of the object affects the characters resistance to piercing weapons
83 A APPLY_RESIST_SLASH	Wearing of the object affects the characters resistance to slashing weapons
84 A APPLY_RESIST_NONMAGIC	Wearing of the object affects the characters resistance to non magical attacks
85 A APPLY_MAGIC	Wearing of the object increases magic damage
86 A APPLY_FIRE	Wearing of the object increases fire damage
87 A APPLY_COLD	Wearing of the object increases cold damage
88 A APPLY_ELECTRICITY	Wearing of the object increases electrical damage
89 A APPLY_ENERGY	Wearing of the object increases energy damage
90 A APPLY_ACID	Wearing of the object increases acid damage
91 A APPLY_POISON	Wearing of the object increases poison damage
92 A APPLY_DRAIN	Wearing of the object increases drain damage

93 A APPLY_HEALING	Wearing of the object increases healing
94 A APPLY_PHYSICAL	Wearing of the object increases physical damage
95 A APPLY_MIND	Wearing of the object increases mind damage
96 A APPLY_BLUDGEON	Wearing of the object increases bludgeon damage
97 A APPLY_PIERCE	Wearing of the object increases piercing damage
98 A APPLY_SLASH	Wearing of the object increases slashing damage
99 A APPLY_WEAPONSPELL_ONE	Casts a spell when hitting use SPELL_ 10 % of the time.
100 A APPLY_WEAPONSPELL_TWO	Casts a spell when hitting use SPELL_ 25 % of the time.
101 A APPLY_WEAPONSPELL_FIVE	Casts a spell when hitting use SPELL_ 50 % of the time.
102 APPLY_DUAL_WIELD	Increases or decreases the dual wield skill
103 APPLY_DISGUISE	Increases or decreases the disguise skill
104 APPLY_LEVEL_ONE_SPELL_SLOTS	Increases or decreases the number of level 1 spell slots
105 APPLY_LEVEL_TWO_SPELL_SLOTS	Increases or decreases the number of level 2 spell slots
106 APPLY_LEVEL_THREE_SPELL_SLOTS	Increases or decreases the number of level 3 spell slots
107 APPLY_LEVEL_FOUR_SPELL_SLOTS	Increases or decreases the number of level 4 spell slots
108 APPLY_LEVEL_FIVE_SPELL_SLOTS	Increases or decreases the number of level 5 spell slots
109 APPLY_LEVEL_SIX_SPELL_SLOTS	Increases or decreases the number of level 6 spell slots
110 APPLY_LEVEL_SEVEN_SPELL_SLOTS	Increases or decreases the number of level 7 spell slots
111 APPLY_LEVEL_EIGHT_SPELL_SLOTS	Increases or decreases the number of level 8 spell slots

112 APPLY_LEVEL_NINE_SPELL_SLOTS	Increases or decreases the number of level 9 spell slots
113 APPLY_SN	Used in spell definitions only, to cast a spell
114 APPLY_WEATHER	Used in spell definitions only, to modify the weather
115 APPLY_EXHAUSTION_MENTAL_STATE	Modifies the exhaustion mental state of the wearer (100 for dead tired, 0 for normal)
116 APPLY_SANITY_MENTAL_STATE	Modifies the sanity mental state of the wearer (100 for mad, 0 for normal)
117 APPLY_STUN_TIMER	Used in spell definitions only
118 APPLY_HELD_TIMER	Used in spell definitions only
119 APPLY_VALUE_ZERO	Used in spell definitions only, to modify an object's value0
120 APPLY_VALUE_ONE	Used in spell definitions only, to modify an object's value1
121 APPLY_VALUE_TWO	Used in spell definitions only, to modify an object's value2
122 APPLY_VALUE_THREE	Used in spell definitions only, to modify an object's value3
123 APPLY_VALUE_FOUR	Used in spell definitions only, to modify an object's value4
124 APPLY_VALUE_FIVE	Used in spell definitions only, to modify an object's value5
125 APPLY_SET_VALUE_ZERO	Used in spell definitions only, to change an object's value0
126 APPLY_SET_VALUE_ONE	Used in spell definitions only, to change an object's value1
127 APPLY_SET_VALUE_TWO	Used in spell definitions only, to change an object's value2
128 APPLY_SET_VALUE_THREE	Used in spell definitions only, to change an object's value3
129 APPLY_SET_VALUE_FOUR	Used in spell definitions only, to change an object's value4
130 APPLY_SET_VALUE_FIVE	Used in spell definitions only, to change an object's value5
131 APPLY_CREATE_MINION	Used in spell_definitions only, to create a minion

132 APPLY_CREATE_OBJECT	Used in spell_definitions only, to create an object
133 APPLY_CREATE_MOBILE	Used in spell_definitions only, to create a mobile
134 APPLY_CALL_PROG	Used in spell_definitions only, to run an area program

OBJECT AFFECT FLAGS

Affects can be applied to objects, and when worn these affects affect the wearer.

AFF_BLIND	Wearer of object is affected by blindness
AFF_INVIS	Wearer of object is invisible
AFF_DETECT_EVIL	Wearer of object can detect evil
AFF_DETECT_INVIS	Wearer of object can detect magic
AFF_DETECT_MAGIC	Wearer of object can detect magic
AFF_DETECT_HIDDEN	Wearer of object can detect hidden creatures
AFF_DETECT_BURIED	Wearer of object can detect buried
AFF_SANCTUARY	Wearer of object is protected by sanctuary
AFF_FAERIE_FIRE	Wearer of object is affected by faerie fire
AFF_INFRARED	Wearer of object has infravision
AFF_CURSE	Wearer of object is cursed
AFF_POISON	Wearer of object is poisoned
AFF_PROTECT	Wearer of object is protected by protection
AFF_PARALYSIS	Wearer of object is paralyzed
AFF_SNEAK	Wearer of object is sneaking
AFF_HIDE	Wearer of object is hiding
AFF_SLEEP	Wearer of object is affected by sleep spell
AFF_CHARM	Wearer of object is charmed

AFF_FLYING	Wearer of object is is flying
AFF_PASS_DOOR	Wearer of object can pass door
AFF_FLOATING	Wearer of object is floating
AFF_TRUE_SIGHT	Wearer of object is affected by truesight
AFF_FIRESHIELD	Wearer of object is affected by fireshield
AFF_SHOCKSHIELD	Wearer of object is affected by shockshield
AFF_ICESHIELD	Wearer of object is affected by iceshield
AFF_BERSERK	Object affects wearer with berserk
AFF_WATER_BREATHING	Wearer of object is affected by water breathing
AFF_GUARDIAN	Object guards the wearer while they sleep

OBJECT PROGRAMS

Like mobiles, objects can also have programs. Refer to the mobprogs listing to see what kind of programs work on objects. An object can have more than one program on it. The sample below has a program that triggers when the item is worn. There are more program samples in the Mud Programs section of the builders lessons.

```
#QQ00
ring pink ice~
{D0}a pink ice ring~
{D0}A pink ice ring is lying here.~
~
ITEM_TYPE_TREASURE
FLAG_GLOW|FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_FINGER
QUALITY_AVERAGE MATERIAL_GOLD CONDITION_PERFECT SIZE_MEDIUM
0 0 0 0 LAYER_ALL 0
E
pink ice ring~
{B0}It is golden ring with {D0}a pretty pink gem in it.
~
>wear_prog 100~
mpechoat $n The ring feels cold on your finger
~
|
```

OBJECT QUALITY, CONDITION AND SIZE

OBJECT QUALITY

Objects of high quality are considered masterwork items. These items offer extra AC for armour, and hitroll for weapons. These items should be considered a quest reward, or offered to select groups of characters. For instance, in a temple the higher quality armour may only be offered to members of the faith. This allows for characters to wear in character armour consistent with what is worn by the faith of a high quality if they so wish.

QUALITY_WORTHLESS	0
QUALITY_INFERIOR	1
QUALITY_LOW	2
QUALITY_AVERAGE	3
QUALITY_HIGH	4
QUALITY_SUPERIOR	5
QUALITY_OUTSTANDING	6

OBJECT CONDITION

In general items that are sold in shops should be of perfect condtion, unless there is a reason for the shop-keeper to be selling items in shoddy condition.

COND_TERRIBLE	0
COND_AWFUL	1
COND_VERY_BAD	2
COND_BAD	3
COND_USABLE	4
COND_GOOD	5
COND_VERY_GOOD	6
COND_SUPERB	7
COND_PERFECT	8

OBJECT SIZE

In the case of items that are worn on the wrist that are size tiny, when a larger creature tries to wear them, they will only fit on their finger. The same with size tiny waist items, they will fit on the wrist of a larger creature. Weapons in general have to be of a certain size. Refer to the weapon sizes listing for more information.

SIZE_TINY	0
SIZE_SMALL	1
SIZE_MEDIUM	2
SIZE_LARGE	3
SIZE_HUGE	4
SIZE_GIANT	5

OBJECT MATERIALS

Only one material type can be used on each object. The materials have different levels of hardness and durablility. Make sure to make your object the material it is supposed to be so it wears as expected.

MATERIAL_UNKNOWN	0	MATERIAL_WOOD	1
MATERIAL_OAK	2	MATERIAL_YEW	3
MATERIAL_EBONY	4	MATERIAL_HARDWOOD	5
MATERIAL_ICE	6	MATERIAL_SOFTWOOD	7
MATERIAL_FLESH	8	MATERIAL_SILK	9
MATERIAL_WOOL	10	MATERIAL_CLOTH	11
MATERIAL_FUR	12	MATERIAL_WATER	13
MATERIAL_METAL	14	MATERIAL_SILVER	15
MATERIAL_GOLD	16	MATERIAL_STEEL	17
MATERIAL_LEAD	18	MATERIAL_BRONZE	19
MATERIAL_COPPER	20	MATERIAL_PLATINUM	21
MATERIAL_TITANIUM	22	MATERIAL_ALUMINIUM	23
MATERIAL_TIN	24	MATERIAL_IRON	25
MATERIAL_BRASS	26	MATERIAL_DIAMOND	27
MATERIAL_PEARL	28	MATERIAL_GEM	29
MATERIAL_RUBY	30	MATERIAL_OBSIDIAN	31
MATERIAL_IVORY	32	MATERIAL_MITHRIL	33

MATERIAL_ADAMANTIUM	34	MATERIAL_ENERGY	35
MATERIAL_GLASS	36	MATERIAL_PAPER	37
MATERIAL_MARBLE	38	MATERIAL_PLANT	39
MATERIAL_STONE	40	MATERIAL_HIDE	41
MATERIAL_BONE	42	MATERIAL_POWDER	43
MATERIAL_LEATHER	44	MATERIAL_OIL	45
MATERIAL_ELVEN	46	MATERIAL_ELECTRUM	47
MATERIAL_EMERALD	48	MATERIAL_SAPPHIRE	49

ARMOUR TYPES

While we expect builders to spell armour with a U in the game to keep with the medieval theme, the code was orginally written by those who do not use correct english (yes the writer of these lessons is a biased Australian). Armour can only have one type. So when setting your armor_type you need to drop the "U". Armor type is set in Value3 of an object that is type armor.

ARMOR_TYPE_BANDED	0
ARMOR_TYPE_BRIGANDINE	1
ARMOR_TYPE_CHAIN_MAIL	2
ARMOR_TYPE_FIELD_PLATE_ARMOR	3
ARMOR_TYPE_FULL_PLATE	4
ARMOR_TYPE_HIDE	5
ARMOR_TYPE_LEATHER	6
ARMOR_TYPE_PADDED	7
ARMOR_TYPE_PLATE_MAIL	8
ARMOR_TYPE_RING_MAIL	9
ARMOR_TYPE_SCALE_MAIL	10
ARMOR_TYPE_SPLINT_MAIL	11
ARMOR_TYPE_STUDDED_LEATHER	12
ARMOR_TYPE_CLOTH	13
ARMOR_TYPE_SHIELD	14

Notes

Make sure that when you make a shield you set the type to shield, not if it is leather or splint or the like. If the shield is to

work with skills like shield block etc, it needs to be set to be type shield.

Armour type plays a big factor in spell and skill failure due to the armour a character wears.

WEAPON TYPES

The weapon type is set in Value3 on an object that is type weapon. A weapon can be of only one type.

WEAPON_TYPE_BASTARD_SWORD	0
WEAPON_TYPE_BATTLE_AXE	1
WEAPON_TYPE_BLACKJACK	2
WEAPON_TYPE_BOAR_SPEAR	3
* WEAPON_TYPE_BOLA	4
* WEAPON_TYPE_BOOMERANG	5
WEAPON_TYPE_BROAD_SWORD	6
WEAPON_TYPE_CAT_O_NINE_TAILS	7
WEAPON_TYPE_FULLBLADE	8
WEAPON_TYPE_CLUB	9
* WEAPON_TYPE_COMPOSITE_BOW	10
WEAPON_TYPE_GREATCLUB	11
WEAPON_TYPE_CUTLASS	12
* WEAPON_TYPE_DAGGER	13
* WEAPON_TYPE_DARTS	14
* WEAPON_TYPE_DIRK	15
WEAPON_TYPE_FALCHION	16
WEAPON_TYPE_FLAIL	17

* WEAPON_TYPE_FISHING_NET	18
WEAPON_TYPE_FOIL	19
* WEAPON_TYPE_GLADIATOR_NET	20
WEAPON_TYPE_HALBERD	21
* WEAPON_TYPE_HANDAXE	22
* WEAPON_TYPE_HEAVY_CROSSBOW	23
* WEAPON_TYPE_HARPOON	24
* WEAPON_TYPE_JAVELIN	25
WEAPON_TYPE_JO	26
WEAPON_TYPE_KATANA	27
WEAPON_TYPE_LANCE	28
WEAPON_TYPE_LASSO	29
* WEAPON_TYPE_LIGHT_CROSSBOW	30
* WEAPON_TYPE_LONG_BOW	31
WEAPON_TYPE_LONG_SWORD	32
WEAPON_TYPE_MACE	33
WEAPON_TYPE_MAIN_GAUCHE	34
WEAPON_TYPE_MORNING_STAR	35
WEAPON_TYPE_NAGINATA	36
WEAPON_TYPE_NUNCHAKU	37
WEAPON_TYPE_PICK	38

* WEAPON_TYPE_PILUM	39
WEAPON_TYPE_QUARTERSTAFF	40
WEAPON_TYPE_RAPIER	41
WEAPON_TYPE_SABRE	42
WEAPON_TYPE_SAI	43
WEAPON_TYPE_SCIMITAR	44
* WEAPON_TYPE_SHORT_BOW	45
WEAPON_TYPE_SHORT_SWORD	46
* WEAPON_TYPE_SHURIKEN	47
* WEAPON_TYPE_SLING	48
WEAPON_TYPE_SPEAR	49
* WEAPON_TYPE_STAFF_SLING	50
WEAPON_TYPE_TOMAHAWK	51
WEAPON_TYPE_TONFA	52
WEAPON_TYPE_TRIDENT	53
WEAPON_TYPE_GREATSWORD	54
* WEAPON_TYPE_LIGHT_HAMMER	55
WEAPON_TYPE_WARHAMMER	56
WEAPON_TYPE_WHIP	57
WEAPON_TYPE_KNIFE	58

WEAPON_TYPE_SICKLE	59
WEAPON_TYPE_SCYTHE	60
* WEAPON_TYPE_BOULDER	61

Notes

* Denotes that this weapon requires a projectile of the same type to shoot.

* Denotes that this weapon can be thrown.

Older builders should be aware that in 2003 the coders changed alot of the weapon types and added new ones. The listing above is current as at the date on the bottom of this page.

WEAPON SIZES

The following list of weapons has the standard size for that weapon. Sometimes the weapon can vary from the norm, if there is a good reason. But for the most part you should make sure weapons of the type you make, are of the standard size for that type.

bastard sword	SIZE_MEDIUM
battle axe	SIZE_MEDIUM
blackjack	SIZE_SMALL
halfspear	SIZE_MEDIUM
bola	SIZE_SMALL
boomerang	SIZE_SMALL
broad sword	SIZE_MEDIUM
cat-o'-nine tails	SIZE_MEDIUM
fullblade	SIZE_LARGE
club	SIZE_MEDIUM
composite bow	SIZE_LARGE
greatclub	SIZE_LARGE
cutlass	SIZE_MEDIUM
dagger	SIZE_TINY
dart	SIZE_SMALL
dirk	SIZE_SMALL
falchion	SIZE_MEDIUM

flail	SIZE_MEDIUM
fishing net	SIZE_MEDIUM
foil	SIZE_MEDIUM
gladiator net	SIZE_MEDIUM
halberd	SIZE_LARGE
handaxe	SIZE_SMALL
heavy crossbow	SIZE_MEDIUM
harpoon	SIZE_LARGE
javelin	SIZE_MEDIUM
jo	SIZE_MEDIUM
katana	SIZE_MEDIUM
lance	SIZE_MEDIUM
lasso	SIZE_MEDIUM
light crossbow	SIZE_SMALL
long bow	SIZE_LARGE
long sword	SIZE_MEDIUM
mace	SIZE_MEDIUM
main gauche	SIZE_MEDIUM
morning star	SIZE_MEDIUM
naginata	SIZE_LARGE
nunchaku	SIZE_SMALL

pick	SIZE_MEDIUM
pilum	SIZE_MEDIUM
quarterstaff	SIZE_LARGE
rapier	SIZE_MEDIUM
sabre	SIZE_MEDIUM
sai	SIZE_SMALL
scimitar	SIZE_MEDIUM
short bow	SIZE_MEDIUM
short sword	SIZE_SMALL
shuriken	SIZE_TINY
sling	SIZE_SMALL
longspear	SIZE_LARGE
staff sling	SIZE_MEDIUM
tomahawk	SIZE_SMALL
tonfa	SIZE_SMALL
trident	SIZE_MEDIUM
greatsword	SIZE_LARGE
light hammer	SIZE_SMALL
warhammer	SIZE_MEDIUM
whip	SIZE_SMALL

lance	SIZE_MEDIUM
lasso	SIZE_MEDIUM
light crossbow	SIZE_SMALL
long bow	SIZE_LARGE
long sword	SIZE_MEDIUM
mace	SIZE_MEDIUM
main gauche	SIZE_MEDIUM
morning star	SIZE_MEDIUM
naginata	SIZE_LARGE
nunchaku	SIZE_SMALL
pick	SIZE_MEDIUM
pilum	SIZE_MEDIUM
quarterstaff	SIZE_LARGE
rapier	SIZE_MEDIUM
sabre	SIZE_MEDIUM
sai	SIZE_SMALL
scimitar	SIZE_MEDIUM
short bow	SIZE_MEDIUM
short sword	SIZE_SMALL
shuriken	SIZE_TINY
sling	SIZE_SMALL

longspear	SIZE_LARGE
staff sling	SIZE_MEDIUM
tomahawk	SIZE_SMALL
tonfa	SIZE_SMALL
trident	SIZE_MEDIUM
greatsword	SIZE_LARGE
light hammer	SIZE_SMALL
warhammer	SIZE_MEDIUM
whip	SIZE_SMALL

WEAPON FLAGS

Weapon flags are magical properties of weapons, that are specific to weapons, unlike the A APPLY's that can apply to objects in general. Make sure to add a magic flag to a weapon that has any of these flags.

```
#QQ27
black handled broadsword sword~
{80}a black handled broadsword~
{80}A black handled broadsword lies rusting on the ground here. ~
~
ITEM_TYPE_WEAPON
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD|CAN_WEAR_BELT
QUALITY_AVERAGE MATERIAL_METAL COND_PERFECT SIZE_MEDIUM
0 WFLAG_BANE RACE_DRAGON WEAPON_TYPE_BROAD_SWORD 0 0
E
black handled broadsword sword~
It is a broad sword with a black handle.
~
```

The weapon above will do extra damage against dragons.

WFLAG_ARROW_DEFLECTION	Acts as if the PC has the arrow deflection feat while wielding a weapon
WFLAG_REFLECTIVE	Weapon will reflect spells back at the caster
WFLAG_BANE	Increased modifiers against a specific race
WFLAG_DISRUPTION	Critical hits have a chance of destroying undead - doing 2-4 times normal damage
WFLAG_RETURNING	Thrown weapons or projectiles to the PC's inventory after use
WFLAG_SPEED	Increased number of attacks
WFLAG_THROWING	A normally unthrown weapon has the ability to be thrown
WFLAG_WOUNDING	Causes the victim to bleed

WFLAG_VENOMOUS	The weapon is always poisonous, it does not need poison to be reapplied
WFLAG_SMITING	Critical hit gives the weapon a chance of destroying constructs
WFLAG_VAMPIRIC	The weapon will give hitpoints to the wielder and take them from the victim
WFLAG_KEEN	Increased critical threat range
WFLAG_VORPAL	Chance of severing limbs
WFLAG_BACKSTABBING	Increased damage from backstabs
WFLAG_HOLY	Increased damage to undead and outsiders
WFLAG_DEFENDER	Increased AC

Value2 on a weapon has the modifier to the flag. For instance, if you apply WFLAG_BANE, you will need to enter the race number of the race that the Bane is against.

WEAPON SPELLS

Weapons can be affected by a spell. There are 4 weapon spell flags. Each one is different in the amount of times that the spell will trigger on the weapons.

APPLY_WEAPONSPELL	The spell works when the weapon hits 100% of the time.
APPLY_WEAPONSPELL_ONE	The spell works when the weapon hits 10% of the time.
APPLY_WEAPONSPELL_TWO	The spell works when the weapon hits 25% of the time.
APPLY_WEAPONSPELL_FIVE	The spell works when the weapon hits 50% of the time.

The possible spells that could be included on a weapon are listed in the table below:

SPELL_NONE -1
SPELL_ACETUM_PRIMUS 1
SPELL_ACID_ARROW 2
SPELL_ACID_BLAST 3
SPELL_ACID_BREATH 4
SPELL_ALERTNESS 5
SPELL_ANIMATE_DEAD 6
SPELL_ANIMATE_OBJECT 7
SPELL_ANTIMAGIC_SHELL 8
SPELL_ARMOR 9
SPELL_ASTRAL_WALK 10

SPELL_BARKSKIN 11
SPELL_BLAZEBANE 12
SPELL_BLESS 13
SPELL_BLINDNESS 14
SPELL_BLOOD_OF_CYRIC 15
SPELL_BURNING_HANDS 16
SPELL_CALL_LIGHTNING 17
SPELL_CAUSE_CRITICAL 18
SPELL_CAUSE_LIGHT 19
SPELL_CAUSE_SERIOUS 20
SPELL_CLAIRVOYANCE 21
SPELL_CHANGE_SEX 22
SPELL_CHAIN_LIGHTNING 23
SPELL_CHARGED_BEACON 24
SPELL_CHARIOT_OF_THE_SUN 25
SPELL_CHARM_PERSON 26
SPELL_CHILL_TOUCH 27
SPELL_COLOR_SPRAY 28
SPELL_CONE_OF_COLD 29
SPELL_CONJURE_ELEMENTAL 30
SPELL_CONTINUAL_LIGHT 31

SPELL_CONTROL_WEATHER 32
SPELL_CREATE_FOOD 33
SPELL_CREATE_SPRING 34
SPELL_CREATE_SYMBOL 35
SPELL_CREATE_WATER 36
SPELL_CURE_BLINDNESS 37
SPELL_CURE_CRITICAL 38
SPELL_CURE_LIGHT 39
SPELL_CURE_POISON 40
SPELL_CURE_SERIOUS 41
SPELL_CURSE 42
SPELL_DETECT_BURIED 43
SPELL_DETECT_EVIL 44
SPELL_DETECT_HIDDEN 45
SPELL_DETECT_INVIS 46
SPELL_DETECT_MAGIC 47
SPELL_DETECT_POISON 48
SPELL_DISJUNCTION 49
SPELL_DISPEL_EVIL 51
SPELL_DISPEL_MAGIC 52
SPELL_DIVINITY 53

SPELL_DISINTEGRATE 54
SPELL_DRAGONSKIN 55
SPELL_DREAM 56
SPELL_EARTHQUAKE 57
SPELL_ENCHANT_WEAPON 58
SPELL_ENERGY_DRAIN 59
SPELL_FAERIE_FIRE 60
SPELL_FAERIE_FOG 61
SPELL_FARHEAL 62
SPELL_FATIGUE 63
SPELL_FEEBLEMIND 64
SPELL_FIND_FAMILIAR 65
SPELL_FIND_TRAPS 66
SPELL_FIREBALL 67
SPELL_FIRE_BREATH 68
SPELL_FLAME_ARROW 69
SPELL_FIRESHIELD 70
SPELL_FLAME_JAWS 71
SPELL_FLAMESTRIKE 72
SPELL_FLY 73

SPELL_FRIENDS 74
SPELL_FROST_BREATH 75
SPELL_FUMBLE 76
SPELL_GAS_BREATH 77
SPELL_GATE 78
SPELL_GOOD_FORTUNE 79
SPELL_HAND_OF_CHAOS 80
SPELL_HARM 81
SPELL_HEAL 82
SPELL_HOLY_SANCTITY 83
SPELL_ICESHIELD 84
SPELL_ICE_STORM 85
SPELL_IDENTIFY 86
SPELL_ILL_FORTUNE 87
SPELL_ILMATERS_BLESSING 88
SPELL_INFRAVISION 89
SPELL_INVIS 90
SPELL_KNOCK 91
SPELL_KNOW_ALIGNMENT 92
SPELL_LEVITATE 93
SPELL_LIGHTNING_BOLT 94

SPELL_LIGHTNING_BREATH 95
SPELL_LOCATE_OBJECT 96
SPELL_MAGIC_MIRROR 97
SPELL_MAGIC_MISSILE 98
SPELL_MAGNETIC_THRUST 99
SPELL_MASS_INVIS 100
SPELL_MIND_WRACK 101
SPELL_MIND_WRENCH 102
SPELL_MINOR_GLOBE 103
SPELL_MNEMONIC_ENHANCER 104
SPELL_MONSTER_SUMMON 105
SPELL_MOONBEAM 106
SPELL_NULL_SPHERE 107
SPELL_PASS_DOOR 108
SPELL_PHOENIX_CLAW 109
SPELL_PASS_PLANT 110
SPELL_POISON 111
SPELL_POLYMORPH 112
SPELL_POSSESS 113
SPELL_PRODUCE_FLAME 114
SPELL_PROTECTION 115

SPELL_QUANTUM_SPIKE 116

SPELL_RAINBOW_PATTERN 117

SPELL_RAZORBAIT 118

SPELL_RECHARGE 119

SPELL_REGENERATE 120

SPELL_RESIST_COLD 121

SPELL_RESIST_ELECTRICITY 122

SPELL_RESIST_FIRE 123

SPELL_REFRESH 124

SPELL_REMOVE_CURSE 125

SPELL_REMOVE_INVIS 126

SPELL_REMOVE_TRAP 127

SPELL_RESILIENCE 128

SPELL_RESTORATION 129

SPELL_RESTORE_MANA 130

SPELL_REVIVE 131

SPELL_SAGACITY 132

SPELL_SANCTUARY 133

SPELL_SCORCHING_SURGE 134

SPELL_SHADOW_WALK 135

SPELL_SHADOW_FIST 136
SPELL_SHADOW_FUNNEL 137
SPELL_SHADOW_DOOR 138
SPELL_SHIELD 139
SPELL_SHOCKING_GRASP 140
SPELL_SHOCKSHIELD 141
SPELL_SLEEP 142
SPELL_SLINK 143
SPELL_SPECTRAL_FIST 144
SPELL_SPECTRAL_HAND 145
SPELL_SPECTRAL_LIGHTNING 146
SPELL_SONIC_RESONANCE 147
SPELL_STRENGTH 148
SPELL_SUNRAY 149
SPELL_STONE_SKIN 150
SPELL_SULFUROUS_SPRAY 151
SPELL_SUMMON 152
SPELL_SWORDBAIT 153
SPELL_TELEPORT 154
SPELL_TOUCH_OF_JUSTICE 155
SPELL_TRANSPORT 156

SPELL_TROLLISH_VIGOR 157

SPELL_TRUE_SIGHT 158

SPELL_VALIANCE 159

SPELL_VAMPIRIC_TOUCH 160

SPELL_VENTRILOQUISM 161

SPELL_WARHORSE 162

SPELL_WATER_BREATHING 163

SPELL_WEAKEN 164

SPELL_WIND_WALK 165

SPELL_WINTER_MIST 166

SPELL_WITCH_LIGHT 167

SPELL_WORD_OF_RECALL 168

SPELL_WRAITHFORM 169

SPELL_WRATH_OF_DOMINUS 170

SPELL_WEB 171

SPELL_TURN_UNDEAD 172

SPELL_SENTRY_OF_HELM 173

SPELL_WATER_TO_WINE 174

SPELL_RAISE_DEAD 175

SPELL_RESURRECTION 176

SPELL_HOLD_PERSON 177

SPELL_SILENCE 178
SPELL_ENTANGLE 179
SPELL_COMPREHEND_LANGUAGE 180
SPELL_MIND_SHIELD 181
SPELL_STONE_WALK 182
SPELL_ENCHANT_ARMOR 183
SPELL_HEROISM 184
SPELL_SHADOW_CONJURATION 185
SPELL_MIRROR_IMAGE 186
SPELL_DELAYED_BLAST_FIREBALL 187
SPELL_MENDING 188
SPELL_NON_DETECTION 189
SPELL_FREEDOM 190
SPELL_CHARM_MONSTER 191
SPELL_HOLD_MONSTER 192
SPELL_CONTROL_UNDEAD 193
SPELL_ACIDSHIELD 194
SPELL_CREATE_OBJECT 195
SPELL_FEAR 196
SPELL_ETHEREAL_FLYER 197

SPELL_RESERVED_FOR_FUTURE 198
SPELL_PHANTASMAL_KILLER 199
SPELL_SPEAK_WITH_DEAD 200
SPELL_FLOATING_DISC 201

Below is a sample weapon that uses the weapon spell code. In general you should only use the lesser rates of success weapon spell applies.

```
#QQ17
large halberd gluttony~
{70}a large halberd~
{70}A large halberd lies on the ground here.~
~
ITEM_TYPE_WEAPON
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_SUPERIOR MATERIAL_STEEL COND_PERFECT SIZE_LARGE
0 0 0 WEAPON_TYPE_HALBERD 0 0
A APPLY_WEAPONSPELL_ONE SPELL_CREATE_FOOD
E
halberd large~
There is an unusual rune on the handle of the halberd.
~
```

MAGIC ITEM SPELLS

Various objects have spells on them. Here is a list of spells that can be placed on those objects. Such objects include, scrolls, potions, staves, wands, salves and weapons.

SPELL_NONE -1
SPELL_ACETUM_PRIMUS 1
SPELL_ACID_ARROW 2
SPELL_ACID_BLAST 3
SPELL_ACID_BREATH 4
SPELL_ALERTNESS 5
SPELL_ANIMATE_DEAD 6
SPELL_ANIMATE_OBJECT 7
SPELL_ANTIMAGIC_SHELL 8
SPELL_ARMOR 9
SPELL_ASTRAL_WALK 10
SPELL_BARKSKIN 11
SPELL_BLAZEBANE 12
SPELL_BLESS 13
SPELL_BLINDNESS 14
SPELL_BLOOD_OF_CYRIC 15
SPELL_BURNING_HANDS 16

SPELL_CALL_LIGHTNING 17
SPELL_CAUSE_CRITICAL 18
SPELL_CAUSE_LIGHT 19
SPELL_CAUSE_SERIOUS 20
SPELL_CLAIRVOYANCE 21
SPELL_CHANGE_SEX 22
SPELL_CHAIN_LIGHTNING 23
SPELL_CHARGED_BEACON 24
SPELL_CHARIOT_OF_THE_SUN 25
SPELL_CHARM_PERSON 26
SPELL_CHILL_TOUCH 27
SPELL_COLOR_SPRAY 28
SPELL_CONE_OF_COLD 29
SPELL_CONJURE_ELEMENTAL 30
SPELL_CONTINUAL_LIGHT 31
SPELL_CONTROL_WEATHER 32
SPELL_CREATE_FOOD 33
SPELL_CREATE_SPRING 34
SPELL_CREATE_SYMBOL 35
SPELL_CREATE_WATER 36
SPELL_CURE_BLINDNESS 37

SPELL_CURE_CRITICAL 38
SPELL_CURE_LIGHT 39
SPELL_CURE_POISON 40
SPELL_CURE_SERIOUS 41
SPELL_CURSE 42
SPELL_DETECT_BURIED 43
SPELL_DETECT_EVIL 44
SPELL_DETECT_HIDDEN 45
SPELL_DETECT_INVIS 46
SPELL_DETECT_MAGIC 47
SPELL_DETECT_POISON 48
SPELL_DISJUNCTION 49
SPELL_DISPEL_EVIL 51
SPELL_DISPEL_MAGIC 52
SPELL_DIVINITY 53
SPELL_DISINTEGRATE 54
SPELL_DRAGONSKIN 55
SPELL_DREAM 56
SPELL_EARTHQUAKE 57
SPELL_ENCHANT_WEAPON 58
SPELL_ENERGY_DRAIN 59

SPELL_FAERIE_FIRE 60
SPELL_FAERIE_FOG 61
SPELL_FARHEAL 62
SPELL_FATIGUE 63
SPELL_FEEBLEMIND 64
SPELL_FIND_FAMILIAR 65
SPELL_FIND_TRAPS 66
SPELL_FIREBALL 67
SPELL_FIRE_BREATH 68
SPELL_FLAME_ARROW 69
SPELL_FIRESHIELD 70
SPELL_FLAME_JAWS 71
SPELL_FLAMESTRIKE 72
SPELL_FLY 73
SPELL_FRIENDS 74
SPELL_FROST_BREATH 75
SPELL_FUMBLE 76
SPELL_GAS_BREATH 77
SPELL_GATE 78
SPELL_GOOD_FORTUNE 79

SPELL_HAND_OF_CHAOS 80
SPELL_HARM 81
SPELL_HEAL 82
SPELL_HOLY_SANCTITY 83
SPELL_ICESHIELD 84
SPELL_ICE_STORM 85
SPELL_IDENTIFY 86
SPELL_ILL_FORTUNE 87
SPELL_ILMATERS_BLESSING 88
SPELL_INFRAVISION 89
SPELL_INVIS 90
SPELL_KNOCK 91
SPELL_KNOW_ALIGNMENT 92
SPELL_LEVITATE 93
SPELL_LIGHTNING_BOLT 94
SPELL_LIGHTNING_BREATH 95
SPELL_LOCATE_OBJECT 96
SPELL_MAGIC_MIRROR 97
SPELL_MAGIC_MISSILE 98
SPELL_MAGNETIC_THRUST 99
SPELL_MASS_INVIS 100

SPELL_MIND_WRACK 101

SPELL_MIND_WRENCH 102

SPELL_MINOR_GLOBE 103

SPELL_MNEMONIC_ENHANCER 104

SPELL_MONSTER_SUMMON 105

SPELL_MOONBEAM 106

SPELL_NULL_SPHERE 107

SPELL_PASS_DOOR 108

SPELL_PHOENIX_CLAW 109

SPELL_PASS_PLANT 110

SPELL_POISON 111

SPELL_POLYMORPH 112

SPELL_POSSESS 113

SPELL_PRODUCE_FLAME 114

SPELL_PROTECTION 115

SPELL_QUANTUM_SPIKE 116

SPELL_RAINBOW_PATTERN 117

SPELL_RAZORBAIT 118

SPELL_RECHARGE 119

SPELL_REGENERATE 120

SPELL_RESIST_COLD 121

SPELL_RESIST_ELECTRICITY 122
SPELL_RESIST_FIRE 123
SPELL_REFRESH 124
SPELL_REMOVE_CURSE 125
SPELL_REMOVE_INVIS 126
SPELL_REMOVE_TRAP 127
SPELL_RESILIENCE 128
SPELL_RESTORATION 129
SPELL_RESTORE_MANA 130
SPELL_REVIVE 131
SPELL_SAGACITY 132
SPELL_SANCTUARY 133
SPELL_SCORCHING_SURGE 134
SPELL_SHADOW_WALK 135
SPELL_SHADOW_FIST 136
SPELL_SHADOW_FUNNEL 137
SPELL_SHADOW_DOOR 138
SPELL_SHIELD 139
SPELL_SHOCKING_GRASP 140
SPELL_SHOCKSHIELD 141

SPELL_SLEEP 142
SPELL_SLINK 143
SPELL_SPECTRAL_FIST 144
SPELL_SPECTRAL_HAND 145
SPELL_SPECTRAL_LIGHTNING 146
SPELL_SONIC_RESONANCE 147
SPELL_STRENGTH 148
SPELL_SUNRAY 149
SPELL_STONE_SKIN 150
SPELL_SULFUROUS_SPRAY 151
SPELL_SUMMON 152
SPELL_SWORDBAIT 153
SPELL_TELEPORT 154
SPELL_TOUCH_OF_JUSTICE 155
SPELL_TRANSPORT 156
SPELL_TROLLISH_VIGOR 157
SPELL_TRUE_SIGHT 158
SPELL_VALIANCE 159
SPELL_VAMPIRIC_TOUCH 160
SPELL_VENTRILOQUISM 161
SPELL_WARHORSE 162

SPELL_WATER_BREATHING 163

SPELL_WEAKEN 164

SPELL_WIND_WALK 165

SPELL_WINTER_MIST 166

SPELL_WITCH_LIGHT 167

SPELL_WORD_OF_RECALL 168

SPELL_WRAITHFORM 169

SPELL_WRATH_OF_DOMINUS 170

SPELL_WEB 171

SPELL_TURN_UNDEAD 172

SPELL_SENTRY_OF_HELM 173

SPELL_WATER_TO_WINE 174

SPELL_RAISE_DEAD 175

SPELL_RESURRECTION 176

SPELL_HOLD_PERSON 177

SPELL_SILENCE 178

SPELL_ENTANGLE 179

SPELL_COMPREHEND_LANGUAGE 180

SPELL_MIND_SHIELD 181

SPELL_STONE_WALK 182

SPELL_ENCHANT_ARMOR 183

SPELL_HEROISM 184
SPELL_SHADOW_CONJURATION 185
SPELL_MIRROR_IMAGE 186
SPELL_DELAYED_BLAST_FIREBALL 187
SPELL_MENDING 188
SPELL_NON_DETECTION 189
SPELL_FREEDOM 190
SPELL_CHARM_MONSTER 191
SPELL_HOLD_MONSTER 192
SPELL_CONTROL_UNDEAD 193
SPELL_ACIDSHIELD 194
SPELL_CREATE_OBJECT 195
SPELL_FEAR 196
SPELL_ETHEREAL_FLYER 197
SPELL_RESERVED_FOR_FUTURE 198
SPELL_PHANTASMAL_KILLER 199
SPELL_SPEAK_WITH_DEAD 200
SPELL_FLOATING_DISC 201

PIPE FLAGS

When the flags are not set, the pipe is defaulted to empty. In most instances when a pipe is made in area file it will be empty. When empty set the flag to 0.

PIPE_TAMPED
PIPE_LIT
PIPE_HOT
PIPE_DIRTY
PIPE_FILTHY
PIPE_GOINGOUT
PIPE_BURNT
PIPE_FULLOFASH

CONTAINER FLAGS

The container conditions can be combined. For instance a container can be all 4 possible flags. Use the bit number in the value field. If a container is locked and closed and closable, then you would add 1 and 4 and 8. This is equal to 13. Put this in the value field. These container fields can also be used on carts.

CONT_CLOSEABLE	1
CONT_PICKPROOF	2
CONT_CLOSED	4
CONT_LOCKED	8

DRINK LIQUID TYPES

If value4 is set to 0 the container junks when empty. If value4 is set to 1 the container can be refilled when empty.

LIQ_WATER	0
LIQ_BEER	1
LIQ_WINE	2
LIQ_ALE	3
LIQ_DARK_ALE	4
LIQ_WHISKEY	5
LIQ_JUICE	6
LIQ_SPIRITS	7
LIQ_PORT	8
LIQ_SLIME_MOLD	9
LIQ_MILK	10
LIQ_TEA	11
LIQ_COFFEE	12
LIQ_BLOOD	13
LIQ_SALTWATER	14
LIQ_COLA	15
LIQ_MEAD	16
LIQ_GROG	17

HERB AND COMPONENT TYPES

longer exists and has been merged with ITEM_TYPE_COMPONENT. Value2 on a component is the herb type. See the list below for a list of current herbs and their bit vectors.

Typing `slookup herbs` on the testport will bring up a list of current herbs. Note that the number in the list does NOT necessarily correspond with the herb's actual number. To find out what the herb's number is, type `slookup herbname` for more information on that herb, or use the table below.

HERB TYPE	BIT VECTOR
HERB_PIPEWEED	0
HERB_DHAT	1
HERB_DWALE	2
HERB_KONEION	3
HERB_MONKSHOOD	4
HERB_CATNIP	5
HERB_CANDLESTICK_PLANT	6
HERB_ADDERS_TONGUE	7
HERB_ALLCURE	8
HERB_ALOE	9
HERB_ARNICA	10
HERB_BLOODROOT	11
HERB_COMFREY	12
HERB_DWALE	13

HERB_ECHINACEA	14
HERB_WOUNDWORT	15
HERB_WORMWOOD	16
HERB_ALL_SAINTS_WORT	17
HERB_CURE_ALL	18
HERB_SHANGNUM_MOSS	19
HERB_MARSH-MALLOW	20
HERB_LUNGWORT	21
HERB_BIRTHWORT	22
HERB_BUGSBANE	23
HERB_SNAKESALVE	24
HERB_SKULLCUP	25
HERB_BING_LANG	26
HERB_HEATHER	27
HERB_HENBANE	28
HERB_HUSHTHORN	29
HERB_JUNIPER	30
HERB_KOLO	31
HERB_BILLBERRY	32
HERB_DARKWEED	33

HERB_GINKO	34
HERB_WINTERSALVE	35

COIN TYPES

To make coins for use on the testport and within progs use the command `mpmakecash`. The syntax to make 100 platinum for instance would be `mpmakecash 100 4`.

`COIN_COPPER 0`
`COIN_SILVER 1`
`COIN_ELECTRUM 2`
`COIN_GOLD 3`
`COIN_PLATINUM 4`

TRAP TYPES

See the lessons on making traps and putting traps in resets for more information on using these types. A current list of traps can be seen by typing `showtraps` on the testport with your builder character. More information about the trap can be seen by typing `showtrap #`.

TRAP TYPE	BIT VECTOR	COMMENT/INFORMATION
TTYPE_NONE	0	No trap at all
TTYPE_SPIKE_MINOR	1	a minor spike trap : 5d6/PIERCE on CHAR
TTYPE_SPIKE_AVERAGE	2	an average spike trap : 10d6/PIERCE on CHAR
TTYPE_SPIKE_STRONG	3	a strong spike trap : 15d6/PIERCE on CHAR
TTYPE_SPIKE_DEADLY	4	a deadly spike trap : 25d6/PIERCE on CHAR
TTYPE_BLADE_MINOR	5	a minor spinning blade trap : 5d6/SLASH on CHAR
TTYPE_BLADE_AVERAGE	6	an average spinning blade trap : 10d6/SLASH on CHAR
TTYPE_BLADE_STRONG	7	a strong spinning blade trap : 15d6/SLASH on CHAR
TTYPE_BLADE_DEADLY	8	a deadly spinning blade trap : 25d6/SLASH on CHAR
TTYPE_STONE_MINOR	9	a minor falling stone trap : 5d6/BASH on CHAR
TTYPE_STONE_AVERAGE	10	an average falling stone trap : 10d6/BASH on CHAR
TTYPE_STONE_STRONG	11	a strong falling stone trap : 15d6/BASH on CHAR
TTYPE_STONE_DEADLY	12	a deadly falling stone trap : 25d6/BASH on CHAR
TTYPE_ACID_MINOR	13	a minor acid trap : 5d8/ACID on CHAR
TTYPE_ACID_AVERAGE	14	an average acid trap : 10d8/ACID on CHAR

TTYPE_ACID_STRONG	15	a strong acid trap : 15d8/ACID on CHAR
TTYPE_ACID_DEADLY	16	a deadly acid trap : 25d8/ACID on CHAR
TTYPE_FROST_MINOR	17	a minor frost trap : 10d4/COLD on CHAR
TTYPE_FROST_AVERAGE	18	an average frost trap : 20d4/COLD on CHAR
TTYPE_FROST_STRONG	19	a strong frost trap : 30d4/COLD on CHAR
TTYPE_FROST_DEADLY	20	a deadly frost trap : 50d4/COLD on CHAR
TTYPE_FIRE_MINOR	21	a minor fire trap : 10d4/FIRE on CHAR
TTYPE_FIRE_AVERAGE	22	an average fire trap : 20d4/FIRE on CHAR
TTYPE_FIRE_STRONG	23	a strong fire trap : 30d4/FIRE on CHAR
TTYPE_FIRE_DEADLY	24	a deadly fire trap : 50d4/FIRE on CHAR
TTYPE_ELECTRICITY_MINOR	25	a minor electrical trap : 8d6/ELECTRICITY on CHAR
TTYPE_ELECTRICITY_AVERAGE	26	an average electrical trap : 16d6/ELECTRICITY on CHAR
TTYPE_ELECTRICITY_STRONG	27	a strong electrical trap : 25d6/ELECTRICITY on CHAR
TTYPE_ELECTRICITY_DEADLY	28	a deadly electrical trap : 42d6/ELECTRICITY on CHAR
TTYPE_NEG_ENERGY_MINOR	29	a minor negative energy trap : 8d8/HEALING on CHAR
TTYPE_NEG_ENERGY_AVERAGE	30	an average negative energy trap : 16d8/HEALING on CHAR
TTYPE_NEG_ENERGY_STRONG	31	a strong negative energy trap : 25d8/HEALING on CHAR
TTYPE_NEG_ENERGY_DEADLY	32	a deadly negative energy trap : 42d8/HEALING on CHAR

TTYPE_SPELL_RAZORBAIT	33	a razorbait spell trap
TTYPE_SPELL_SWORDBAIT	34	a swordbait spell trap
TTYPE_SPELL_WINTER_MIST	35	a winter mist spell trap
TTYPE_SPELL_BLAZEBANE	36	a blazebane spell trap
TTYPE_SPELL_CHARGED_BEACON	37	a charged beacon spell trap
TTYPE_SPELL_WEAKEN	38	a weaken spell trap
TTYPE_SPELL_FUMBLE	39	a fumble spell trap
TTYPE_SPELL_CURSE	40	a curse spell trap
TTYPE_SPELL_ILL_FORTUNE	41	an ill fortune spell trap
TTYPE_SPELL_BLINDNESS	42	a blindness spell trap
TTYPE_SPELL_ENTANGLE	43	an entangle spell trap
TTYPE_SPELL_HOLD_MONSTER	44	a hold monster spell trap
TTYPE_SPELL_RAINBOW_PATTERN	45	a rainbow pattern spell trap
TTYPE_SPELL_COLOR_SPRAY	46	a color spray spell trap
TTYPE_SPELL_FAERIE_FIRE	47	a faerie fire spell trap
TTYPE_SPELL_POISON	48	a poison spell trap
TTYPE_SPELL_DISPEL_MAGIC	49	a dispel magic spell trap
TTYPE_SPELL_CAUSE_LIGHT	50	a cause light spell trap
TTYPE_SPELL_CAUSE_SERIOUS	51	a cause serious spell trap
TTYPE_SPELL_CAUSE_CRITICAL	52	a cause critical spell trap
TTYPE_SPELL_HARM	53	a harm spell trap

TTYPE_SPELL_SHOCKING_GRASP	54	a shocking grasp spell trap
TTYPE_SPELL_BURNING_HANDS	55	a burning hands spell trap
TTYPE_SPELL_CHILL_TOUCH	56	a chill touch spell trap
TTYPE_SPELL_MAGIC_MISSILE	57	a magic missile spell trap
TTYPE_SPELL_ACID_ARROW	58	an acid arrow spell trap
TTYPE_SPELL_FLAME_ARROW	59	a flame arrow spell trap
TTYPE_SPELL_FLAMESTRIKE	60	a flamestrike spell trap
TTYPE_SPELL_PHOENIX_CLAW	61	a phoenix claw spell trap
TTYPE_SPELL_FIREBALL	62	a fireball spell trap
TTYPE_SPELL_SOUND_BURST	63	a sound burst spell trap
TTYPE_SPELL_ACID_BLAST	64	an acid blast spell trap
TTYPE_SPELL_LIGHTNING_BOLT	65	a lightning bolt spell trap
TTYPE_SPELL_CHAIN_LIGHTNING	66	a chain lightning spell trap
TTYPE_SPELL_CONE_OF_COLD	67	a cone of cold spell trap
TTYPE_SPELL_ICE_STORM	68	an ice storm spell trap
TTYPE_SPELL_ENERGY_DRAIN	69	an energy drain spell trap
TTYPE_SPELL_PHANTASMAL_KILLER	70	a phantasmal killer spell trap
TTYPE_SPELL_DISINTEGRATE	71	a disintegrate spell trap

TRAP TRIGGERS

A trap object can use more than one trigger flag, it needs to be separated by a pipe. TRIGGER_PICK triggers on knock, doorbash and pick. Doorbash will check TRIGGER_PICK, then TRIGGER_UNLOCK then TRIGGER_OPEN. If two of those triggers are used it will only trigger once. So when coded into an area, the lowest level trigger should be used, ie no need to use TRIGGER_PICK and TRIGGER_UNLOCK because anything that triggers TRIGGER_PICK will also trigger TRIGGER_UNLOCK.

TRIGGER	BIT VECTOR
TRIGGER_NONE	0
TRIGGER_GET	1
TRIGGER_OPEN	2
TRIGGER_SHOVE	4
TRIGGER_PUT	8
TRIGGER_EXAMINE	16
TRIGGER_USE	32
TRIGGER_UNLOCK	64
TRIGGER_CLOSE	128
TRIGGER_MOVE	256
TRIGGER_PICK	512

TRIGGER_PICK will trigger on knock, doorbash and pick

Doorbash will check TRIGGER_PICK, then TRIGGER_UNLOCK then TRIGGER_OPEN.

If two of those triggers are used it will only trigger once. So when coded, the lowest level trigger should be used, ie no need to use TRIGGER_PICK and TRIGGER_UNLOCK because anything that triggers PICK will also trigger unlock.

LEVER TRIGGER FLAGS

The following triggers can be used on levers and buttons. For some of the triggers that open doors, you will need to make sure that you have TRIG_DOOR utilised. You can have more than one trigger on a lever or button. Just seperate them with a pipe | . Buttons should have the TRIG_UP and TRIG_AUTORETURN triggers set on them. Anything that is only meant to be pulled should have the TRIG_AUTORETURN trigger set.

TRIGGER	BIT VECTOR	COMMENT
TRIG_NONE	0	Use 0 when there are no triggers
TRIG_UP	1	Lever starts out in the up position
TRIG_UNLOCK	2	This will set a newly created exit or existing one to unlocked
TRIG_LOCK	4	This will set a newly crated exit or exiting one to locked
TRIG_D_NORTH	8	Sets the exit to open to the north
TRIG_D_SOUTH	16	Sets the exit to open to the south
TRIG_D_EAST	32	Sets the exit to open to the east
TRIG_D_WEST	64	Sets the exit to open to the west
TRIG_D_UP	128	Sets the exit open up
TRIG_D_DOWN	256	Sets the exit to open down
TRIG_DOOR	512	This is required to make any of the triggers utilising exits work
TRIG_CONTAINER	1024	This trigger is not used
TRIG_OPEN	2048	This trigger opens any door that is generated with other triggers
TRIG_CLOSE	4096	This trigger closes any door that is generated with other exit triggers
TRIG_PASSAGE	8192	This trigger is needed to create a new exit.

TRIG_OLOAD	16384	Loads up an object. Value1 is the Room. Value2 is the mob number.
TRIG_MLOAD	32768	Loads up a mobile. Value1 is the Room. Value2 is the mob number.
TRIG_TELEPORT	65536	Teleports lever puller or button pusher to set vnum
TRIG_TELEPORTALL	131072	Teleports everyone in the room to set vnum
TRIG_TELEPORTPLUS	262144	This trigger is not used
TRIG_DEATH	524288	This trigger is not used
TRIG_CAST	1048576	Casts a spell on lever/button pusher. Value1 needs to contain the spell number
TRIG_FAKEBLADE	2097152	This trigger is not used
TRIG_RANDFOUR	4194304	Randomises the existing exits of the vnum in value1 to NSEW. It will not add new exits, just change where the existing ones go.
TRIG_RANDSIX	8388608	Randomises the existing exits of the vnum in value1 to NSEWDU. It will not add new exits, just change where the existing ones go.
TRIG_TRAPDOOR	16777216	This trigger is not used
TRIG_ANOTHEROOM	33554432	This trigger is not used
TRIG_USEDIAL	67108864	This trigger is not used
TRIG_ABSOLUTEVNUM	134217728	This trigger is not used
TRIG_SHOWROOMDESC	268435456	Required in order to allow the teleported to know that they have been teleported
TRIG_AUTORETURN	536870912	This will make the trigger go back to the original position
TRIG_NOTRAP	536870912	Will bypass any traps on the exit if used. It allows for a trap to be on a door that will not trigger if the lever is used to open it instead of other means.

FURNITURE STATES

FURNITURE_CHAIR	0
FURNITURE_BED	1
FURNITURE_LECTERN	2
FURNITURE_ALTAR	3

If furniture is set as a chair, a PC can sit on it. If furniture is set as a bed, a PC can rest and sleep on it. They must first sit on it. If furniture is set as a lectern, a PC can stand at it, and if it is an altar they can kneel at it.

OBJECT ID CODE

When an object gets a magical apply via the APPLY system, when it is identified, that property shows up in the identify output. However, when an object gets a magical affect by an object prog, this does not show up. Sometimes a builder does not want the magic of the item easily known via identify, but other times they do. When you do want it known you can add an I section to the item.

```
#25752
gold ivory bracer golden rose mage school guild reward~
{B0}a gold and {F0}ivory {F0}bracer~
{B0}A gold and {F0}ivory {F0}bracer lays here.~
~
ITEM_TYPE_TREASURE
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_WRIST
QUALITY_AVERAGE MATERIAL_GOLD COND_PERFECT SIZE_MEDIUM
0 0 0 0 LAYER_OVER 0
E
gold ivory bracer goldenrose mage school~
{B0}An elegant bracer patterned in rose and jasmine etchings.
~
A APPLY_MANA 10
I
This bracer has a magical property that drains anyone who is not a mage
who wears it.
~
>rand_prog 80~
if guild($c) != Mages
  mpmadd $c currmana -20
  mpmadd $c currhp -40
  mpmadd $c currmove -30
  mpechoat $c $O is draining your very essence away from you.
  mpechoaround $c $C pales as $s essence is drained away by $O.
endif
~
```

The text that is after the I will show up when the PC casts identify on the object. The mana apply on the object will also show up in the identify spell output. The information does not have to be super specific. General ID does not give out information in specific numbers anyway, so it is best to follow along with the general theme we have set up for the game.

Resizing Quest Objects

Your quest gives out an object that is to be worn by the PC, and it has been especially made for them. Since the introduction of all the races, do do if checks for all the races to determine what size to make the item would be extremely tedious. So we added a very simple command to allow the item to be resized to the size of the PC.

Place the following command AFTER you load up the item in the prog, and BEFORE you give it to the PC. If it is an item you want to be truely unique we encourage you to also add the mark to the object to help immortals keep track of any abuse.

```
mpresize iVNUM $n
```

PROGRAM LAYOUT

SPACING OF PROGRAMS

It is important to use spacing in your programs. It helps to see, and find errors in programs. For every if check you should indent your program by 2 spaces. This way you can see that every if check has its own endif.

The area admins will insist on correctly spaced programs. It makes it far easier for us to be able to read the programs, and as my old programming teacher drummed into me, neatly laid out code is very important.

```
>time_prog 6~
if rand(20)
  mpgoto 8030
  mpecho The Town Crier starts his rounds.
else
  if rand(20)
    mpgoto 8511
    mpecho The Town Crier starts his rounds.
  else
    if rand(20)
      mpgoto 8385
      mpecho The Town Crier starts his rounds.
    else
      if rand(20)
        mpgoto 8810
        mpecho The Town Crier starts his rounds.
      else
        mpgoto 8631
        mpecho The Town Crier starts his rounds.
      endif
    endif
  endif
endif
~
|
```

The above prog is the program from the Waterdeep Town Crier. For every if check the following parts of the program is indented two spaces. It is very easy to match up the following endifs. The more complex a program gets the more important this spacing becomes to you. It really does make a difference, Dalvyn has created some super long programs that actually used up the coded limit for programs. He had to remove his spaces, to make them fit. He had a missing endif in there one time, and it was near impossible to find because the program was not spaced.

TILDAS AND PIPES

Often new builders are not sure where tildas (~) and pipes (|

a tilda, and at the end of the program. The pipe only goes AFTER all the progs for that particular mobile/object/room. Each program that the mobile has is ended by a tilda.

Here is a sample of multiple progs on a mobile.

```
#8050
Piergieron~
Piergieron~
Lord Piergieron stands here.
~
He is the acknowledged ruler of Waterdeep, but in reality he is just one
of the council of twelve Lords of Waterdeep.  Most of the lords are secret,
but it is generally know that Piergieron is one of the council.
~
U 50 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_REQUEST|ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 1000
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
%15 2 charisma~
> death_prog 100~
if questr(15100, 19, 5, $n) == 12
  mpoload 8073
  mpechoat $n You have killed the most well known Lord of Waterdeep!
  mpechoaround $n $n has killed the most well known Lord of Waterdeep!
  mpmset $n questr 15100 19 5 13
endif
~
>intercept_prog sleep~
sayto $n You cannot sleep here.
sayto $n Go pay for a room in an inn.
~
>give_prog i7007~
sayto $n Argh! What are you giving me this disgusting thing for?
drop i7007
```

```
mpecho Lord Piergieron grimaces in disgust.
sayto $n My officials handles the rewards for this head.
sayto $n During the day they can be found any of the gates.
~
|
```

Lord Piergieron has 3 programs on him for the purpose of this example, in the game he actually has much more. Each is finished by a tilda, but at the end of the last one before the next mobiles information starts is the pipe.

OR FUNCTION

The or function allows you to check if more than one instance of the same kind of thing is true in a prog. For instance if you wanted to check if the PC was either a priest or wizard, and if they were then the next part of the program would activate.

```
>intercept_prog molira tethos~
if class($n) == Wizard
or class($n) == Priest
  if wear_loc($o) != -1
    if objval5($o) == 0
      cast 'sanctuary' $n
      mposet on $n iQQ01 value5 1
      mpechoat $n {80}A black aura surrounds your body making you feel strangely
protected.
      mpechoaround $n {80}A deep black aura swirls around $N's body.
    endif
  endif
endif
~
|
```

The above is an object program. If the PC is either a priest or a wizard, the item will then move on to the next part of the program and see if the next things are true. These are if the item is worn, and if object value5 is equal to 0.

TAVERN PROGRAMS

Simple method using quest bits in the area

Many of the taverns and inns around the kingdoms offer a room for the night for a few copper or silver. To follow is the programs on the inn keeps and tavern guards in Waterdeep. Further down is a more complicated method, the one that is employed by the Lucky Drunk Tavern in Waterdeep.

First of all a basic inn setup needs to have two rooms for this to work. Any more than two rooms will make for different programs.

Make an Innkeeper in your tavern. They can of course also be a shop and sell ale and food and other bits and pieces.

```
#8041
innkeeper~
an innkeeper~
An innkeeper stands behind the bar here.
~
Dressed in a plain white shirt and pants, he wipes down the bar.
~
U 25 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 700
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_NONE RIS_NONE RIS_NONE
>greet_prog 50~
sayto $n So what will it be? A hearty meal? An ale?
sayto $n Or for 3 copper you can get a bed for the night.
~
>bribe_prog 3~
sayto $n Go on up.  Our mattresses are made of nice soft feathers.
sayto $n Sleep well.
mpmset $n quest 15 1 1
mpforce $n up
mpjunk coins
~
>intercept_prog sleep~
sayto $n What ye be thinking you can sleep down here like a drunk?
```

```
  sayto $n Either pay 3 copper for a room or be off with ye!
  ~
  |
```

This innkeeper has a 50% chance of telling the PC that he has rooms for rent. They cost 3 copper. When he is given the copper, he sets a quest bit up on the PC and forces them up. Make sure you junk the coins. He also makes sure that no one sleeps in his tavern main room.

Now make a guard to put in the bedroom.

```
#8047
tavern inn guard~
a tavern guard~
A tavern guard is here making sure you have paid.
~
He looks pretty mean.  His job is to make sure that all who try to sleep
in the rooms above the inn have paid their way.
~
U 45 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 800
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
>greet_prog 100~
if quest(15,1,$n) == 0
  sayto $n Hey you cant come up here without paying!
  sayto $n Go down and pay your copper before trying to sleep up here.
  mpforce $n down
else
  if quest(15,1,$n) ==1
    sayto $n Sleep well.
    mpmset $n quest 15 1 0
  endif
endif
~
|
```

This guard checks for quest bits. If they are not set to having paid he kicks the player out. I put the same guard in all the taverns in a city. So if you force them in a direction you need to make sure that your bedroom is in the same direction in the Taverns. If they have paid he sets their bits back to 0 so the next time they want to rent a room they have to repay.

Simple method checking Waterdeep quest bits

If you want to save quest bits in your area, you can actually use the quest bits in Waterdeeps area by doing the following. In fact I fully encourage you to do so.

```
#8041
innkeeper~
an innkeeper~
An innkeeper stands behind the bar here.
~
Dressed in a plain white shirt and pants, he wipes down the bar.
~
U 25 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 700
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON|LANG_DWARVEN
LANG_COMMON|LANG_DWARVEN
RIS_NONE RIS_NONE RIS_NONE
>greet_prog 50~
sayto $n So what will it be? A hearty meal? An ale?
sayto $n Or for 3 copper you can get a bed for the night.
~
>bribe_prog 3~
sayto $n Go on up.  Our mattresses are made of nice soft feathers.
sayto $n Sleep well.
mpmset $n questr 8000 15 1 1
mpforce $n up
mpjunk coins
~
>intercept_prog sleep~
sayto $n What ye be thinking you can sleep down here like a drunk?
sayto $n Either pay 3 copper for a room or be off with ye!
~
|
```

```
#8047
tavern inn guard~
a tavern guard~
A tavern guard is here making sure you have paid.
~
He looks pretty mean.  His job is to make sure that all who try to sleep
in the rooms above the inn have paid their way.
~
U 45 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 800
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
>greet_prog 100~
if questr(8000,15,1,$n) == 0
  sayto $n Hey you cant come up here without paying!
  sayto $n Go down and pay your copper before trying to sleep up here.
  mpforce $n down
else
  if quest(8000,15,1,$n) ==1
    sayto $n Sleep well.
    mpmset $n questr 8000 15 1 0
  endif
endif
~
|
```

Note the use of questr to check quest bits in vnum block 8000 which is Waterdeep.

More complicated Lucky Drunk Method

The Lucky Drunk method 2 seperate vnum keys, one for each room. Make sure that each room has a different cost, because you will need to use bribe progs of different values to make this work.

Also, you do not want PC's running around with keys outside of the area. So you need to get them to return it when they are done. One way would be to put a "deposit" on the key, they get some money back when they return the key to the innkeeper. Another, and something you could do as well, would be to have a guard demand the key returned before allowing them to leave the inn area. You could use quest bits to check this as well. You may need to make your area unable

to be recalled out of or disallow the use of transport spells. If you did this, it would need to be IC.

Here are some samples of progs from the lucky drunk.

```
>speech_prog optional stocked dry bar~
sayto $n The luxury suites and honeymoon suite each have an optional stocked
sayto $n dry bar. For 10 platinum I will have your dry bar stocked.
~
>bribe_prog 5000~
sayto $n Okay. I will have your dry bar stocked. Enjoy.
smile $n
mpmset $n quest 0 1 1
mpjunk coins
~
```

The mobile has offered choices to the PC, and then has speech_progs that activate when the PC makes their choice of room.

Also a check can be made to make sure the PC does not already have a key in his or her possession. See below.

```
>intercept_prog buy~
if actorhasobjnum(9826)
 sayto $n You already have a key.
 sayto $n Give it to me if you want
 sayto $n another one.
else
 if actorhasobjnum(9827)
   sayto $n You already have a key.
   sayto $n Give it to me if you want
   sayto $n another one.
 else
   if actorhasobjnum(9828)
     sayto $n You already have a key.
     sayto $n Give it to me if you want
     sayto $n another one.
   else
     if actorhasobjnum(9829)
       sayto $n You already have a key.
      sayto $n Give it to me if you want
      sayto $n another one.
     else
```

```
        mpunintercept
      endif
    endif
  endif
endif
~
```

And don't forget progs to make sure $n does not leave with the key.

```
>intercept_prog south~
if inroom($i) == 9800
  if actorhasobjnum(9826)
   sayto $n Hey there.
   sayto $n You cannot leave with a room key.
   sayto $n Give it back to me please.
  else
    if actorhasobjnum(9827)
      sayto $n Hey there.
      sayto $n You cannot leave with a room key.
      sayto $n Give it back to me please.
    else
      if actorhasobjnum(9828)
        sayto $n Hey there.
        sayto $n You cannot leave with a room key.
        sayto $n Give it back to me please.
      else
        if actorhasobjnum(9829)
          sayto $n Hey there.
          sayto $n You cannot leave with a room key.
          sayto $n Give it back to me please.
        else
          mpunintercept
        endif
      endif
    endif
  endif
else
  mpunintercept
endif
~
```

Now the above method will not work if $n has the key in their pack. So to overcome this, Caius flagged his keys as type

inventory which prevents the PC from putting them into their pack. This has been pretty effective at stopping PCs from leaving the area with the keys. Caius also made pertinant areas in his area no recall to stop them recalling with the key.

```
#9827
economy key~
{70}an economy key~
{70}A plain key lies on the ground here.~
~
ITEM_TYPE_KEY
FLAG_INVENTORY
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_IRON COND_SUPERB SIZE_MEDIUM
0 0 0 0 0 0
E
economy key~
{70}It's a plain iron key with a long thin barrel. On both sides the
key is marked, "Lucky Drunk".
~
```

YELL FOR HELP PROGRAMS

Please note that the new justice system code for a Guard vnum will protect your area. See the lesson about the Justice System for more detail. However, if a mobile has a fight prog, they will not utilise the justice system guard to call for help. The following system, was our old system and is not used for MOST mobiles. It is used for mobiles who have a fight prog for some other reason.

The mobiles in your area come under attack from a PC. Do they fight back, do they run for their lives, do they call for help? It all depends on what is IC for that area. But for many areas, like temples and cities, if someone is attacked, then help will come running. The following are progs that all for help to come.

```
>fight_prog 100~
if ispc($n)
  if rand(50)
    yell Guards! $N is attacking me!
    yell I am on $b.
    if rand(50)
      if quest(0,3,self) < 3
        mpecho A guard comes rushing to the aid of $I.
        mpmload 22850
        mpforce guardian mpkill $n
        mpmadd self quest 0 3 1
      endif
    endif
  endif
endif
~
|
```

When the mobile with this prog is attacked, he will yell for the guards. He will state where he is, which is $b. $b will echo the room name. Then 50% of the time a guard will be loaded up. Quest bits are set on the mob each time a guard is loaded. To prevent the room being filled with an unlimited amount of guards, the quest bits will only allow 3 guards maximum to come running to the aid of the mobile.

These progs load up a naked guard. You could get more elaborate with your prog, and load them into a room you have set up for that purpose, and with mpats, force them to load equipment with mpoload and force them to wear it, and then force them to mpgoto to the room where the mobile under attack is. An equiped guard makes them a more formidable foe. To prevent abuse of the fact that equipment is being loaded, it is good to include in the death prog, and mpjunk of the equipment. It can happen all the time or it can happen a percentage of the time.

Note that when you are setting the quest bits on a mob, it sets the quest bits for ALL of the mobs of the same VNUM. So that is why it is good to set the bit back to 0 when the mob dies, so that if another mob is attacked they can call for help and get guards as well.

SHOOTING MOBS

You have protected your area with archers, but they only attack when the PC is in the room with them, not from a distance. How do you make them attack a PC from a room or two away like an archer should?

Use a greet prog in the rooms before the archers, and with the use of the mpat command you can get your archers happily shooting PC intruders. Another is to put fight progs on the mobiles who are currently fighting the PC, to have the archers shoot from the other room to aid in the battle. To follow are samples from the Storm Keep area where mobiles with ranged weapons shoot at PC's as the enter the hallway.

```
#2201
Troll Guard~
{20}a troll guard~
{20}A troll guard stands here.
~
This troll is unusually big and ugly, with dark yellowing teeth and
breath that reeks of rotting flesh. He is hunched over, and though his
red eyes are filled with malice they show little signs of intelligence.
~
S 45 CLASS_FIGHTERS RACE_TROLL SEX_MALE POS_STANDING DEITY_NONE ACT_SENTINEL
>greet_prog 60~
say You not welcome. Go away!
mpkill $n
~
>fight_prog 80~
if inroom($i) == 2200
or inroom($i) == 2201
or inroom($i) == 2202
  mpat 2203 mpforce m2208 mpoload 2207
  mpat 2203 mpforce m2208 shoot south $n
  mpechoat $n {80}Someone shoots a crossbow bolt at you, from the north.
  mpechoaround $n {80}Someone shoots a crossbow bolt at $N, from the north.
else
  if inroom($i) == 2212
  or inroom($i) == 2211
    mpat 2203 mpforce m2208 mpoload 2207
    mpat 2203 mpforce m2208 shoot east $n
    mpechoat $n {80}Someone shoots a crossbow bolt at you, from the west.
    mpechoaround $n {80}Someone shoots a crossbow bolt at $N, from the west.
  endif
endif
~
|
```

In the above example when this troll is under attack, trolls that are a room or two away will shoot into the battle at the PC. What is being loaded is a fresh crossbow bolt. The trolls have the actual crossbow on them in resets, but the code makes them load a fresh bolt each time.

In another area I use a greet prog in the room to get mobiles to shoot from a few rooms away at an entering PC. See the sample below.

```
>greet_prog 100~
if level($n) > 10
  mpat 6592 mpforce m6500 shoot south $n
endif
~
|
```

It should be noted that if the mobiles are not in the room in question, then this prog will not work. So if the PC has already killed the mobile then it wont work. You will perhaps need to make sure that your mobiles are sentinel so that there are there to shoot at the PC, and not wandered off to another room.

GAMBLING PROGRAMS

Gambling programs can get very complex. It is something that in general only an experienced builder comfortable with mob programs can do. The following program was written by Dalvyn, whom I would consider our most savvy area coder. As you can see its quite a complex prog.

```
#15906
gamemaster master~
{70}a gamemaster~
{70}A gamemaster, clad in a silvery robe, stands here.~
This aged human seems to enjoy his work in the Gamblin Hall very
much. He is clad in a silvery robe decorated with a silver coin,
the symbol of Tymora.
~
U 35 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_TYMORA
ACT_SENTINEL|ACT_CITIZEN
0
ARMOR_TYPE_CLOTH MATERIAL_CLOTH
d8+2 650
13 13 13 18 13 13 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
0 0 0
>time_prog 5~
wake
mpwalkto 15921
~
>time_prog 23~
yawn
say Enough for today. I need some rest.
mpwalkto 15927
~
>arrival_prog 15927~
yawn
sleep
~
>greet_prog 100~
mpmset $n quest 20 9 0
mpmset $n quest 30 1 0
mpmset $n questr 8200 4 1 0
sayto $n If you want to play, you have found the right place!
sayto $n Read the rules on this plaque
```

```
sayto $n and give me a token, then we will start.
mpecho The gamemaster points to the plaque on the wall.
~
>give_prog 15900~
if quest(20,3,$n) == 0
  mpjunk all
  mpmset $n quest 20 3 1
  sayto $n Very well... we will play for a copper token.
  sayto $n Throw the dice when you want.
else
  sayto $n You have already given me a token
  give token $n
endif
~
>give_prog 15901~
if quest(20,3,$n) == 0
  mpjunk all
  mpmset $n quest 20 3 2
  sayto $n Very well... we will play for a silver token.
  sayto $n Throw the dice when you want.
else
  sayto $n You have already given a token.
  give token $n
endif
~
>give_prog 15902~
if quest(20,3,$n) == 0
  mpjunk all
  mpmset $n quest 20 3 3
  sayto $n Very well... we will play for a gold token.
  sayto $n Throw the dice when you want.
else
  sayto $n You have already given me a token
  give token $n
endif
~
>give_prog 15903~
if quest(20,3,$n) == 0
  mpjunk all
  mpmset $n quest 20 3 4
  sayto $n Very well... we will play for a platinum token.
  sayto $n Throw the dice when you want.
else
  sayto $n You have already given me a token
  give token $n
endif
```

```
~
>give_prog 15904~
if quest(20,3,$n) == 0
  mpjunk all
  mpmset $n quest 20 3 5
  sayto $n Very well... we will play for a luck token.
  sayto $n Throw the dice when you want.
else
  sayto $n You have already given me a token
  give token $n
endif
~
>intercept_prog throw~
if quest(20,3,$n) > 0
  if quest(23,3,$n) == 0
    if rand(16)
      mpechoat $n You throw the dice and get 1 star.
      mpechoaround $n $n throws a dice and gets 1 star.
      mpmset $n quest 23 3 1
    else
      if rand(20)
        mpechoat $n You throw the dice and get 2 stars.
        mpechoaround $n $n throws a dice and gets 2 stars.
        mpmset $n quest 23 3 2
      else
        if rand(25)
          mpechoat $n You throw the dice and get 3 stars.
          mpechoaround $n $n throws a dice and gets 3 stars.
          mpmset $n quest 23 3 3
        else
          if rand(33)
            mpechoat $n You throw the dice and get 4 stars.
            mpechoaround $n $n throws a dice and gets 4 stars.
            mpmset $n quest 23 3 4
          else
            if rand(50)
              mpechoat $n You throw the dice and get 5 stars.
              mpechoaround $n $n throws a dice and gets 5 stars.
              mpmset $n quest 23 3 5
            else
              mpechoat $n You throw the dice and get 6 stars!
              mpechoaround $n $n throws a dice and gets 6 stars!
              mpmset $n quest 23 3 6
            endif
          endif
        endif
```

```
      endif
    endif
    sayto $n Do you want to keep this result?
    sayto $n Or to throw again?
    mpmset $n questr 8200 4 1 1
  else
    sayto $n If you want to throw the dice again, say so.
    sayto $n Otherwise, tell me that you want to keep this result.
  endif
else
  sayto $n You must give me a token first!
endif
~
>speech_prog keep~
if questr(8200,4,1,$n) == 1
  if quest(20,3,$n) > 0
    if quest(23,3,$n) > 0
      mpmset $n questr 8200 4 1 0
      sayto $n Alright. My turn to throw the dice now!
      if rand(8)
        mpecho The gamemaster throws a dice and gets 1 star.
        mpmset $n quest 26 3 1
      else
        if rand(10)
          mpecho The gamemaster throws a dice and gets 2 stars.
          mpmset $n quest 26 3 2
        else
          if rand(12)
            mpecho The gamemaster throws a dice and gets 3 stars.
            mpmset $n quest 26 3 3
          else
            if rand(20)
              mpecho The gamemaster throws a dice and gets 4 stars.
              mpmset $n quest 26 3 4
            else
              if rand(50)
                mpecho The gamemaster throws a dice and gets 5 stars.
                mpmset $n quest 26 3 5
              else
                mpecho The gamemaster throws a dice and gets 6 stars!
                mpmset $n quest 26 3 6
              endif
            endif
          endif
        endif
      endif
```

```
    else
      mpechoat $n The gamemaster motions for you to throw the dice.
      mpechoaround $n The gamemaster motions for $n to throw the dice.
    endif
  else
    sayto $n You should give me a token first!
  endif
endif
~
>speech_prog throw again~
if questr(8200,4,1,$n) == 1
  if quest(20,3,$n) > 0
    if quest(23,3,$n) > 0
      mpmset $n questr 8200 4 1 0
      mpechoat $n The gamemaster motions for you to throw the dice again.
      mpechoaround $n The gamemaster motions for $n to throw the dice again.
      if rand(16)
        mpechoat $n You throw the dice and get 1 star.
        mpechoaround $n $n throws a dice and gets 1 star.
        mpmset $n quest 23 3 1
      else
        if rand(20)
          mpechoat $n You throw the dice and get 2 stars.
          mpechoaround $n $n throws a dice and gets 2 stars.
          mpmset $n quest 23 3 2
        else
          if rand(25)
            mpechoat $n You throw the dice and get 3 stars.
            mpechoaround $n $n throws a dice and gets 3 stars.
            mpmset $n quest 23 3 3
          else
            if rand(33)
              mpechoat $n You throw the dice and get 4 stars.
              mpechoaround $n $n throws a dice and gets 4 stars.
              mpmset $n quest 23 3 4
            else
              if rand(50)
                mpechoat $n You throw the dice and get 5 stars.
                mpechoaround $n $n throws a dice and gets 5 stars.
                mpmset $n quest 23 3 5
              else
                mpechoat $n You throw the dice and get 6 stars!
                mpechoaround $n $n throws a dice and gets 6 stars!
                mpmset $n quest 23 3 6
              endif
            endif
```

```
          endif
        endif
      endif
      sayto $n My turn now!
      if rand(8)
        mpecho The gamemaster throws a dice and gets 1 star.
        mpmset $n quest 26 3 1
      else
        if rand(10)
          mpecho The gamemaster throws a dice and gets 2 stars.
          mpmset $n quest 26 3 2
        else
          if rand(12)
            mpecho The gamemaster throws a dice and gets 3 stars.
            mpmset $n quest 26 3 3
          else
            if rand(20)
              mpecho The gamemaster throws a dice and gets 4 stars.
              mpmset $n quest 26 3 4
            else
              if rand(50)
                mpecho The gamemaster throws a dice and gets 5 stars.
                mpmset $n quest 26 3 5
              else
                mpecho The gamemaster throws a dice and gets 6 stars!
                mpmset $n quest 26 3 6
              endif
            endif
          endif
        endif
      endif
    else
      mpechoat $n The gamemaster motions for you to throw the dice.
      mpechoaround $n The gamemaster motions for $n to throw the dice.
    endif
  else
    sayto $n You should give me a token first!
  endif
endif
~
>rand_prog 60~
if quest(20,3,$r) > 0
 if quest(23,3,$r) > 0
  if quest(26,3,$r) > 0
   mpjunk all
   if quest(20,3,$r) == 1
```

```
 mpoload 15900
 mpoload 15900
else
 if quest(20,3,$r) == 2
  mpoload 15901
  mpoload 15901
 else
  if quest(20,3,$r) == 3
   mpoload 15902
   mpoload 15902
  else
   if quest(20,3,$r) == 4
    mpoload 15903
    mpoload 15903
   else
    mpoload 15904
    mpoload 15904
   endif
  endif
 endif
endif
if quest(23,3,$r) == 6
 if quest(26,3,$r) == 6
  sayto $r Same results! You get reimbursed.
  give token $r
 else
  sayto $r You win!
  give token $r
  give token $r
 endif
else
 if quest(23,3,$r) == 5
  if quest(26,3,$r) == 6
   sayto $r I win... and you lose!
  else
   if quest(26,3,$r) == 5
    sayto $r Same results! You get reimbursed.
    give token $r
   else
    sayto $r You win!
    give token $r
    give token $r
   endif
  endif
 else
  if quest(23,3,$r) == 4
```

```
 if quest(26,3,$r) > 4
  sayto $r I win... and you lose!
 else
  if quest(26,3,$r) == 4
   sayto $r Same results! You get reimbursed.
   give token $r
  else
   sayto $r You win!
   give token $r
   give token $r
  endif
 endif
else
 if quest(23,3,$r) == 3
  if quest(26,3,$r) > 3
   sayto $r I win... and you lose!
  else
   if quest(26,3,$r) == 3
    sayto $r Same results! You get reimbursed.
    give token $r
   else
    sayto $r You win!
    give token $r
    give token $r
   endif
  endif
 else
  if quest(23,3,$r) == 2
   if quest(26,3,$r) > 2
    sayto $r I win... and you lose!
   else
    if quest(26,3,$r) == 2
     sayto $r Same results! You get reimbursed.
     give token $r
    else
     sayto $r You win!
     give token $r
     give token $r
    endif
   endif
  else
   if quest(26,3,$r) == 1
    sayto $r Same results! You get reimbursed.
    give token $r
   else
    sayto $r I win... and you lose!
```

```
          endif
         endif
        endif
       endif
      endif
     endif
    endif
    mpmset $r quest 20 9 0
    mpmset $r quest 30 1 0
    mpjunk all
   endif
  endif
 endif
~
>fight_prog 20~
yell Help! I am being attacked by $N
mpat 4500 yell Help! I am being attacked by $N in Fortune Hall.
if quest(0,3,self) < 5
  mpmadd self quest 0 3 1
  mpecho A guard comes to the help of $I
  if rand(20)
    mpmload 4500
    mpforce m4500 mpkill $n
  else
    mpmload 15911
    mpforce m15911 mpkill $n
  endif
endif
~
>death_prog 100~
mplog WITNESS: (on $v) $I has been killed by $n in Fortune Hall.
mpmset self quest 0 3 0
~
|
```

The first give_prog are to trigger the game, when the game master is given a token (representing the amount that he is willing to bet). The game consists of two contestants rolling 1d6. The player has the choice to reroll once if he wishes (in which case he's forced to keep the second result). It uses qbits on the player to indicate the state: before the first roll, after the first roll, after the second roll, and so on. It uses qbits to store the result of the both the player's and the game master's dice.

20,3 designates the amount that was bet (actually, the type of token). 0 means that no token has been given. 23,3 indicates the result of the player's dice (0 = not thrown, 1-6 = result). 26,3 indicates the result of the game master's dice (0 = not thrown, 1-6 = result)

The program uses 3 intercept progs: throw (for rolling the dice the first time), keep (if the player wants to keep the first result, indicating that the game master must roll his dice), and again (if the player wants to reroll). These intercept prog only determine the results randomly.

Then the rand prog is triggered after some time, and the result of the game is shown (either the player get new tokens if he won, or the qbits are reset if he lost). The other progs are just greet prog, and fight prog with no relation to the actual gambling game.

STABLING PROGRAMS

PC's with pets and mounts like to be able to stable them in a safe place, so they are not killed or stolen from. If you put a stables in your area you are encouraged to have this aspect to your stables.

A stable that sells pets AND stables them will need to have 3 rooms allocated to it. The main room where the buying, selling, and paying for the stabling is done. The second room, which must be the next vnum after the first, where pets are sold, and then a third room, that is not able to be entered by normal means, where PC mounts and pets are stored.

```
#22877
stable selune oracle tristi~
{C0}Tristi~
{C0}Tristi grooms a horse here.
~
{C0}A tall and slender elven woman who works with the horses in
the temple. She has a rare touch with them that some claim is
nearly magical.
~
S 40 CLASS_BARDS RACE_ELF SEX_FEMALE POS_STANDING DEITY_SELUNE
ACT_SENTINEL|ACT_CITIZEN
>greet_prog 100~
sayto $n For 1 gold coin, I can stable your horse.
~
>bribe_prog 100~
mptransfer $n 22891 pet
mpat 22891 mptransfer $n 22889
sayto $n Here you go.
sayto $n Give me this token when you want your mount back.
mpechoat $n $I hands you a starstone token.
mpechoaround $n $I hands $N a starstone token.
mpechoat $n $I leads your mount off to a private stall.
mpechoaround $n $I leads $N's mount off to a private stall.
mpoload 22892
mpgive i22892 $n
mpjunk all
~
>give_prog i22892~
sayto $n Very well, let me get your mount.
mpecho $I heads out to the back of the stables.
mptransfer $n 22891
mpat 22891 mptransfer $n 22889 pet
mpechoat $n $I leads your mount out to you.
```

```
mpechoaround $n $I leads $N's mount out to $m.
mpjunk i22892
~
```

This mob sits in the main room where pets are sold. She doesn't sell anything else because if she did it would actually interfere with the code to sell pets, that is set on that room. When the PC gives coin to stable their mount, they are given an object, a token of some sort that indicates to the PC where they have stabled their mount. The PC AND mount and pet are transferred to the storage room, and then the PC is transferred back out alone. As there are no mpforces to look, this action is not visable to the PC.

When the PC wants their pet/mount back, they give the object to the mobile. The mobile then transfers only the PC to the storage room and then transfers the PC with pets back out to the main room. Again this is not a visable thing to the PC.

RESTRINGING GENERIC OBJECTS

There are some objects in the game that have been hard coded to do specific things, but yet we as builders would like to make them unique. We cannot make a new vnum of them because then the hard code would not work, but we can rename them using progs. Such items that have been renamed in the game for quests etc have been amulets of communication, spell pouches, spell books and so on.

The following is a prog that when the PC buys a spell pouch from the shop keeper, it hands them one that has been renamed to have the guilds colours. This rename is only available to guild members so it helps to make the renamed item unique to that group of characters that is a member of that guild.

```
>buy_prog i68~
if guild($n) == Enchanters
  if actorhasobjnum(25419)
    mpecho $I picks up a black velvet spellpouch off the shelf.
    mposet on $n i68 name i68 velvet spellpouch pouch emerald runes
    mposet on $n i68 short {80}a velvet spellpouch {20}embossed with emerald runes
    mposet on $n i68 long {80}A velvet spellpouch {20}embossed with emerald runes
{80}lies here.
    mposet on $n i68 ed addline 'velvet spellpouch emerald runes' {20}Emerald runes
         sparkle faintly{80}  on the soft black velvet of the spellpouch.
  endif
endif
~
|
```

Vnum 68 is the standard spellpouch. Note for the purpose of happy html, I had to word wrap the ed line. You would not do this in your area file, because then the command would not work.

IN FILE PROGRAMS

In file progs are used when you use the SAME program over many mobiles. We use them on mobs in various cities for citizens to call up guards when they are attacked. If you wish to use in_file_progs discuss them with us soon. Basically the program wanted on all the mobs is saved into a .prg file and stored in a special directory in the area directory of the game. In the mobs information you put in a command that tells the game to use the program in that directory. I will use the example of the Waterdeep Fight Program.

This file is called waterdeepfight.prg, and is stored in its own directory on the game.

```
>fight_prog 20~
if ISPC($n)
  yell Help! Guards! $N is attacking me on $b!
  yell I am on $b.
  if rand(10)
    mpecho A guard comes rushing to the aid of $I.
    mpmload 8023
    mpforce wyverncompanysoldier mpkill $N
  endif
endif
~
|
```

To each mobile that should have that program a line is added - `>in_file_prog waterdeepfight.prg~` . This line of code calls to ths file when the area file is loaded up into the game.

```
#8183
priestess tymoramob~
the priestess of Tymora~
A priestess of Tymora is here playing a game of chance.
~
She is very young and very pretty.  She is dressed simply in
a dress that one would find a tavern wench wearing.  She laughs
merrily as she throws the dice.
~
U 50 CLASS_TYMORA RACE_HUMAN SEX_FEMALE POS_STANDING DEITY_TYMORA
ACT_SENTINEL|ACT_REQUEST|ACT_NOSHOVE|ACT_IS_HEALER|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
```

forum. You can ask general building questions on this forum. If you have a question that would reveal IC information about your area then you are encouraged to email builders@forgottenkingdoms.com with your question.

We are not accepting new hard coders. Anyone who has been in the general mudding community for an extended period of time knows that a team that works well together, for a long period of time is a rare thing. And our team is special, and we work well together. We have no interests in adding anyone else to the team, or in upsetting that special dynamic.

LOGGING EVENTS

These conditions are logged using the mplog command. They are then grepped every month or so and posted to the deities board so they may roleplay accordingly. Any deity who is on line will see the log as it happens. Note that the key logging words need to be capitalised and followed with a colon.

MAKESYMBOL:	Someone has made a holy symbol with an object.
TRADESKILLS:	This shows when someone learns a trade and how many points they get in that trade.
SUPPLICATE:	This one shows us when someone uses the supplicate command. This one is mainly used as part of the supplicate command but if the supplicate is simulated in a temple program to give a special item then it should be logged there.
DEITYFOLLOW:	This one will happen from now on to show when someone has followed a god by wearing a holy symbol. This one is only relevant for holy symbols in pantheon.are
WITNESS:	This one records when someone has killed a mob that may have IC repurcussions. Best not to be used on normal leveling mobs.
GIVEAWAY:	THis one records when an object is given away from one person to another. This is for special objects that need to be kept track of like deity objects.
SACRIFICE:	This is on objects that will have IC repurcussions for being sacrificed. For instance holy symbols and deity objects should all have this.
QUESTCOMPLETE:	This one will be added to many of the quests to show when someone has completed a quest that we want to know about.
STEAL:	Records when someone has stolen or attempted to steal. This on is mainly used in the steal command itself though there may be instances where it is used in an area.
EVENT:	This covers anything that we want recorded in the logs that is not covered by any of the above.
PALADINS:	This covers events associated with the paladin quests.
DWELLINGS:	This covers any events associated with dwellings.
LEGAL:	For all the events with the legal system.

DEATH:	For all instances of death and bringing back from death.
TITHE:	This logs where a PC donates funds to their church.
GLORY:	Logs of glory being given.
BLESS:	Logs of pc being bless or damned.
REWARD:	Logs of rewards and punishments.
KNOWLEDGE:	Logs of raisings in the knowledge skills.
BARDSONG:	Logs of bardsong learnings
DECKOFFATE:	Logs of deck of fate activities
FIGHTERSGUILD:	Logs of fighter's guild activities

Sample code

```
mplog GIVEAWAY: $n has given $o to $t (using give_prog 100)
mplog WITNESS: $n has killed $I on $b in Area Name. (in death progs)
mplog TRADESKILLS: $n has been given 3 skill points in the mining skill.
mplog SACRIFICE: $n has sacrificed 1323 - $O on $b. (using sac_prog 100)
```

MARKING OBJECTS

You have the ability to mark an object as being "owned" by a PC. This field is only viewable by an immortal. It can be used to prevent someone using an object that was not used by them, or just to monitor who was given the object first. It can be used for IC reasons, and it is used for OOC reasons when we suspect abuse of an object is happening.

To set a mark on an object use mposet.

```
>speech_prog p key words go here~
if quest(15, 4, $W) == 8
  mpoload 7501
  mposet i7501 mark $n
  mpgive i7501 $n
  mpmset $W quest 15 4 9
endif
~
```

The following program checks to see if the PC owns the object before allowing them to use it.

```
>wear_prog 100~
if ownsmark($n)
  mpechoat $n {80}You feel the dagger merge with your soul.
else
  mpechoat $n The dagger rejects you as it you are not its soul owner.
  mpechoaround $n The dagger rejects $N.
  mpmadd $n currhp -200
  mplog EVENT: $n has been rejected by a black dagger because $e did not own it.  Vnum
i7501
  mpecho {80}The dagger collapses into a shadow.
  mpjunk i7501 $n
endif
~
|
```

This is what an immortal sees on an object that has been marked when they type ostat.

```
Name: shield faerdale i28406
Vnum: 28406  Type: shield  Count: 1  Gcount: 1  Cost: 5385  Weight: 6
Serial#: 8954  TopIdxSerial#: 8954  TopSerial#: 11571
Short description: a shield of faerdale
Long description: A shield of Faerdale lies here.
Ownership Mark: Mystra
Wear flags : take hold
Extra flags: magic
```

The other thing you can do is log when an object is given away. The deities will see the log as it happens if they are on line. Also we grep the logs every month or two, or when a problem is reported.

```
>give_prog 100~
mplog GIVEAWAY: $n has give a faerdalian shield to $t.
~
```

$t is the recipient of the shield. GIVEAWAY is the logging condition being grepped for. A builder can type help logging on the testport to see all that we log for in the game.

Another way to protect a quest object from being abused, is to place FLAG_UNIQUE on it. This prevents more than one of this object from being worn by a PC.

STRING IF CHECK FUNCTION

The string and stringprefix functions are very powerful code added to the game in 2004 by Tyr. It gives builders alot of power and flexibility in their programs allowing them to check for all sorts of conditions.

And example of how it has been used so far in the game, has been to work out what spell has been cast, so that an intercept program on cast reacts a certain way to different spells.

```
>intercept_prog cast~
if quest(5,5,$n) == 6
  if stringprefix($1) == friends
    mpunintercept
    if string($2) == leperous
    or string($2) == witch
      mpunintercept
      mpmset self flags sentinel
      mpecho $I gasps and looks cautiously at her reflection in the floors.
      mpecho $I doesn't look that much better to you, but seems overjoyed as she scuttles
to your side.
      sayto $n Oh, thank you! For your help,
      sayto $n I will tell you that my husband
      sayto $n is an expert on the knowledge you
      sayto $n seek, but in his current state he
      sayto $n is too sleep deprived to be of any
      sayto $n help to you.
      sayto $n I have a thought on the matter.
      sayto $n Are you willing to listen?
      mpmset $n quest 5 5 7
    else
      mpunintercept
    endif
  else
    mpunintercept
  endif
else
  mpunintercept
endif
~
```

In this prog we wanted to determine if the PC was casting friends. If so they would be doing what the quest wanted of them. $0 is the entire line after the command. $1 is the first argument/word, $2 is the second and so on, up to $9. The

word cast is not processed by the string functions, it is processed by the actual intercept, so it is not part of the equation.

You could also do `cast 'word of recall'`. This would make $0 would be "word of recall", $1 would be "word of recall", $2 would be undefined. Anything inside quotes counts as a single argument where these $# variables are concerned just as in a command like cast.

The difference between string and stringprefix, is that string will require the whole word to be typed in that is being checked for, stringprefix will be satisfied for just a few letters being typed in. This helps builders deal with lazy typists :)

LOADING QUEST MOBS

You want a mob to only appear when a PC is on a quest, and not be in the game all the time.

```
>greet_prog 100~
if mobinroom(13925) == 0
  if quest(0,4,$n) == 0
    if isevil($n)
      mpmload 13925
      mpechoat $n An old bearded human steps out of the forest and hails you.
      mpechoaround $n An old bearded human steps out of the forest and hails $n.
      mpechoat $n He looks all around before whispering to you.
      mpechoaround $n He looks all around before whispering to $n
      mpforce dorgor sayto $n Greetings.
      mpecho The old man glances towards the huge mansion to the north before continuing.
      mpforce dorgor sayto $n Do you have some time to listen to me?
    endif
  endif
endif
~
|
```

The above program is in a room. There is a mobinroom check to make sure that the mobile is not already in the room.

If you wanted the mob to load into a random room, you could put the prog onto a MOBINVIS mobile, and set it to NOWANDER, or STAY_AREA and that will vary the room that the mob is loaded up into.

ALIGNMENT CHECKS

There are several different ways to check alignment. The first way is to check the alignment number. See the mobile alignment lesson for details on the specific numbers. The disadvantage of this method is if you are checking for a specific alignment, old characters from the first year that the game was running may not have that set number. When the game first opened, actions affected alignment. This was later changed to having a chosen alignment that didn't change and a hidden alignment that did.

```
>greet_prog 100~
if align($n) == 1000
or align($n) == 650
or align($n) == 200
  mpecho do things for people who can be accepted
else
  mpecho do things for people who cannot be accepted
endif
~
```

The above checks for the specific numbers which will work for most, but not all characters.

```
>greet_prog 100~
if align($n) < 100
  mpecho do things for people who cannot be accepted
else
  if align($n) > 600
  or align($n) < 300
    mpecho do things for people who can be accepted
  else
    mpecho do things for people who cannot be accepted
  endif
endif
~
|
```

The above checks for a range of alignments.

There is a simpler way to check alignments by using the if isevil, if isgood checks etc.

```
if questr(16500,0,5,$n) == 3
  sayto $n So, you have come to me to be tested. Very
  sayto $n well, let's see if you are worthy of the
  sayto $n City Watch.
  mpechoat $n $I prays as he makes strange gestures toward you.
  mpechoaround $n $I prays as he makes strange gestures towards $N.
  if islawful($n)
    if isgood($n)
    or isneutral($n)
      sayto $n Very good, you possess the qualities
      sayto $n we seek for Watch members.
      sayto $n Please return to the Headmaster.
      mpmset $n questr 16500 0 5 4
    else
      sayto $n I am sorry, but you do not
      sayto $n have the qualities we are
      sayto $n seeking.
      sayto $n Please return to the Headmaster.
      mpmset $n questr 16500 0 5 5
    endif
  else
    sayto $n I am sorry, but you do not
    sayto $n have the qualities we are
    sayto $n seeking.
    sayto $n Please return to the Headmaster.
    mpmset $n questr 16500 0 5 5
  endif
endif
~
```

The above needed to know if the PC was either Lawful Good or Lawful Neutral in alignment. First we checked to see if the PC was lawful, and the to see if they were good or neutral.

The available checks for this are: isgood, isneutral, isevil, islawful, ischaotic, isunconcerned. When you combine these checks you can determine the alignment of a PC. To determine True Neutral you need to combine isneutral and isunconcerned.

GREED PROGRAMS

You have a mobile that sells a variety of very good items. Sometimes, he will only allow the PC to shop with him at the end of a hard quest, but you do not want the PC to buy every item on offer, and more importantly buy every item for all his friends. You would like to limit the PC to choosing just one or two of the items on offer. You can do this with an intercept_prog on buy, and a buy prog on each of the vnums for sale that sets up a quest bit on the player. There is no real way to do this without using quest bits.

The sample below is for a mobile that sells special shields. He wants the PC to only choose one of his shields and then he will sell no more of them.

```
>intercept_prog buy~
if quest(0,1,$n) == 1
  sayto $n No more shields for you.
else
  mpunintercept
endif
~
>buy_prog i17227~
mpmset $n quest 0 1 1
~
>buy_prog i17226~
mpmset $n quest 0 1 1
~
>buy_prog i17225~
mpmset $n quest 0 1 1
~
>buy_prog i17224~
mpmset $n quest 0 1 1
~
>buy_prog i17223~
mpmset $n quest 0 1 1
~
>buy_prog i17222~
mpmset $n quest 0 1 1
~
>buy_prog i17221~
mpmset $n quest 0 1 1
~
```

First off the intercept_prog on buy checks to make sure that the quest bit has not been reached for purchasing at the shop. If it has he tells the PC's he has no more stock for them. If it hasn't then mpunintercept allows them to buy the item. If they do not have enough money, because of the way this has been done, the quest bit will not be set up.

For each item that the mobile sells that the mobile only wants to sell one of, there is a buy prog. When the item is purchased a quest bit is set up on the PC. The PC buys the item normally. If you wanted, you could have the mobile say something about the qualities of the item.

CHARGED MAGICAL OBJECTS

Value5 on an object is not used by any of the objects to set the fields specific to the ITEM_TYPE's. This makes it ideal to use in magical objects that you want to have a limited life or limited charges. You can modify value5 with each use of the object, and then when value5 of the object reaches a certain point, then the item has run out of magical charges, or it is destroyed. This allows builders to be able to put some rather powerful magical items into the game without unbalancing it.

Below is a sample of a ring, that when a command word is typed by the PC allows the PC to fly. If the ring has charges then the PC will fly.

```
>intercept_prog ringcommandword~
if wear_loc($o) != -1
  if objval5($o) == 0
    cast 'fly' $n
    mposet on $n iVNUM value5 1
    mpechoat $n You rise in the air.
    mpechoaround $n $N rises in the air.
  else
    mpechoat $n Nothing happens.
    mpechoat $n Perhaps the ring needs to be recharged?
  endif
else
  mpechoat $n You should wear the ring first.
endif
~
>time_prog 10~
if day() == 20
  if objval5($o) != 0
    mpechoat $n Your ring glows briefly as it is recharged.
    mposet on $n iVNUM value5 0
  endif
endif
~
```

If you want the user to know whether the ring is charged or not, you could add an exa_prog to the ring.

```
>exa_prog 100~
if objval5($o) == 0
  mpechoat $n The ring glows faintly.
else
  mpechoat $n The ring does not glow.
endif
~
```

The echoes in the exa_prog will appear after the extra description of the ring, if it has any.

Here is an example for a ring that would enable the user to cast fly three times per day.

In this case, value5 is the number of charges left on the ring (it needs to be set to 3 in the object description if you want the ring to be fully charged when it is loaded). The object is recharged each day at 10am.

```
>intercept_prog ringcommandword~
if wear_loc($o) != -1
  if objval5($o) == 0
    mpechoat $n Nothing happens.
    mpechoat $n Perhaps the ring needs to be recharged?
  else
    cast 'fly' $n
    mpoadd on $n iVNUM value5 -1
    mpechoat $n You rise in the air.
    mpechoaround $n $N rises in the air.
  endif
else
  mpechoat $n You should wear the ring first.
endif
~
>time_prog 10~
if objval5($o) < 3
  mpechoat $n Your ring glows briefly as it is recharged.
  mposet on $n iVNUM value5 3
endif
~
```

Note that if the PC has the item in a container, then the time prog will not trigger for the object and the item will never recharge.

IMBRICATED RAND PROGS

The following lesson was written by Dalvyn after he was exploring the use of 'if rand(%)' to make the actions of mobs less predictable.

First let us look at this example fight prog on a wizard:

```
>fight_prog 100~
if rand(10)
  cast 'stone skin'
else
  if rand(50)
    cast 'magic missile' $n
  else
    if rand(10)
      cast 'shield'
    else
      if rand(10)
        cast 'armor'
      else
        if rand(10)
          cast 'bulls strength'
        else
          if rand(10)
            cast 'fire shield'
          else
            cast 'shocking grasp' $n
          endif
        endif
      endif
    endif
  endif
endif
~
```

Is there any reason this might be a bad idea? If I had the patience, I could make mobs do some neat stuff with code like this.

The answer below is a rather long and boring explanation for those who want to know the details. If you just want a balanced program (where you have 3, 4, 5, ... options all of them with the same chance of happening) and do not want the details, the answer is further below.

All classes come with a standard fight_prog I think. But if you really want your mob to be efficient and if you don't want it to just do some standard actions / cast some standard spells, I think it's a very good idea to add a fight_prog with some random actions.

A note about imbricated rand tests, that is, rand tests in other rand tests... you have to remember that each "rand" test is like a choice with 2 options: the "if" part and the "else" part. I saw in your program above that you have rand(..) tests with numbers summing up to 100%. This program above won't make the mob cast "magic missile" with 50% chance, or "fire shield" with 10% chance though... it's a bit more tricky.

I'll use the (smaller) program below to explain

```
if rand(40)
  cast 'magic missile'
else
  if rand(50)
    cast 'lightning bolt'
  else
     cast 'shocking grasp'
  endif
endif
```

In this program, the first test is "if rand(40)". That means that there is 40% chance that the mob casts 'magic missile' and thus, 60% chance that the "else" part is used. In this else part, there is 50% chance that the mob casts "lightning bolt" and, if not, the mob casts "shocking grasp". So, if you want to sum it up with a small diagram...

```
      /-- 40% -- magic missile
      |
------|
      |                            /-- 50% -- lightning bolt
      |                            |
      \-- 60% -- first else part ----|
                                   |
                                   \-- 50% -- shocking grasp
```

That means that, when this fight_prog is triggered,

- there is 40% chance that the mob casts 'magic missile'
- and 60% chance that the mob does something else.

In these 60%, there are 2 options:

- 50% chance to cast lightning bolt
- 50% chance to cast shocking grasp

```
/----------------------|-----------------------------\
|       40 %           |           60 %              |
\----------------------|-----------------------------/
    magic missile     /---------------|-------------\
                      |    50 %       |    50 %     |
                      \---------------|-------------/
                       lightning bolt shocking grasp
```

So, finally, there is

- 40% chance for magic missile
- 30% chance for lightning bolt (50% of 60%)
- 30% chance for shocking grasp (the last 50% of 60%)

In your longer program above, you would have

- 10% for stone skin (90% left for the else part)
- 45% (50% of 90%) for magic missile (45% left for the second else part)
- 4.5% (10% of 45%) for shield (40.5% left for the third else part)
- 4.05% (10% of 40.5%) for armor (36.45% left for the fourth else part)
- 3.645% (10% of 36.45%) for bulls strength (32.803% left for the fifth else part)
- 3.2803% (10% of 32.803%) for fire shield (29.5247% left for last else part)
- 29.5247% for shocking grasp

All that sums up to "the numbers in rand() tests can be tricky".

For example, if you want a rand_prog with 4 options that all have the same chance of happening, you can't use:

```
if rand(25)
  option 1
else
  if rand(25)
    option 2
```

```
    else
      if rand(25)
        option 3
      else
        option 4
      endif
    endif
  endif
```

Because that would make

- 25% for option 1 (75% left)
- 18.75% (25% of 75%) for option 2 (56.25% left)
- 14.0625% (25% of 56.25%) for option 3 (42.1875% left)
- 42.1875% for option 4

So, option 4 is very likely to be be chosen in this case. What you want is 25% for each of those options.

- So, rand(25) for option 1, and there's 75% left
- You want 25%, that is, 1/3rd of what is left, so you'll use rand(33) for option 2, then there's 50% left
- You want 25%, that is, 1/2 of what is left, so you'll use rand(50) for option 3

```
if rand(25)
  option 1
else
  if rand(33)
    option 2
  else
    if rand(50)
      option 3
    else
      option 4
    endif
  endif
endif
```

That is not too surprising actually... option 1 is a one-in-four chance of happening, so 25%. If option 1 is NOT chosen, you only have 3 options left... so option 2 is a one-in-three chance of happening, rand(33). Then, if neither option1 nor option2 is chosen, you have only 2 options left, so option 3 is rand(50).

For 2 options with the same chance to happen:

```
if rand(50)
  option1
else
  option2
endif
```

For 3 options with the same chance to happen:

```
if rand(33)
  option1
else
  if rand(50)
    option2
  else
    option3
  endif
endif
```

For 4 options with the same chance to happen:

```
if rand(25)
  option1
else
  if rand(33)
    option 2
  else
    if rand(50)
      option3
    else
      option4
    endif
  endif
endif
```

The numbers for the rand checks are (from the most inner one to the most outer one): 50, 33, 25, 20, 17, 14, 12, ...

The easiest way to simulate delay is to use a rand_prog. You would have to choose the parameter of the rang_prog based on several things, such as:

- how long you want the delay to be
- how many characters would be in the room at the same time (the more characters there are, the longer it will take for the rand_prog to trigger for the character who has the right quest bits).

Here is an example for a mobprog (but, depending on what you want to do, the rand_prog might be on the room or on an object).

```
>greet_prog 100~
if quest(0,5,$n) == 7
  mpechoat $n {70}The mob looks at you thoughtfully and remains silent for a few seconds.
  mpechoaround $n {70}The mob looks at $N thoughtfully and remains silent for a few
seconds.
  mpmset $n quest 0 5 8
endif
~
>rand_prog 60~
if quest(0,5,$r) == 8
  sayto $r I am done thinking and I will start
  sayto $r talking with you now.
  ...
endif
~
```

Just remember to use $r instead of $n in the rand_prog.

CONNECT COMMAND

You can connect and disconnect room exits in a mob/room/object program as follows:

```
connect east 8765
```

This will remove the connection east (if there is one) and make a connection east to the room 8765.

```
connect east
```

This will simply remove the connection east (if there is one).

```
connect east 8765 3 -1 door
```

This will remove the connection east (if there is one) and make a connection east to the room 8765 with a closed door that has the keyword 'door'.

```
connect east 8765 7 8766 gate
```

This will remove the connection east (if there is one) and make a connection east to the room 8766 with a closed and locked door that has the keyword 'gate' and can be unlocked with the key vnum 8766

For a "somewhere" connection, you would use this.

```
connect +somewhere roomvnum 524288 keyword
```

And to remove this connection,

```
connect somewhere
```

SAMPLE OBJECT WEAPONS

Non magical weapon

Below is a dagger that is not magical in any way. It can be hung on the belt (note that it will take up all of the belt layers as weapons cannot be layered.) It is tiny in size, and it is made of silver. Its quality is high, and it is perfect condtion. It is a pink colour because of the hilt.

```
#8005
jewelled ladies dagger~
{D0}a Jewelled Ladies Dagger~
{D0}A Jewelled Ladies Dagger lies here. ~
~
ITEM_TYPE_WEAPON
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD|CAN_WEAR_BELT
QUALITY_HIGH MATERIAL_SILVER COND_PERFECT SIZE_TINY
0 0 0 WEAPON_TYPE_DAGGER 0 0
E
jewelled ladies dagger~
{70}It is a small silver dagger with a small hilt, so even the smallest
ladies hand can grasp it with ease.  {D0}The hilt is inlaid with fine jewels.
~
```

Magical weapon

The longsword below is what would be considered a +1 longsword. It has an apply to it to add 1 damroll and 1 hitroll to it. It also has a the magic flag on it so that when detect magic is cast, it can be seen that the item is magical.

```
#8247
brilliant looking longsword sword~
{70}a brilliant looking longsword~
{70}A brilliant looking longsword lies here.~
~
ITEM_TYPE_WEAPON
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD|CAN_WEAR_BELT
QUALITY_AVERAGE MATERIAL_STEEL COND_PERFECT SIZE_MEDIUM
0 0 0 WEAPON_TYPE_LONG_SWORD 0 0
A APPLY_DAMROLL 1
```

```
A APPLY_HITROLL 1
E
brilliant looking longsword sword~
This longsword is made of steel, and has engravings up the blade.
~
```

The following weapon uses the specialised weapon flags.

```
#QQ30
dagger runed rune green speed~
{A0}a green-runed {70}dagger~
{A0}A green-runed {70}dagger lies here. ~
~
ITEM_TYPE_WEAPON
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD|CAN_WEAR_BELT
QUALITY_HIGH MATERIAL_SILVER COND_PERFECT SIZE_TINY
0 WFLAG_SPEED 0 WEAPON_TYPE_DAGGER 0 0
E
green runed rune dagger~
{A0}The runes have a slight glow to them.
~
```

SAMPLE OBJECT - ARMOUR

The doublet below is made of silk and is type cloth. It will not affect a mages spell casting abilities. It will also not hold up well in battle. It is worn on the body in the armour location.

```
#8255
green silk doublet~
{A0}a green silk doublet~
{A0}A green silk doublet lies on the ground here.~
~
ITEM_TYPE_ARMOR
0
CAN_WEAR_TAKE|CAN_WEAR_BODY
QUALITY_HIGH MATERIAL_SILK COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_CLOTH 0 0
```

The shield below is made of wood. Note that the armour type is shield.

```
#8434
diamond-shaped large hide kite shield~
{30}a hide kite shield~
{30}A large diamond-shaped shield lies here.~
~
ITEM_TYPE_ARMOR
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_WOOD COND_PERFECT SIZE_LARGE
0 0 LAYER_ARMOR ARMOR_TYPE_SHIELD 0 0
E
hide kit shield~
It is a shield nearly as tall as a man, formed of hides
stretched over a canvas frame and further reinforced by
several steel staples lined up vertically on the face of
the shield.
~
```

SAMPLE OBJECTS - TREASURE

Treasure can be layered and it can be worn in any location. It can be damaged. Type armor protects better than treasure. Treasure is reserved for jewellry and crowns and the like. They layer value is different for treasure than armour. It is in Value4.

```
#8080
heavy gold signet ring gold~
{B0}a heavy gold Signet Ring~
{B0}A ring of gold lies here.~
~
ITEM_TYPE_TREASURE
0
CAN_WEAR_TAKE|CAN_WEAR_FINGER
QUALITY_SUPERIOR MATERIAL_GOLD COND_PERFECT SIZE_MEDIUM
0 0 0 0 LAYER_ALL 0
E
heavy signet ring gold~
It is a gold ring.
~
```

Note the transparent flag on the belt buckle below. Because it is worn on the belt, you want all of the items to be seen. If there is no transparent flag, then the items that layer below the buckle would not be seen when the character is looked at. Transparent flag allows for items beneath an item to be seen.

```
#8242
brass buckle~
{30}a brass buckle~
{30}A brass buckle catches your eye.~
~
ITEM_TYPE_TREASURE
FLAG_TRANSPARENT
CAN_WEAR_TAKE|CAN_WEAR_BELT
QUALITY_HIGH MATERIAL_BRASS COND_PERFECT SIZE_MEDIUM
0 0 0 0 LAYER_OVER 0
```

SAMPLE OBJECTS - LIGHTS

Lights can be worn in any location and they can be layered. It must make IC sense for an item to be a light if its not in the normal hold or float locations. Lights are lit automatically when held by the player. The light below will last for 40 game hours before going out. When a light goes out it junks.

```
#8546
torch~
{B0}a torch~
{30}A torch lies on the ground here.~
~
ITEM_TYPE_LIGHT
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_WOOD COND_PERFECT SIZE_SMALL
0 0 40 0 0 0
E
torch~
It is a bundle of dried sticks tied together with the end
soaked in pitch.
~
```

The light below floats around the users head. It will last 100 game hours before going out. Because it is floating, it would be considered to have some sort of magic to allow it to do so, so we suggest in most instance that it should have a magic flag.

```
#8100
brightly coloured ball light~
{B0}a {C0}coloured {D0}ball {E0}of {F0}light~
{B0}A brightly coloured ball of light floats here.~
~
ITEM_TYPE_LIGHT
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_FLOATING
QUALITY_AVERAGE MATERIAL_ENERGY COND_PERFECT SIZE_MEDIUM
0 0 100 0 0 0
```

Notes

Lights will only work in the inventory of a PC. If you wish to have a light that works in the room (ie like a torch on the wall) then use ITEM_TYPE_FIRE.

SAMPLE OBJECTS - FOUNTAINS

In order to make sure a fountain has sufficent drinks and skin refills over the reboot time of the game, give value0 and value1 large numbers. Most fountains should not be able to be picked up, so make sure not to put any wear flags on them.

```
#8003
water fountain~
{C0}a water fountain~
{C0}A fountain spills crystal clear water from a dolphin's mouth here. ~
~
ITEM_TYPE_FOUNTAIN
0
0
QUALITY_AVERAGE MATERIAL_MARBLE COND_PERFECT SIZE_MEDIUM
100000 100000 0 0 0 0
E
water fountain~
{C0}The fountain is a marvellous sculpture with a sea theme.  Dolphins
cavort happily, and one has water flowing from its mouth.
~
```

SAMPLE OBJECTS - DRINKS AND FOOD

Drinks

The skin below can hold up to 5 servings of liquid. It currently has 4 servings. It holds water that is not affected by a herb/component (NOT_POISONED) and after it is empty it will not junk. If you want the container to junk after drinking then set value4 to 0. When working out what material to make a drink, make it the material of the container rather than the liquid within.

```
#8033
travellers water skin waterskin~
{30}a travellers water skin~
{30}A travellers water skin leaks water onto the ground here. ~
~
ITEM_TYPE_DRINK_CON
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD|CAN_WEAR_BELT
QUALITY_AVERAGE MATERIAL_LEATHER COND_PERFECT SIZE_MEDIUM
5 4 LIQ_WATER NOT_POISONED 1 0
E
travellers water skin~
Made of animal skin, it holds a good supply of water for
the hardy traveller.
~
```

The drink below is affected by a herb.

```
#11022
vial xorn saliva~
{30}a vial of xorn saliva~
{30}A vial of xorn saliva has been left here.~
~
ITEM_TYPE_DRINK_CON
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_GLASS COND_PERFECT SIZE_MEDIUM
1 1 LIQ_WATER HERB_MONKSHOOD 0 0
```

Food

Value0 on food determines how many hours the food will fill the stomach for. When trying to work out this value, think about how long that food would last you if you were to eat it in real life before you got hungry again. For instance small items like an apple should only have a value0 of 1, and large hearty stews might be good for 4 hours. Anything higher than 4 hours should be rare or magical.

Value1 on food is a timer for how long the food lasts after it is bought. This value would be higher for preserved foods, and much lower for fresh foods. Once the timer is reached the food decays. Fresh food can last up to 6 hours without the aid of magic.

Value2 on food is used to set if the food is raw or cooked. If it is raw, then the PC can cook it. The default value of 0 is raw. Many foods in the game will show up as raw until all areas are revised with this new code.

Value3 on food sets the herb/component that the food might be affected by. A value of 0 or NOT_POISONED means the food is not affected by any herbs.

```
#8032
travellers rations~
{30}travellers rations~
{30}A packet of travellers rations is moulding on the ground here. ~
~
ITEM_TYPE_FOOD
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_MEDIUM
5 100 FOOD_COOKED NOT_POISONED 0 0
E
travellers rations packet~
Tightly bound in a waterproof hide, are some dried rations suitable
for the intrepid traveller.
~
```

The food below offers no nourishment at all, but it will affect the eater with a herbal affect.

```
#8594
powder caffe~
{80}some caffe powder~
{80}Some caffe powder lies here.~
~
ITEM_TYPE_FOOD
0
```

```
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_TINY
0 0 0 HERB_HENBANE 0 0
```

SAMPLE OBJECTS - CONTAINERS

There are a couple of containers that you cannot make because of the way they are hard coded.

- Spell Pouch - If you want to sell spell pouches in your area, then you will need to use vnum 68. You can rename the spellpouch so that it can be different colours etc. See the lesson on restringing generic objects. If you do make a spellpouch, it will not work like the hard coded one, that allows players to store components in the pouch and cast spells without having to take the components out and put them into inventory.
- Money Pouch - The hard coded money pouch allows players to store money in the money pouch and not have to take the money out when paying for goods. Again this can be restrung with programs if you wish to fancy it up. Use vnum 8132 if you want to sell money pouches in your area.

The container below can be closed, but it cannot be locked. When bought it is open. It can hold 100 pounds worth of items. Containers that can hold more than 100 pounds worth of items should be rare. Value0 determines the carrying capacity of the container. Note that the layer value is not the same as for armour. Also the container is flagged transparent so when it is worn on the body, it does not hide the armour.

```
#8036
travellers pack~
{30}a travellers pack~
{30}A travellers pack lies here. ~
~
ITEM_TYPE_CONTAINER
FLAG_TRANSPARENT
CAN_WEAR_TAKE|CAN_WEAR_BODY
QUALITY_AVERAGE MATERIAL_LEATHER COND_PERFECT SIZE_MEDIUM
100 CONT_CLOSEABLE -1 0 LAYER_OVER 0
E
travellers pack~
It is a pack that one would use to put things in before
travelling.
~
```

If you want any kind of armour to have pockets then you need to make it type container. It will not protect as well however. Make sure the carrying capacity reflects the kind of container it is. For instance this apron holds far less than the pack above. The PC has a choice as to if they will wear it on their waist or on their legs. Very few would wear this into battle, but there are plenty of roleplay opportunites for an item like this with trades etc. Because an apron would not wrap all the way around, it is flagged transparent so what is worn underneath it can be seen. Although, if it is worn at the waist, the layer aspect will not work as the waist location cannot be layered.

```
#8267
apron with pockets~
{70}an apron with pockets~
{70}An apron lies here.~
~
ITEM_TYPE_CONTAINER
FLAG_TRANSPARENT
CAN_WEAR_TAKE|CAN_WEAR_WAIST|CAN_WEAR_LEGS
QUALITY_AVERAGE MATERIAL_CLOTH COND_PERFECT SIZE_MEDIUM
20 0 0 0 LAYER_OVER 0
E
apron pockets~
This apron is made from a plain white cloth with
two large pockets.
~
```

SAMPLE OBJECTS - SCROLLS AND POTIONS

Do NOT make recall scrolls and potions. Use Vnums 50 and 51 in your area. Also check the free objects list to see if the potion with the spell you want is not already available in that list.

Potions

In order for a potion to be reused with the brew skill you need to follow a set naming convention. Below are two potions that demonstrate the convention. You must first describe the potion container. For instance the ones below are glass jars. After the container then you will state the spell that is within it. This means once quaffed what is left is just a yellow glass jar for the first, and a pink glass jar for the second. The jars are empty. The jars can then be used by characters who use the brew skill. The main thing that the code looks for is the word "of". What is before the of will be what remains after the potion is quaffed, and what is after the of, including the of, will be removed by the code after the potion is quaffed.

You can put up to three spells in a potion. The level of the spell and the number of spells will determine the value of the potion. Potions are magical items so be sure to put a magic flag on them. This magic flag does last even when the potion jar is empty. In order to drink a potion a player must hold it so you must put CAN_WEAR_HOLD on the object.

```
#8549
potion invisibility yellow glass jar~
{B0}a yellow glass jar of invisibility~
{B0}A yellow glass jar lies here.~
~
ITEM_TYPE_POTION
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_GLASS COND_PERFECT SIZE_SMALL
30 SPELL_INVIS SPELL_NONE SPELL_NONE 0 0
E
yellow glass jar~
It is made of glass that is as yellow as the sun.
~
#8550
potion strength pink glass jar~
{D0}a pink glass jar of strength~
{D0}A pink glass jar lies here.~
~
ITEM_TYPE_POTION
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_GLASS COND_PERFECT SIZE_SMALL
```

```
30 SPELL_STRENGTH SPELL_NONE SPELL_NONE 0 0
E
pink glass jar~
The glass is a pretty pink and it is shaped like a flower.
~
```

Scrolls

As with potions, scrolls can have up to three spells on them. In order to recite a scroll, a player must be able to hold it, so make sure to put the CAN_WEAR_HOLD flag on the object. Scrolls are magical items, so should be flagged as magic.

```
#8039
wizardly scroll protection~
{70}scroll of protection~
{70}A wizardly scroll of protection lies abandoned here. ~
~
ITEM_TYPE_SCROLL
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
30 SPELL_PROTECTION SPELL_NONE SPELL_NONE 0 0
E
wizardly scroll protection~
It's a scroll that contains a spell that will help protect you.
~
```

SAMPLE OBJECTS - WANDS AND STAVES

Because of the high cost of wands and staves, you must be very careful as to how you place them in your area. Do not put them on a mobile that can be killed over and over, and therefore will drop many. This will lead to an extremely quick and easy profit for the player. In general our aim is that when not for sale in shops, wands and staves can only ever be obtained once freely by the PC.

Wands

Wands can only have one spell in them. The wand below has 6 charges out of 6 of the spell magic missile. If left at 0 charges then it will work the once before disintergrating. Wands are magic items and should have a magic flag on them.

```
#8141
wand magic missiles pinkish~
{D0}a wand of magic missiles~
{D0}A pinkish coloured wand lies on the ground here.~
~
ITEM_TYPE_WAND
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_WOOD COND_PERFECT SIZE_MEDIUM
25 6 6 SPELL_MAGIC_MISSILE 0 0
E
wand magic missiles pinkish~
It is a straight wooden stick that glows with a pinkish aura.
~
```

Staves

Staves can only have one spell on them, but they can have several charges of that spell, and can also be recharged. Once the staff gets to 0 charges it will work once more before disintegrating. Staves are magic items and should have a magic flag on them.

```
#8279
staff pigs head create food~
{30}a staff with a pigs head on top~
{30}A staff with a pigs head on top lies here.~
~
ITEM_TYPE_STAFF
```

```
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_WOOD COND_PERFECT SIZE_MEDIUM
25 10 5 SPELL_CREATE_FOOD 0 0
```

SAMPLE OBJECTS - PILLS AND SALVES

Pills are eaten, and salves are applied. The following are samples of each.

```
#QQ48
fireweed herb balm~
{90}some fireweed herb balm~
{90}Some fireweed herb balm spills over here.~
~
ITEM_TYPE_SALVE
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_MEDIUM
10 3 0 HERB_BLOODROOT SPELL_CURE_LIGHT 0
E
fireweed herb balm~
{90}This balm is in a little pot and is red in colour.
~
```

The salve above will work three times. The spell is level 10. It also has the herbal affect of bloodroot. The thorn below when eaten, will do a level 40 spell of cure light on the eater.

```
#QQ62
golden thorn~
{B0}golden thorn~
{B0}A Golden Thorn plant grows here.~
~
ITEM_TYPE_PILL
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_MEDIUM
40 SPELL_CURE_LIGHT SPELL_NONE SPELL_NONE 0 0
E
golden thorn~
It is an exotic looking plant with thorns that look like they
are made of gold.  It is said to have magical properties.
~
```

SAMPLE OBJECTS - FURNITURE AND TRASH

Trash

Because an item is trash doesn't mean it should be discarded. The ITEM_TYPE_TRASH is often used for items that don't fit in as any other type. Applies and wear locations can be given to trash as with many of the other item types. You may find that you have to apply value as well when the item you have made has value in the game but the fact that it is type trash has lowered the value in the game. Trash items default to 1 copper in the shops.

```
#8320
waterdeep newspaper waterdeepnews~
{70}the waterdeep news paper~
{70}A newspaper flutters about here.~
~
ITEM_TYPE_TRASH
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
0 0 0 0 0 0
E
paper waterdeep news~
It is a small paper printed with small print.
~
>exa_prog 100~
mpechoat $n You take a look at the Newspaper with interest.
mpechoaround $n $n takes a look at the Newspaper with some interest.
if rand(30)
  mpforce $n help waterdeepnews
else
  mpforce $n help waterdeepnews2
endif
~
|
```

Furniture

Furniture is usually reserved for items that are not takable and not fountains. Remember that chests need to be containers.

```
#8614
```

```
large banner~
{E0}a large banner~
{E0}A large banner hangs from the gate towers.~
~
ITEM_TYPE_FURNITURE
0
0
QUALITY_AVERAGE MATERIAL_CLOTH COND_PERFECT SIZE_MEDIUM
0 0 0 0 0 0
>exa_prog 100~
mpforce $n help waterdeepbannerone
~
|
```

If furniture is either a bed, chair, lecturn or altar, it can be set in the code so PC's can interact with it. See the Furniture Flags Listing for more information. Value2 is set for the kind of furniture that it is.

```
#4232
altar golden torm~
{B0}the golden altar~
{B0}A golden altar to Torm glows here.~
~
ITEM_TYPE_FURNITURE
FLAG_GLOW|FLAG_MAGIC
0
QUALITY_AVERAGE MATERIAL_GOLD COND_PERFECT SIZE_MEDIUM
0 0 FURNITURE_ALTAR 0 0 0
E
altar golden torm~
A well crafted altar dedicated to Torm.
~
```

SAMPLE OBJECTS - KEYS

Keys do not save if left in inventory when a player quits. They will only save if the character puts the key into a container. Also there is a chance of the key breaking etc when being used. In order for a door to be picked it must have a key, even if they key can never be found. Note that when setting the key vnum in the door information, the vnum does not have to be type key.

```
#8076
dungeon key~
{80}a dungeon key~
{80}A dungeon key lies here. ~
~
ITEM_TYPE_KEY
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_WORTHLESS MATERIAL_IRON COND_PERFECT SIZE_MEDIUM
0 0 0 0 0 0
E
dungeon key~
This key is made of iron.  It looks like it would
fit the dungeon door.
~
```

SAMPLE OBJECTS - BOOKS

There are essentially two types of books in the game. The kind that players read, that bards have written to be published, and the kind that wizards can learn new spells from. Below is samples of both kinds. Please note that in general the books that bards write are stored in the Waterdeep Area file so that the books can be sold in different locations around the game.

Published books for reading

If you are wanting to do this kind of book in your area, check with the builders administrators first. If approved follow the format below. We like to have books have the same standards across the game, so that players know how to read a book when they pick it up.

```
#8207
book cale sune sunes sune's mercy~
{70}a book - "A Song of Sune's Mercy" by Cale~
{70}The book "A Song of Sune's Mercy" lies here.~
~
ITEM_TYPE_BOOK
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
0 0 0 LANG_COMMON 25 0
A APPLY_VALUE -3000
E
book~
{F0}A Song of Sune's Mercy
  by Cale Silvermoon of the Starry Quill.
Chapter's {90}ONE{F0} up to Chapter {90}TWO
{F0}Type look chapter number to read the book. ie {90}look ONE
~
E
one~
A Song Of Sune's Mercy
There once lived an Elf maiden, surpassingly fair
Golden her skin and flaxen her hair
Her smile was sunlight and raindrops her sigh
Her beauty was such to ensorcell the eye.
As she walked through bright Tethyr, one soft summer's day
A young human Ranger chanced to cross her way
From that first brief meeting, his spirit was bound
His reason for being, her laughter's sweet sound.
To win her passion was then his life's goal
```

```
To gain her love he would give up his soul
At last she was moved by his devotion and love
And they declared their lives joined, blessed by Sune, above
~
E
two~
Time passed softly, sweetly, like autumn leaves falling
And alas, all too soon cruel time began calling
The young Ranger's laughter now was silent and gone
his brief human life had at last moved on.
With passion and tears, the Elf begged for Sune's power
to birth him again, for love still to flower
Sune smiled on them, and took them both to her heart
They dwell with her still, no more ever to part.
- written by my hand.
  Cale Silvermoon, A Bard of Mystra.
~
```

Magic spell books

Wizard characters can copy spells in books into their spellbook. It is a good quest reward to give out for rare and hard to find spells. This kind of book should be flagged as magic.

```
#8281
dusty red book spellbook disintegrate~
{90}a dusty red spellbook~
{90}A dusty red spellbook lies here with its pages bared here.~
~
ITEM_TYPE_BOOK
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_SUPERIOR MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
0 SPELL_DISINTEGRATE LANG_ELVEN 25 0 0
A APPLY_VALUE 10000
>exa_prog 100~
if class($n) == Wizard
  if int($n) > 15
    mpechoat $n You make out the word disintegrate on the top of a page.
    mpechoaround $n $n opens a spellbook and examines it.
    mpechoaround $n $n gives a smile of satisfaction as $e reads the runes.
  else
    mpechoat $n You cant quite make out the runes.
    mpechoaround $n $n opens a spellbook and examines it.
```

```
    mpechoaround $n $n has a look of frustration on $s face.
  endif
else
  mpechoat $n The runes on the book seem to swim before your eyes.
  mpechoaround $n $n opens a spellbook and examines it.
  mpechoaround $n $n has a look of total incomprehension on $s face.
endif
~
|
```

SAMPLE OBJECTS - INSTRUMENTS

Not all instruments have to be magical. Bards will use them just for roleplay, to sing with and play etc. Make sure to flag magic those that have magical affects to them.

Magical Instrument

```
#8305
silver stringed harp~
{F0}a silver stringed harp~
{F0}A silver stringed harp lies here.~
~
ITEM_TYPE_INSTRUMENT
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_SILVER COND_PERFECT SIZE_MEDIUM
40 4 4 SPELL_CHARM_PERSON 0 0
```

Non-magical Instrument

Note that by hard code, only bards are able to use the play command to activate an instrument. If you are wanting to allow someone else to play it then you will need to put a program on the harp like the one below to make it happen.

```
#8535
mouth harp~
{30}a mouth harp~
{30}A small instrument of some type lies here.~
~
ITEM_TYPE_INSTRUMENT
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_SOFTWOOD COND_PERFECT SIZE_TINY
0 0 0 0 0 0
E
mouth harp~
It's a piece of wood with several stiff strips of metal
projecting from it.  They can be plucked to make a sort
of music.
~
```

```
intercept_prog play~
if wear_loc($o) != -1
  if guild($n) == Bards
    mpechoat $n You play a lively tune on the mouth harp.
    mpechoaround $n $n plays a lively tune on the mouth harp.
  else
    if guild($n) == Rangers
      mpechoat $n You play a woodsy tune on the mouth harp.
      mpechoaround $n $n plays a woodsy tune on the mouth harp.
    else
      mpechoat $n You attempt to play a tune on a mouth harp but sound like a cat in
heat.
      mpechoaround $n $n attempts to play a tune on a mouth harp but sounds like a cat in
heat.
    endif
  endif
else
  mpechoat $n You need to hold the mouth harp in order to play it.
endif
~
|
```

SAMPLE OBJECTS - QUIVERS

Value0 on the quiver sets how much it can hold. Keep this number fairly low. If the quiver is made to be worn over other things, make sure to flag it transparent.

```
#8217
buckskin quiver~
{30}a buckskin quiver~
{30}A buckskin quiver lies on the ground here.~
~
ITEM_TYPE_QUIVER
FLAG_TRANSPARENT
CAN_WEAR_TAKE|CAN_WEAR_BODY
QUALITY_AVERAGE MATERIAL_HIDE COND_PERFECT SIZE_MEDIUM
20 0 0 0 LAYER_OVER 0
E
buckskin quiver~
Made of a heavy buckskin this quiver can contain upto
20 normal arrows.
~
```

SAMPLE OBJECTS - RANGED WEAPONS AND PROJECTILES

There are two types of ranged weapons. The first being those that double as hand to hand weapons as well. Small axes and daggers can be thrown as well. These are denoted with a green asterik (*) on the Weapon Types Listing. This lesson however covers the second type of ranged weapon. This type needs projectiles/ammunition to be used. This includes weapons like bows and crossbows. It is a good idea when making a ranged weapon in your area to also have the projectiles that are used in the weapon available too.

```
#8476
birch bow longbow~
{30}a birch longbow~
{30}A birch longbow lies here.~
~
ITEM_TYPE_WEAPON
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_HARDWOOD COND_PERFECT SIZE_LARGE
0 0 0 WEAPON_TYPE_LONG_BOW 0 0
E
longbow~
It is a long bow, as tall as a tall man.  It is simply carved,
and uses animal gut for a bowstring.
~
```

A projectile needs to have the weapon type set that it will work with. Crossbow bolts will not work in longbows for instance.

```
#8663
birch arrow~
{30}a birch arrow~
{30}A birch arrow lies here.~
~
ITEM_TYPE_PROJECTILE
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_HIGH MATERIAL_WOOD COND_PERFECT SIZE_MEDIUM
0 0 0 WEAPON_TYPE_LONG_BOW 0 0
E
birch arrow~
```

```
A well crafted birch arrow.
{00}.{30}        _.-"/_____________________/////
{00}.{80} jgs   {30} `'-.\^^^^^^^^^^^^^^^^^^^^^^\\\\\
{00}.{30}
~
```

SAMPLE OBJECTS - SHEATHS

Sheaths can only hold one object in them, a weapons. And value0 is the maximum weight of the weapon that the sheath can hold. For smaller sheaths that hold daggers keep that value small, as in the example below. For sheaths that hang on the belt, make sure to flag it transparent.

```
#8218
plain leather scabbard~
{20}a plain leather scabbard~
{20}A plain leather scabbard lies on the ground here.~
~
ITEM_TYPE_SHEATH
FLAG_TRANSPARENT
CAN_WEAR_TAKE|CAN_WEAR_BELT
QUALITY_AVERAGE MATERIAL_LEATHER COND_PERFECT SIZE_MEDIUM
20 0 0 0 LAYER_ARMOR 0
```

The following sheath could be closed if the PC wished.

```
#8322
silver wrist sheath~
{70}a silver wrist sheath~
{70}A silver wrist sheath lies on the ground here.~
~
ITEM_TYPE_SHEATH
0
CAN_WEAR_TAKE|CAN_WEAR_WRIST
QUALITY_AVERAGE MATERIAL_SILVER COND_PERFECT SIZE_MEDIUM
4 0 0 0 LAYER_ALL 0
E
silver wrist sheath~
This sheath would hold a dagger perfectly.
~
```

SAMPLE OBJECT - FIRE

Fire is needed for some trades like smithing, and brewing. Fire will also light a room.

```
#8571
hot smithys forge~
{90}a hot smithy's forge~
{90}A hot smithy's forge burns brightly here.~
~
ITEM_TYPE_FIRE
0
0
ITEM_QUALITY_AVERAGE MATERIAL_ENERGY COND_VERY_GOOD SIZE_MEDIUM
1110 1110 0 0 0 0
```

SAMPLE OBJECTS - CARTS

Carts use the same values as containers. They can potentially hold more weight than a container that a PC carries. However, make sure to make value0 more than 200. Carts can be locked with a key, but remember if you sell them with the key, others can buy the key an open each others carts. If you are selling to a limited clientel, this would work. For instance carts only for those of a particular faith. Players are able to get carts with unique keys through the dwelling system.

```
#6736
small wooden ore cart~
{30}a small wooden ore cart~
{30}A small wooden ore cart is here.~
~
ITEM_TYPE_CART
0
0
ITEM_QUALITY_AVERAGE MATERIAL_WOOD COND_PERFECT SIZE_SMALL
80 CONT_CLOSEABLE -1 0 0 0
```

MONEY OBJECT

You can put money in resets on the ground, or in containers. Once a PC picks up the item it turns into coins. The sample below is worth 10 copper.

```
#QQ04
treasure coins~
{B0}a huge treasure~
{B0}There is a huge treasure here, looking very valuable.~
~
ITEM_TYPE_MONEY
0
ITEM_WEAR_TAKE
ITEM_QUALITY_AVERAGE MATERIAL_TYPE_EXOTIC COND_PERFECT SIZE_MEDIUM
10 0 0 0 0 0
E
treasure~
This looks like a whole lot of coins.
~
```

SAMPLE OBJECTS - PIPES AND TABACCO

Value3 on a pipe is its status. In most cases this will be set to 0. For a list of possible status flags check here. Value0 sets how much herb a pipe can hold, value1 is how much is currently in the pipe. Value2 is the herb type that the pipe will use. In general this will be normal tabacco which means you will set this value to 0.

```
#8540
corncob pipe~
{B0}a corncob pipe~
{B0}A pipe made of a corncob lies here.~
~
ITEM_TYPE_PIPE
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_TINY
2 2 0 0 0 0
E
corncob pipe~
{B0}It is a pipe made of a hollowed out cob of corn, stuck on
the end of a similarly hollowed out wooden stick.  What it
lacks in elegance it makes up for in price.
~
```

A pipe will also need herbs to go in it. Note that herbs are now type component.

```
#8146
tabacco~
{30}some tabacco~
{30}Some tabacco lies on the ground here.~
~
ITEM_TYPE_COMPONENT
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_MEDIUM
2 0 0 0 0 0
```

SAMPLE OBJECT - TRAPS

Refer to the traps listing to see what traps are available for use and what they do. See also the trap triggers listing for what will trigger a trap. A trap can have more than one trigger. Trap objects can be sold and retrieved from disarmed traps by characters with the detrap skill.

```
#QQ97
minor spike trap wicked spiked contraption~
{70}a minor spike trap~
{70}A wicked spiked looking contraption lies here.~
~
ITEM_TYPE_TRAP
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_STEEL COND_PERFECT SIZE_SMALL
TTYPE_SPIKE_MINOR 2 TRIGGER_PICK 0 0 0
E
minor spike trap~
It is a contraption that has lethal looking spikes.
~
```

The player can use settrap to place this trap on a door or object to protect it from anyone trying to use lockpick or doorbash.

SAMPLE OBJECTS - LEVERS AND BUTTONS

You need to set the trigger condition in value0 of the lever or button. There can be more than one trigger condition on an object and it needs to be seperated by a pipe | . Some triggers require the presence of TRIG_DOOR to work, and possibly the directional trigger. All triggers start out in the down position by default. This can be changed with the use of TRIG_UP. The lever below transports everyone in the room to room number 96 in the area after someone pushes or pulls it.

```
#QQ07
long steel lever~
{70}a long steel lever~
{70}A long steel lever is on one wall here.~
~
ITEM_TYPE_LEVER
FLAG_MAGIC
0
QUALITY_AVERAGE MATERIAL_STEEL COND_VERY_GOOD SIZE_LARGE
TRIG_TELEPORTALL QQ96 0 0 0 0
E
long steel lever~
It is a long lever that looks like it could be pulled or pushed up or down.
~
```

The button below when pushed or pulled will load up the mob with the vnum of 105 (a goblin) in room number 8.

```
#QQ08
jade green button sphere~
{A0}a jade green sphere~
{A0}A jade green sphere is on one wall here.~
~
ITEM_TYPE_BUTTON
FLAG_MAGIC
0
QUALITY_AVERAGE MATERIAL_STEEL COND_VERY_GOOD SIZE_LARGE
TRIG_MLOAD|TRIG_AUTORETURN QQ08 105 0 0 0
E
jade green button sphere~
It looks like it could be pushed in.
~
```

The chain below (which is type lever) will open up a passage when pushed or pulled. Value1 is the vnum of the room where the passage will open from. Value2 is the vnum of the room where it will connect to. Notice that all of the trigger flags are needed to make this lever work. Also note that TRIG_AUTORETURN is needed to make sure that you do not push the chain.

```
#QQ09
long silver chain~
{70}a long silver chain~
{70}A long silver chain hangs from the ceiling here.~
~
ITEM_TYPE_LEVER
FLAG_MAGIC
0
QUALITY_AVERAGE MATERIAL_STEEL COND_VERY_GOOD SIZE_LARGE
TRIG_PASSAGE|TRIG_DOOR|TRIG_D_UP|TRIG_AUTORETURN QQ50 QQ99 0 0 0
E
long silver chain~
It is made from a shiny silver and hangs from the ceiling.
~
```

SAMPLE OBJECT - QUILLS AND PAPER

Below is the parchment that is used to write notes in the game. You do not need to make any paper objects for that purpose. Paper objects are good for notes that have been written by one mobile to another in quests and so on, but cannot be used for writing notes.

```
#36
piece parchment paper note~
{70}a piece of parchment~
{70}A piece of parchment has been discarded here.~
~
ITEM_TYPE_PAPER
0
ITEM_WEAR_TAKE|ITEM_WEAR_HOLD
ITEM_QUALITY_AVERAGE MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
0 0 0 0 0 0
```

Value0 on TYPE_PEN is the amount of ink in the pen. The standard quill below can write 5 notes before it runs out. Vnum 37 can be sold in any area. If you make quills then make them unique. If you just want plain ones, then use the one from the limbo area file.

```
#37
quill pen~
{F0}a quill~
{F0}A feather dipped in ink here lies.~
~
ITEM_TYPE_PEN
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_IVORY COND_PERFECT SIZE_MEDIUM
5 0 0 0 0 0
```

SAMPLE OBJECT - CORPSES

Builders should never make an object ITEM_TYPE_CORPSE_PC. This is used by hard code to generate the corpse of a player character. In general builders will not make a NPC corpse either. This again is generated by the hard code after a mobile is killed.

Unlike other objects value5 on corpses is used by the hard code. The level of the PC or NPC is placed here when the PC or mobile dies.

SAMPLE OBJECT - BLOOD AND BLOODSTAINS

The blood below has 3 "portions" in it. It will decay after 100 hours.

```
#27
vial blood~
{10}a vial of blood~
{10}A vial of blood spills over on the ground here.~
~
ITEM_TYPE_BLOOD
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_WATER COND_PERFECT SIZE_MEDIUM
0 3 100 0 0 0
E
vial blood~
It is a small glass vial filled with fresh blood.
~
```

The bloodstain below will decay after 1 game hour. This is the bloodstain that is generated automatically by code on the death of monster or PC.

```
#18
bloodstain~
{10}the bloodstain~
{10}Blood stains the ground here.~
~
ITEM_TYPE_BLOODSTAIN
0
0
QUALITY_AVERAGE MATERIAL_FLESH COND_SUPERB SIZE_MEDIUM
0 0 1 0 0 0
```

SAMPLE OBJECT - SCRAPS

Value2 on scraps is the decay timer, that is the time in hours before it junks. To date the only requirement for scraps has been the limbo item that is generated automatically by hard code after armour etc is broken to the point where it is unusable scraps. If you are wanting to put scraps in your area, please contact the builders administration team.

SAMPLE OBJECTS - COMPONENTS AND HERBS

Value0 on components is the number of uses that a PC can get out of it before it disappears when the component is set to be good for more than one use for the spell it is being used for. Value2 is for the herb type if applicable. If there is no herbal affect then the value is set to 0.

Below is a standard component that has no magical properties in its own rights. It can be used in spells however. If the spell allows for the component to be used more than once, then the character can get 10 uses out of the powder before the component is used up (providing its all used on the same spell, some spells use components at a different rate).

```
#26
gem powder~
{D0}some gem powder~
{D0}Some gem powder lies here.~
~
ITEM_TYPE_COMPONENT
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_POWDER COND_PERFECT SIZE_MEDIUM
10 0 0 0 0 0
```

The herb below is of type component but there is a side affect for when the herb is used.

```
#100
adders tongue herb~
{A0}adders tongue~
{A0}An adders tongue herb plant grows here.~
~
ITEM_TYPE_COMPONENT
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PLANT COND_PERFECT SIZE_MEDIUM
2 0 HERB_ADDERS_TONGUE 0 0 0
E
adders tongue~
This herb has one leaf which grows from a stalk about three
inches from the ground.  Its appearance gives its name.
~
```

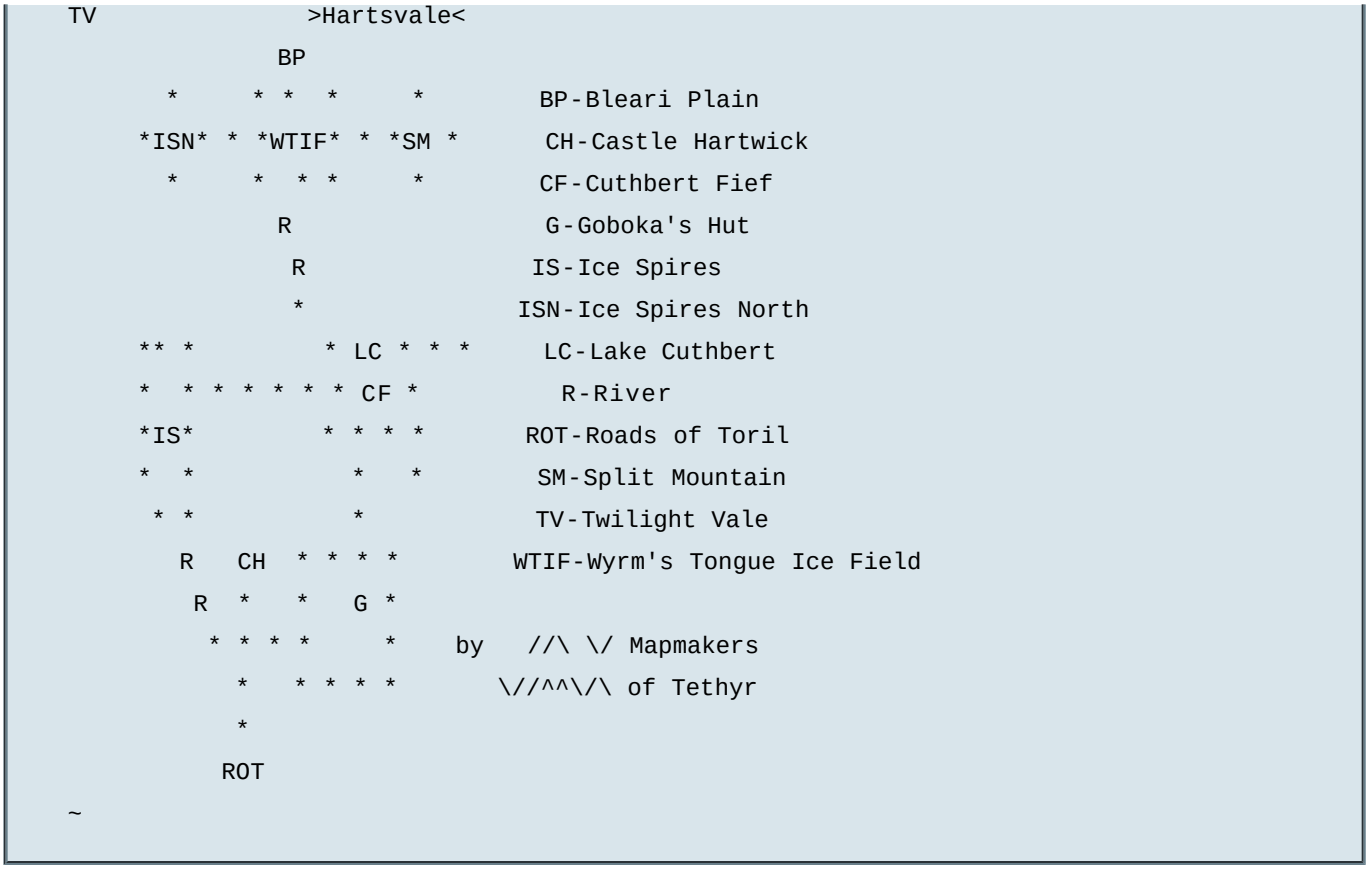
SAMPLE OBJECTS - MAPS

When a character is holding an item that is type map, they will be able to use the map command providing they are within the vnum range specified in value1 and value2. The map below for Waterdeep will work in the main area of Waterdeep. It wont work in attached areas because they are not between 8000 and 8999 which is the vnums of the main Waterdeep area file. So there is some limitations. Also if there is an official Forgotten Kingdoms map created in character by one of our players, please link to the URL in an extra description.

```
#QQ62
map waterdeep~
{70}a map of Waterdeep~
{70}A map lies here in a ball.~
~
ITEM_TYPE_MAP
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
0 8000 8999 0 0 0
E
map waterdeep~
www.forgottenkingdoms.com/maps/waterdeep.html
Please refer to this map in character.
~
```

You could also do an ascii art representation of the map. This map below will still work in Hartsvale with the map command because the vnums have been set in value1 and value2.

```
#8063
map hartsvale~
{70}a map of Hartsvale~
{70}A map lies here in a ball.~
~
ITEM_TYPE_MAP
0
ITEM_WEAR_TAKE|ITEM_WEAR_HOLD
ITEM_QUALITY_AVERAGE MATERIAL_PAPER COND_PERFECT SIZE_MEDIUM
0 5700 5799 0 0 0
E
map hartsvale~
```


TV >Hartsvale<
BP
BP-Bleari Plain
CH-Castle Hartwick
CF-Cuthbert Fief
G-Goboka's Hut
IS-Ice Spires
ISN-Ice Spires North
LC-Lake Cuthbert
R-River
ROT-Roads of Toril
SM-Split Mountain
TV-Twilight Vale
WTIF-Wyrm's Tongue Ice Field
by //\ \/ Mapmakers
\//^^\/\ of Tethyr
ROT

SAMPLE OBJECT - SHOVEL

Shovels are needed to bury items and dig them up. Below is a sample of a shovel.

```
#QQ30
sturdy shovel~
{70}a sturdy shovel~
{70}A sturdy shovel lies here. ~
~
ITEM_TYPE_SHOVEL
0
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_AVERAGE MATERIAL_METAL COND_PERFECT SIZE_MEDIUM
0 0 0 0 0 0
```

SAMPLE OBJECT - TRADEGOODS

Tradegoods have the same settings as trash, but they are heavier to carry around. If you wish to use tradegoods in your area please contact the area administrators.

SAMPLE OBJECT - HIDE

Type hide can be used in leather armour making. Most of the time players will use the tanned skins that are gained with the slice skill. Sometimes however in a quest you will want to generate a tanned hide. Please note that this is not a raw skin, but a hide that has been through the tanning process. It is not the intent of this type to allow builders to override that products that come from the slice skill. This is for rare and odd instances where a hide may be needed. Hide uses the same values as armour, but in general you should set it to ARMOR_TYPE_HIDE.

```
#QQ66
tanned leather hide deer~
{30}hide of a deer~
{30}A tanned leather hide of a deer lies spread out on the ground here.~
~
ITEM_TYPE_HIDE
0
CAN_WEAR_TAKE|CAN_WEAR_BODY
QUALITY_AVERAGE MATERIAL_LEATHER COND_PERFECT SIZE_MEDIUM
0 0 LAYER_OVER ARMOR_TYPE_HIDE 0 0
```

SAMPLE OBJECTS - OBJECTS WITH MAGICAL APPLIES

There are over 100 possible applies that can be applied to an object. Refer to the applies list for more information on what is availble and a short summary of what each one does. Below are many samples of the way to use just some of the applies. It should be noted that for skill applies that unless the character knows some of the skill in the first place, the item will not work. Due to abuse of the system we had to stop characters getting applies from skills they did not know.

```
#QQ12
perky green velvet hat~
{A0}a perky green velvet hat~
{A0}A perky green velvet hat lies here.~
~
ITEM_TYPE_ARMOR
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HEAD
QUALITY_SUPERIOR MATERIAL_CLOTH COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_CLOTH 0 0
A APPLY_STEAL 1
A APPLY_PICK 1
E
perky green velvet hat~
It is made of a green velvet
~
```

The hat above adds 1 to the steal and pick skills of the wearer.

```
#QQ13
large wedged boots~
{30}larged wedged boots~
{30}A pair of larged wedged boots lies here.~
~
ITEM_TYPE_ARMOR
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_FEET
QUALITY_SUPERIOR MATERIAL_LEATHER COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_LEATHER 0 0
A APPLY_HEIGHT 10
E
larged wedged boots~
The soles of these boots are at least 10 inches high.
```

```
~
```

The boots above add 10 inches to the characters height.

```
#QQ44
tight white girdle~
{F0}a tight white girdle~
{F0}A tight white girdle lies here.~
~
ITEM_TYPE_ARMOR
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_BODY
QUALITY_SUPERIOR MATERIAL_CLOTH COND_PERFECT SIZE_MEDIUM
0 0 LAYER_UNDER ARMOR_TYPE_CLOTH 0 0
A APPLY_WEIGHT -20
E
tight white girdle~
When worn this girdle will make it hard to breath but you will
swear that you are at least 10 pounds lighter.
~
```

The girdle above takes 20 pounds off the characters weight (not the weight the character can hold). This kind of thing could be handy for characters who want to get rides on other larger characters.

```
#QQ35
light hand crossbow distance~
{70}a light hand crossbow~
{70}A light hand crossbow lies on the ground here.~
~
ITEM_TYPE_WEAPON
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_SUPERIOR MATERIAL_OAK COND_PERFECT SIZE_SMALL
0 0 0 WEAPON_TYPE_LIGHT_CROSSBOW 0 0
A APPLY_RANGE 1
E
light hand crossbow~
It is of superior construction and tightly wound.
It appears to be made of oak.
~
```

The crossbow above will shoot 1 room further than a normal one.

```
#QQ76
helmet hardened mud~
{30}a helmet of hardened mud~
{30}A helmet of hardened mud lies here.~
~
ITEM_TYPE_ARMOR
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HEAD
QUALITY_SUPERIOR MATERIAL_STONE COND_PERFECT SIZE_MEDIUM
0 0 LAYER_ARMOR ARMOR_TYPE_SPLINT_MAIL 0 0
A APPLY_AFFECT AFF_DETECT_BURIED
E
helmet hardened mud~
{30}Pressed into the mud are crude runes.
~
```

The helmet above is affected with the affect of detect buried. This means the wearer can see things that are buried.

```
#QQ17
large halberd gluttony~
{70}a large halberd~
{70}A large halberd lies on the ground here.~
~
ITEM_TYPE_WEAPON
FLAG_MAGIC
CAN_WEAR_TAKE|CAN_WEAR_HOLD
QUALITY_SUPERIOR MATERIAL_STEEL COND_PERFECT SIZE_LARGE
0 0 0 WEAPON_TYPE_HALBERD 0 0
A APPLY_WEAPONSPELL_ONE SPELL_CREATE_FOOD
E
halberd large~
There is an unusual rune on the handle of the halberd.
~
```

The halberd above will have a 10% chance of creating food when it hits in battle.

ROOMS

AREA LAYOUT AND ROOM DESCRIPTIONS

Area layout and map

I have found that different builders work in different ways when working out the layout of an area. Some will work on paper, some will work out the layout on the computer, others will just go with it on the fly. This is how I (Mystra) work. If you are unsure where to start, then give my way a try, till you find your own way of working on the layout of an area.

First map out your area on paper. I have a pad of graph paper that I use. I use a mechanical pencil because it gives a nice fine line. Each box is 1/4 inch square. I prefer to represent each room as a box, giving it a title as needed. So my drawn square will cover 4 of the smaller squares on the graph paper. Sometimes I will use 1 square of the graph paper to be the size of the room square, and I put a number in the square and then I put a legend on the page, so I can refer to it, to see what the number represents. I only tend to do this when I am working from a real Forgotten Realms area that has a published map and legend.

Generally when you are working on your first area, you have been allocated 50 or 100 vnums. These are easier to layout than some of the bigger areas. Some of my bigger areas ended up being on 4 bits of paper taped together.

Once you have it all mapped out number each room from 0. I tend to do this in red pen, so it stands out differently from my penciled numbers and notes. Before I mark the map in red pen that cannot be easily erased, I will do a quick count to make sure I have not gone over my vnum limit for the area. If I find I have 208 rooms, I will prune my area down to 200. I allocate builders vnums in blocks of 50 for small areas, and then blocks of 100 for larger areas. The number of total vnums available is not infinite so we must use our vnums wisely and well. Remember as you are numbering the rooms from 0, your final room number will be 49 or 99 or 199 and so on (depending on how many vnum blocks you plan on using). I often try to not use up all of my room vnums that I have allocated the area, I like to leave a handful like 5 free. I sometimes find that when I am writing a quest for the area, that I need to add a room to make the quest work, and having a couple of rooms spare allows me to do that.

Room Descriptions

I work on my room descriptions (and the area file as a whole) in a plain text editor program. My personal recommendation is textpad. But dos edit, note pad and word pad will work, though without the niceties that textpad offers. (You can use Word but I found it irritating to use in the way it likes to save things as a doc and uses auto correct) Use the following format for your room and room description. (Please note that the following example creates a no exit room - exits are covered in following lessons).

```
#QQ01
Room Name~
{70}This is the description of the room. Make it a minimum of 4 lines if you
can. We dont mind if you cut and paste to copy similar rooms, we are not as
fanatical about such things as making you make all rooms unique. Try and
```

```
make the room description no more than about 10 lines. You dont want the
description being too big for someones telnet client. It is important that
you put a tilda after the room name and the room description like this.
~
0 0 0 0 0 0
S
```

Now lets look at each aspect of the room's code:

`#QQ01` - The hash is used to denote to the game code that you are starting a new rooms information. QQ is the vnum of the area. For instance Waterdeep had all the QQ's replaced with 80 as its vnums were 8000. If you have more than 100 rooms then for room 100 use QR00. And so on up if you go to more than 200 rooms it would be QS.

`Room Name~` - The room name is what you see above the exits in vt. Don't forget the tilda ~.

`{70}This is the description .... ~` - This is where you put your room description paragraph. If you read the paragraph of our sample room, it has some guidelines for how we would like the room description to be. All rooms must be colourised. See the lesson on room colourisation for more information. Don't forget the tilda ~.

`0 0 0 0 0 0` - This is where our room and sector flags go. This is covered seperately in the rest of the room lessons.

`S` - Finish each room off with an S. At this point your room is a no exit room with no room flags.

You don't have to worry about exits for now. However if you want to do them all in one hit (I do) continue with the rest of the room lessons. Later we (the area admins) will do a global replacement on the QQ's (and QR's etc if needed) to your vnums once you have submitted your area for testing. You will be allocated vnums once your area is ready for testing. Do not try to allocate numbers to yourself. You have no means of knowing what numbers are available.

The rooms tend to be what many builders find the most tedious and I personally have always made a policy to get them out of the way first before getting into the real fun stuff :) OLC when you earn the rights will make rooms far easier to make.

ROOM COLOURISATION STANDARDS

Refer to the lesson on colourising an area for the syntax on putting colour into an area.

When colourising a room we try to give an overall feeling for where the room is by the colour. For instance if the room is a dark tunnel we will use {80}dark grey, or if its the inside of a tavern where everything in predominantly wood, we will use {30}. A forest would be {A0} or {20} to give the impression on green. A forest that is darker would be {20} over {A0}. Sometimes you might want to highlight a feature of the room, like a stained glass window, or a painting, or the fireplace. When doing this, do not try to colourise every individual aspect of the room description, because when that is done, the room becomes very busy and hard to read.

Below are some sample rooms:

```
#QQ11
A mine shaft~
{80}The walls are pockmarked with pick and shovel marks, showing where
veins of metal had been located. Torches spaced intermittently
through the cavern provide flickering illumination. Sturdy timbers
form crossbeams for the tunnel, while the mountain overhead is
supported by carved pillars of stone.
~
0 ROOM_NO_MOB|ROOM_INDOORS|ROOM_NO_ASTRAL SECT_UNDERGROUND 0 0 0
DDIR_UP
~
~
0 -1 QQ10 1
DDIR_DOWN
~
~
0 -1 QQ12 1
S
```

The room above is an underground room that would be darkish in colour so the dark gray colour is used to give that over all impression.

```
#QQ65
Ardeep Forest~
{A0}The trees in this part of the wood are tall and mossy with the passing of
years.  The ground seems to be somewhat softer here, with many low-growing
```

```
succulent leafy plants inhabiting the area.  Perhaps this has something to do
with the sounds of water reaching you from the north and east.
~
0 ROOM_DARK SECT_FOREST 0 0 0
DDIR_NORTH
A Creek in Ardeep Forest~
~
0 -1 QQ73 0
DDIR_EAST
A Creek in Ardeep Forest~
~
0 -1 QQ66 0
DDIR_SOUTH
Ardeep Forest~
~
0 -1 QQ62 0
DDIR_WEST
Ardeep Forest~
~
0 -1 QQ64 0
E
tree trees~
{A0}Here there is a small grove of duskwood.  {50}Low bushes
with purple leaves grow about the bases of the trees.
~
E
succulent plant plants leafy~
{A0}They are brilliant green and have bunches
of tiny white flowers sprouting from their stalks.
~
S
```

The room above is a forest. Its descriptions are done with the brighter of the two greens available. The extra descriptions have been colourised as well. The builder has also chosen to put comments in the first comment line of the room exits. This is not necessary as it is not seen in the game at all, but as a builder you may like that kind of detail when looking through your area file.

```
#QQ46
Golden Horse Shoe~
{30}This long building contains the numerous stables of The Lucky Drunk.
This stable gets its name from the {B0}golden horse shoe {30}nailed above
the entrance. The stables are kept very clean and the sweet smell of
fresh hay is very strong. There are also many cages for smaller pets
```

```
and on one wall is a pegboard. Bales of hay and barrels of grains
are stacked in one corner.
~
0 ROOM_INDOORS|ROOM_PET_SHOP|ROOM_NO_RECALL|ROOM_NO_ASTRAL  SECT_INSIDE 0 0 0
DDIR_EAST
~
~
0 -1 QQ00 1
S
```

In the description above the builder has tastefully highlighted just one feature that he wants to draw attention to, the golden horseshoe. Drawing attention to too many things with many different colours becomes too busy for the player to read. So try and keep it to just one item if you want to highlight something.

ROOM FLAGS AND ROOM SECTOR

The room flags and room sector type are set in the line of six values after the room description, before the exit information. Room Flags describe specific attributes of the room, and your room can have more than one flag. The room can only have one sector type.

In the example below pay particular attention to the line of 6 values after the actual room description and the room description's tilda. Note that our sample room is still a no exit room. Exits will be covered in other room lessons.

```
#QQ00
A gate to a walled forest~
{70}A stone gateway in the middle of a stone wall is here.  Over the
top of the wall can be seen the green of tall trees.  To the east
is the hustle and bustle of the ever busy streets of Waterdeep,
but from over the wall can be heard the sounds of birds.
~
0 ROOM_NO_MOB|ROOM_DARK SECT_CITY 0 0 0
S
```

Lets look in detail at the line of six values after the room description:

- Value0 - The first value of 0 refers to a now defunct field no longer used by the game so just put a 0 in. You cant ommit it altogther as the game will crash. You need to have a 0 in there as a place marker. The coders may in time decide to use that field again for something new that they might code.
- Value1 -The next value refers to the actual room flags. A room can have more than one room flag and it can even have none. If it has none just put in a 0. This particular room is dark so for those races without infrared vision, will need to have a light to see what is in the room. It is also flagged no mob. This will prevent mobiles from wandering in and out of the room. See the below table for the possible room flags for your room. A room can have more than one room flag.M/li>
- Value2 - The next value refers to the rooms sector type. This refers to the type of terrain the room is and affects how a character moves through that room. Some sectors require special capabilities for a character to move through it (ie to swim or fly). See the Sectors table below for a list of the allowable sectors. A room can have only one sector type.
- Value3 - When a PC enters this room, after the delay time set with this value the PC is teleported out of the room to the vnum set in Value4. The room must have the ROOM_TELEPORT flag set on it for this aspect of the room code to work.
- Value4 - This is the vnum that the PC is teleported to after the delay set with Value3.
- Value5 - This is known as the tunnel value. This can determine how many characters can fit into the room. The solitary and private flags will set the numbers to 1 and 2 respectively, but if you want a different amount of PC's allowed in a room, then you may set this value.

ROOM FLAGS LIST

ROOM_DARK	light is needed in room
ROOM_DEATH	Do not use
ROOM_NO_MOB	mobs cannot enter room
ROOM_INDOORS	not affected by weather
ROOM_LAWFUL	good aligned chars only
ROOM_NEUTRAL	neutral aligned characters only
ROOM_CHAOTIC	evil aligned chars only
ROOM_NO_MAGIC	players cannot cast spells
ROOM_NO_TUNNEL	Do not use
ROOM_PRIVATE	only 2 players may enter room
ROOM_SAFE	no fighting in this room
ROOM_SOLITARY	only 1 player may enter room
ROOM_PET_SHOP	room is a petshop
ROOM_NO_RECALL	players cannot recall
ROOM_DONATION	Prevents players from using 'get all'
ROOM_NODROPALL	stops players doing "drop all", ideal for public squares etc
ROOM_SILENCE	players may not speak or emote
ROOM_LOGSPEECH	Do not use without builder admin consultation
ROOM_NODROP	players may not drop stuff

ROOM_CLANSTOREROOM	Used for guild storerooms. Ask a builders admin first.
ROOM_NO_SUMMON	player cannot be summoned from room
ROOM_NO_ASTRAL	cannot gate or magically transport to or from this room
ROOM_TELEPORT	will teleport the PC after the delay set in value3 to the vnum set in value4
ROOM_TELESHOWDESC	when teleported it shows the PC's the description of the new room
ROOM_NOFLOOR	players and objects fall to (down) room
ROOM_PROTOTYPE	used by OLC. Do not use.
ROOM_INN	allows PC's to heal at a faster rate

SECTORS LIST

SECT_INSIDE	inside a building or structure etc. It is always lit
SECT_CITY	typical city street, it is always lit
SECT_FIELD	a grassy field
SECT_FOREST	heavily wooded forest
SECT_HILLS	rolling hills
SECT_MOUNTAIN	mountainous terrain
SECT_WATER_SWIM	calm water
SECT_WATER_NOSWIM	swimming skill required
SECT_UNDERWATER	Water-breathing required. Character swims.
SECT_AIR	flying required
SECT_DESERT	dry sandy terrain
SECT_DUNNO	Do not use. Reserved for future use.

SECT_OCEANFLOOR	Underwater. Breathwater is required. Character can WALK.
SECT_UNDERGROUND	underground structure
SECT_WOODS	lightly wooded terrain
SECT_ROAD	roads outside of cites
SECT_TUNDRA	cold scrub land/frozen wastes
SECT_BARREN	barren lands/moors/rocky, treeless plains

ROOM EXITS

There are 6 possible normal room exits. North South West East Up and Down. Please note that non wilderness area's may NOT use the exits northeast, northwest, southeast and southwest. If your area arrives with those exits in it, your area will be sent back to be reworked. The first room example has only one room exit. Take note of the four lines that are in an exit. Often new builders submit their area for the first time with these lines out of order.

```
#QQ00
The Riven Shield Shop~
{30}This shop is famous up and down the Sword Coast for its large and varied
assortment of secondhand arms and armour.  No one is tempted to steal
any of the more valuable pieces for it is widely known that some of the
shields on the ceiling contain magically imprissoned monsters that
can be released to fight as an ally of the shield wearer.
~
0 ROOM_INDOORS SECT_INSIDE 0 0 0
DDIR_WEST
~
~
0 -1 QQ01 0
S
```

The room above has one exit out of the room. It leads to the west and goes to room number 01 in the area. Lets look at the elements of the exit information:

`DDIR_WEST` - This is telling the game which way you want the direction to go. This is the format that you should use.

`~` - The tilda's on each of the next two lines are needed by the code. The first line can be used for a short comment before the tilda if you so wish. For instance when I am connecting up another area to this room, I will put a comment stating the name of the area that is being connected to it. The second tilda line is used for keywords for doors and gates. More information is about that below.

`0 -1 QQ01 0` - The `0` is where the door flags would go if there were any. The `-1` is the vnum of the door key. Because there is no door and therefore no lock in this room, the `-1` denotes that there is no key. The `QQ01` is the vnum of the room that the exit is going to. The `0` is the size of the exit. 0 or 1 denotes that there are no size limits on the room. See the table below for size limits.

Note that the `S` is after all the room's information. This tells the game when all the information about the room is in, even the programs and extra descriptions. (See the lessons about room programs and extra descriptions).

To follow is an example of a room using all allowed exits.

```
#QQ00
Refrectory~
{30}A large and airy room that holds several long tables, crafted out
of gleaming wood. This room serves as an informal meeting places
for students and teachers alike. Quarters are scattered around
for overnight stays and delicious smells drift from the kitchen.
~
0 ROOM_INDOORS|ROOM_NO_ASTRAL  SECT_INSIDE 0 0 0
DDIR_NORTH
~
~
0 -1 QQ19 1
DDIR_EAST
~
~
0 -1 QQ20 1
DDIR_SOUTH
~
~
0 -1 QQ22 1
DDIR_WEST
~
~
0 -1 QQ21 1
DDIR_UP
~
~
0 -1 QQ23 1
DDIR_DOWN
~
~
0 -1 QQ02 1
S
```

Rooms with doors and gates

Below is a sample of a room that has a locked door.

```
#QQ14
```

```
Waterdeep Dungeon Entrance~
{80}It is damp and cold in here.  The door is made of a heavy iron. It was designed
to keep people in rather than out. There is a dripping sound in the distance.  Not
a pleasant place to have to spend the night or any length of time.  Mouldering
straw is piled up in the corners.
~
0 ROOM_INDOORS SECT_INSIDE 0 0 0
DDIR_UP
~
~
0 -1 QQ04 0
DDIR_DOWN
~
iron door~
EX_ISDOOR|EX_CLOSED|EX_LOCKED QQ76 QQ15 0
S
```

The first exit of this room is heading up is like the others in the previous examples. It has no doors. There is nothing stopping the way up. The exit down however is barred by a locked door that requires a key to unlock it. There is nothing to prevent it from being picked, bashed or passed through by magic. There are flags that you can add to an exit to prevent all or some of these events. The last two lines of the exit code are what is important when making a room. The DDIR line and the first tilda line are the same as any other exit.

`iron door~` - If you have a door or a gate in your room this is the field where you put in the keyword for the exit. For instance if it is a gate you will put gate in as keyword. Or you could put in trapdoor. The player would have to type open doorname. If there is no door just put in a tilda ~ like in the other rooms. If you want to have an ivory door and you want it to show that the player opens the ivory door rather than opens the ivory you would put the ivory door between quotes like this `'ivory door'~`.

`EX_ISDOOR|EX_CLOSED|EX_LOCKED QQ76 QQ15 0` This is a room exit with a closed locked door. In order for a door to be you must have the ISDOOR flag. When the game boots up after copyover or crash, this door will be locked and closed. It will remain thus until someone opens it. Once open it will stay that way until someone closes it. If you want a door close automatically, you can set a door reset (see the lessons about door resets), or you can put programs on a mobile who is guarding the door or gate. Object QQ76 will open this door. QQ76 may not be of ITEM_TYPE_KEY but could be any object. As you can see you can have more than one room flag separated by a | pipe. QQ15 is the room that the exit is going to. The 0 is the field for the size of the PC that can fit through this exit. In this case any sized PC may fit through this door.

Somewhere Exits

A somewhere exit is where the player types a keyword to enter the area rather than heading in a direction. Somewhere exits can be on top of all the other exits, and you can have more than one somewhere exit in a room providing they have

different keywords. Below is a sample of a somewhere exit.

```
#QQ01
Before the gates of Daggerford.~
{70}The large gates of the city of Daggerford stand before you.  Although not as big
as its nearby cousin, Waterdeep, the city of Daggerford has much to offer an
adventurer.  There is a constant traffic of caravaners and traders into and out
of the city.
~
0 0 SECT_ROAD 0 0 0
DDIR_SOMEWHERE
Daggerford~
path~
EX_XAUTO -1 QQ02 1
S
```

The somewhere exit above can be entered by typing the word path. The EX_AUTO flag is needed to make the somewhere exit work correctly. When the PC types path, they will end up in room 02 of the area.

Door Flags Listing

Below is a list of possible door flags. Try and stick to the simple flags, unless you have good reason don't use some of the more "exotic" sounding exits.

EX_ISDOOR	exit has a door
EX_CLOSED	exit with door is closed
EX_LOCKED	exit with door is locked
EX_SECRET	cannot be seen or opened
EX_PICKPROOF	door cannot be picked
EX_FLY	player must fly to pass the door
EX_CLIMB	players must climb to pass
EX_DIG	players must dig the exit

EX_NOPASSDOOR	passdoor doesn't work
EX_HIDDEN	cannot be seen but can open
EX_PASSAGE	passage opened by a mob program
EX_PORTAL	spells that create portals
EX_XCLIMB	typing 'climb' will scoot you out this exit
EX_XENTER	typing 'enter' will move a PC out of this exit
EX_XLEAVE	typing 'leave' will send a PC out this exit
EX_XAUTO	players will 'automatically' use this exit. This is required for somewhere exits with keyword entrances.
EX_XSEARCHABLE	door can be found in standard search
EX_BASHED	exit has been bashed
EX_BASHPROOF	exit cannot be bashed
EX_NOMOB	mobiles cannot pass
EX_WINDOW	players can look even thru closed door
EX_XLOOK	players can look through the exit

Exit sizes

You can limit the size of your exit and make it impossible for creatures larger than a certain size to fit through. For instance you can set your exit that only creatures that are sized small may fit through the exit. This is ideal for simulating tiny tunnels that only a small gnome might fit through but not a larger creature like an orc. The table below shows the numbers to be used for the sizes.

EXIT SIZE	BIT VECTOR
EXIT_SIZE_ANY	0
EXIT_SIZE_TINY	1000

EXIT_SIZE_SMALL	1001
EXIT_SIZE_MEDIUM	1002
EXIT_SIZE_NORMAL	1002
EXIT_SIZE_LARGE	1003
EXIT_SIZE_HUGE	1004
EXIT_SIZE_GIANT	1005

EXTRA ROOM DESCRIPTIONS

Extra descriptions can be put in for keywords in your normal room description. For instance you might have written something about markings on the wall in your normal description which would indicate that if a player typed look markings or look wall they would get additional information. Extra descriptions give your area more depth, and make it far more interesting for players. Extra descriptions are ideal to use in quests where you want the character to really take notice of their surroundings when working out what to do next in the quest. A room can have more than one extra description, but each extra description needs to have different keywords. A sample room follows.

```
#QQ15
Waterdeep Dungeon~
{80}In the corner is a pile of straw for you to sleep on.  It is none too
clean and it looks to be rather damp.  There is no windows and you are
somewhat below sea level as the walls are very damp.  You see some
intriguing scratchings on the wall.
~
0 ROOM_DARK|ROOM_INDOORS|ROOM_NO_ASTRAL|ROOM_NO_MAGIC|ROOM_NO_SUMMON|ROOM_NO_RECALL
SECT_INSIDE 0 0 0
DDIR_UP
~
iron door~
EX_ISDOOR|EX_CLOSED|EX_LOCKED QQ76 QQ14 0
E
scratch scratchings~
Darrak was here first!
~
S
```

In the room above when the player looks at the scratchings on the wall they will see etched into the wall the words "Darrak was here fist!".

As you can see you put the extra description information after the room exits. The E indicates to the game that to follow is an extra room description. The next line lists the keywords that a player can type to look at. Note that shortening the keyword will not work, (ie to mark) they must type in the full word. So if you want to be kind to players you would put in the words scratch and scratchings into the keywords. Make sure you end the keywords with a tilda ~.

After the keywords put in the actual text that the player will see on typing look markings. Make sure you end the description with a tilda ~.

ROOM PROGRAMS

Like mobiles and objects, rooms can have programs in them. More information on writing programs is available in the section of the lessons devoted to programs. For a list of program triggers that will work in rooms, see the list of mobprogs.

The following is a sample of the usage of a program in a room. Note that the S in the room information comes AFTER the program's pipe. There are two programs in the room below. The rirst one is one that triggers 30% of the time. It places food in and out of the dungeon. The second one prevents players from using supplicate in this room.

```
#8115
Waterdeep Dungeon~
In the corner is a pile of straw for you to sleep on.  It is none too
clean and it looks to be rather damp.  There is no windows and you are
somewhat below sea level as the walls are very damp.  You see some
intriguing scratchings on the wall.
~
0 ROOM_DARK|ROOM_INDOORS|ROOM_NO_ASTRAL|ROOM_NO_MAGIC|ROOM_NO_SUMMON|ROOM_NO_RECALL
SECT_INSIDE 0 0 0
DDIR_UP
~
iron door~
EX_ISDOOR|EX_CLOSED|EX_LOCKED 8076 8114 0
E
scratch scratchings~
Darrak was here first!
~
>rand_prog 30~
if quest(16,5,$r) < 14
  if quest(16,5,$r) > 0
    mpmadd $r quest 16 5 1
    if rand(20)
      mpecho You hear the rustle of a rat in dark corner.
    else
      if rand(20)
        mpecho Someone shoves a plate of gruel though a flap in the door.
        mpoload 8286
        drop i8286
      else
        if rand(20)
          mpecho Someone shoves a battered tin cup of water through a flap in the door.
          mpoload 8287
          drop i8287
```

```
        else
          if rand(20)
            if ovnumroom(8286) > 0
              mppurge i8286
              mpecho Someone reaches in through a flap and takes the old gruel away.
            endif
          else
            if ovnumroom(8287) > 0
              mppurge i8287
              mpecho Someone reaches in through a flap and takes the old tin cup away.
            endif
          endif
        endif
      endif
    endif
  endif
else
  mpechoat $r A guard comes and tells you that you have served your time.
  mpechoat $r He escorts you to the gates of the city.
  mpechoaround $r A guard comes and tells $r $e has served $s time and takes $m away.
  mpmset $r quest 16 5 0
  mptransfer $r 8001 pet
  mpat 8001 mpforce $r look
  mpat 8001 mpechoaround $r $r is shoved out the gates of the city by a guard.
endif
~
>intercept_prog supplicate~
mpechoat $n OOC: We have disabled this function for the dungeon.
mpechoat $n If you wish to supplicate and use favour to get out of the dungeon,
mpechoat $n then it is our feeling that you should roleplay it with your
mpechoat $n deity rather than using the automatic function.
~
|
S
```

SHOPS & REPAIR

GENERAL SHOPS

Shops need to be set up in the #SHOPS section of the area file and what they sell set up in the #RESETS section of the file. The #SHOPS section only determines what a shop will BUY and that the mobile is a shop. It will set the opening and closing hours, and the profit from buying and profit from selling percentages. Shops are determined by the mobile and not the room. If the mobile moves, so then does the shop. The shops section starts with #SHOPS and finishes with a 0.

A sample shop - The Gentlemens Emporium in Waterdeep, where Danillo Thann is the shop-keeper. Note that when Danilo moves away then he technically takes the shops inventory with him and nothing can be bought from the room alone.

```
#SHOPS
8000 ITEM_TYPE_ARMOR ITEM_TYPE_TREASURE ITEM_TYPE_CONTAINER ITEM_TYPE_NONE ITEM_TYPE_NONE
150  50  7 19    ; Danilo in Gentlemens Emporium
0
```

Lets break the code above down into each part:

8000 - This is the vnum of the shopkeeper. You would need to make the shop-keeper in the #MOBILES section of your area. When creating the area this would be QQ00.

ITEM_TYPE_ARMOR ITEM_TYPE_TREASURE ITEM_TYPE_CONTAINER ITEM_TYPE_NONE ITEM_TYPE_NONE - This is what the shop will BUY not sell. What is sells is determined by what you give the shopkeeper in its inventory in the #RESETS section. A shop can buy up to 5 item types. For a list of all possible types look at the item types list. The shop in the sample above will buy armour, treasure and containers from characters. Once he has bought that stuff from them, it is available for resale. So make sure you are going to be happy with what the shop-keeper is reselling. For instance, if your shop-keeper is a specialist jeweller, then hes not going to want to buy furniture or armour, he will likely only buy ITEM_TYPE_TREASURE.

150 50 - This is the profit-buy and profit-sell. This is a markup for characters buying the item from the shop-keeper, in percentage points. The markup is taken from the actual value of the item. 100 is nominal price; 150 is 50% markup, and so on. The 'profit-sell' number is a markdown for players selling the item, in percentage points. 100 is nominal price; 75 is a 25% markdown, and so on. The buying markup should be at least 120, and the selling markdown should be at most 80. Common shops in Forgotten Kingdoms are set to 150 buy and 50 sell.

7 19 - This is the open-hour and close-hour. These numbers define the hours when the shopkeeper will do business. For a 24-hour shop, these numbers would be 0 and 23. This particular shop opens at 7 am and stays open till 7 pm. On Forgotten Kingdoms we want realistic trading hours. Most of the shops around Waterdeep are only open during the day.

Innkeepers who are also shops have much longer trading hours only shutting in the wee hours of the morning. If you are making a large city or township, it is a good idea to have a tavern or two open late to trade food and drink to characters.

`; Danilo in Gentlemens Emporium` - This is the comment which starts with a `;` All your shops should have comments, as it makes the area far easier to read and edit when they do.

Note that there is no room number for a shop. Just load the shopkeeper mobile into the room of your choice in resets, and make it a sentinel. Or, for a roving shopkeeper, just make it non-sentinel. Lets at what else we need to make our shop work. First we need to make the actual mobile and put it in the #MOBILES section of our area file.

```
#8000
danilothannmob thann~
{70}Danilo~
{70}Danilo Thann gives you a wicked grin.
~
He is a handsome man, dressed to the hilt in the finest of clothes.
He is from one of the most wealthy families of Waterdeep.
Actually he looks like a bit of a dandy.
~
U 45 CLASS_BARDS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_LEATHER MATERIAL_LEATHER
d6+1 1000
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON|LANG_ELVEN
LANG_COMMON|LANG_ELVEN
RIS_NONE RIS_NONE RIS_NONE
```

Our shop-keeper is flagged sentinel so that he will remain in the room that is his shop. If he was a wandering merchant we would not put the sentinel flag on him and allow him to wander the area. If you wanted the shop to only wander a particular sector type then you would use `ACT_NOWANDER`, and if you wanted him to stay within your area you would use `ACT_STAY_AREA`.

The objects a shopkeeper sells are those loaded by 'G' reset commands for that shopkeeper. These items replenish automatically. If a player sells an object to a shopkeeper, the shopkeeper will keep it for resale if they do not already have an identical object. Items sold to shops do not replenish. To limit the stock that a shop-keeper has of an item you would set a maximum number of the item in the game in the RESETS. It is not defined in the #SHOPS section of the code. Let us look at our examples resets. More information on resets are available in the resets section of the lessons page.

```
M 0  8000   1  8089;                 Danilo in Gentlemen's Emporium
G 1  8022  30 ;  blue velveteen breeches
G 1  8023  30 ;       a purple silk robe
G 1  8025  30 ;    a maroon leather belt
G 1  8026  30 ;      green silk slippers
G 1  8229 30 ; green velvet cape
G 1  8079 100 ; a fine cotton undershirt
G 1  8087 100 ;a pair of fine wool socks
G 1  8230 20 ; jade green coat
G 1  8231 20 ; jade green trousers
G 1  8234 30 ; ivory ring
```

Danilo is loaded into room 8089 and there is only 1 of him. He sells alot of clothing. It is easy to see what he sells because the resets have been well commented. If there are 30 blue velveteen breeches in the game already, then while the characters see this item on list, Danilo will be out of stock when they go to buy the item. This is a good way to limit some rarer items you might have for sale that you do not want to see on every character in the game. Note that if Danilo was equipped with clothing and weapons with the E reset, he would not sell these items. He only sells what is given to him with a G reset.

OBJECT TYPES

The 'item-type' is the type of the item (weapon, armor, potion, etc). Depending on the item type, value0 through value5 will have different meanings. Any value that is not used is set to 0. EX. for a light value2 is the number of hours left until the light burns out. Value5 on all objects is unused by hard code and can be used by builders in object programs to set and check the status on an object. As a result Value5 is not shown on this table.

A spell number of zero or negative value means 'no spell'.

Bit Vector/Item Type	Value0	Value1	Value2	Value3	Value4
1 ITEM_TYPE_LIGHT	unused	unused	hours left, 0 is dead, -1 is infinite. Infinite lights are to be rare magical items.	unused	unused
2 ITEM_TYPE_SCROLL	level of spell/s *	spell number 1	spell number 2	spell number3	unused
3 ITEM_TYPE_WAND	level of spell	max charges	charges left	spell number	unused
4 ITEM_TYPE_STAFF	level of spell	max charges	charges left	spell number	unused
5 ITEM_TYPE_WEAPON	unused	weapon flag	weapon flag modifiers	Weapon Type	unused
7 ITEM_TYPE_SHEATH	capacity in pounds	container flags	key vnum	unused	layers
8 ITEM_TYPE_TREASURE	unused	unused	unused	unused	layers
9 ITEM_TYPE_ARMOR	unused	unused	layers	Armor type	unused
10 ITEM_TYPE_POTION	level of spells	spell number 1	spell number 2	spell number 3	unused
12 ITEM_TYPE_FURNITURE	unused	unused	Furniture State	unused	unused

13 ITEM_TYPE_TRASH	unused	unused	unused	unused	unused
15 ITEM_TYPE_CONTAINER	capacity in pounds	container flags	key vnum	unused	layers
17 ITEM_TYPE_DRINKCON	total amount of drinks	current amount of drinks	liquid #	component/herb value	junks on use or not
18 ITEM_TYPE_KEY	unused	unused	unused	unused	unused
19 ITEM_TYPE_FOOD	nourishment value	decay timer	FOOD_RAW or FOOD_COOKED, 0 is raw	component/herb value	unused
20 ITEM_TYPE_MONEY	# of coins	coin type	unused	unused	unused
21 ITEM_TYPE_PEN	amount of ink	unused	unused	unused	unused
23 ITEM_TYPE_CORPSE_NPC	unused	unused	decomposition timer	25 * Race Size	unused
24 ITEM_TYPE_CORPSE_PC	unused	unused	unused	unused	unused
25 ITEM_TYPE_FOUNTAIN	unused	Amount of drinks	Liquid Type	unused	unused
26 ITEM_TYPE_PILL	level of spells	spell number 1	spell number 2	spell number 3	unused
27 ITEM_TYPE_BLOOD	unused	quantity	decay timer	unused	unused
28 ITEM_TYPE_BLOODSTAIN	unused	unused	decay timer	unused	unused
29 ITEM_TYPE_SCRAPS	unused	unused	decay timer	unused	unused
30 ITEM_TYPE_PIPE	maximum capacity of	amount of herb in the	herb type	pipe flags	unused

	pipe	pipe			
34 ITEM_TYPE_FIRE	unused	unused	hours left, 0 is dead, -1 is infinite	unused	unused
35 ITEM_TYPE_BOOK	unused	spell number	unused	Language	Skill Level (From 1 to 25)
37 ITEM_TYPE_LEVER	lever trigger flag	vnum of teleport room or spell number or start room or room to be randomised	room to load the mob or object into	object or mob to be loaded	unused
39 ITEM_TYPE_BUTTON	lever trigger flag	vnum of teleport room or spell number	unused	unused	unused
44 ITEM_TYPE_TRAP	trap type	number of reloads	trap trigger	unused	unused
45 ITEM_TYPE_MAP	unused	low room vnum	high room vnum	unused	unused
46 ITEM_TYPE_PORTAL	unused	unused	unused	unused	unused
47 ITEM_TYPE_PAPER	text status	subject status	to status	language number	language skill level
57 ITEM_TYPE_PROJECTILE	unused	weapon flag	weapon flag modifiers	Weapon Type	unused
58 ITEM_TYPE_QUIVER	capacity in pounds	container flags	key vnum	unused	layers
59 ITEM_TYPE_SHOVEL	unused	unused	unused	unused	unused

60 ITEM_TYPE_SALVE	level	Number of uses	unused	herb type	spell slot number
61 ITEM_TYPE_SYMBOL	NO. spell component uses	unused	unused	unused	unused
62 ITEM_TYPE_TRADEGOODS	unused	unused	unused	unused	unused
63 ITEM_TYPE_INSTRUMENT	level of spell	max charges	charges left	spell number	unused
64 ITEM_TYPE_HIDE	unused	unused	unused	mob vnum	race number
65 ITEM_TYPE_CART	capacity	container flags	key vnum	unused	unused
66 ITEM_TYPE_COMPONENT	number of uses for the component and amount of herb	unused	herb type	unused	unused

Scroll Notes

The level of the spell determines the cost. For scrolls that are sold make the spell level high. For scrolls that are found make the spell level low.

Symbol Notes

Builders are not to set any of their objects as type symbol. Type symbol objects have already been set up in the game.

Corpse Notes

Builders should never use the ITEM_TYPE_CORPSE_PC type in objects.

Portal Notes

Builders are not to use ITEM_TYPE_PORTAL in objects. It is used by hard code in the gate spell.

Unused Types Notes

The following types can be found on the FKBIT.LST but are no longer used by the game. Builders should not use them at all. They may eventually be replaced with new types.

```
ITEM_TYPE_FIREWEAPON 6
ITEM_TYPE_WORN 11
ITEM_TYPE_OLDTRAP 14
ITEM_TYPE_NOTE 16
ITEM_TYPE_BOAT 22
ITEM_TYPE_HERB_CON 31
ITEM_TYPE_HERB 32 - Merged with ITEM_TYPE_COMPONENT
ITEM_TYPE_INCENSE 33
ITEM_TYPE_SWITCH 36
ITEM_TYPE_PULLCHAIN 38
ITEM_TYPE_DIAL 40
ITEM_TYPE_RUNE 41
ITEM_TYPE_RUNEPOUCH 42
ITEM_TYPE_MATCH 43
ITEM_TYPE_TINDER 48
ITEM_TYPE_LOCKPICK 49
ITEM_TYPE_SPIKE 50
ITEM_TYPE_DISEASE 51
ITEM_TYPE_OIL 52
ITEM_TYPE_FUEL 53
ITEM_TYPE_SHORT_BOW 54
ITEM_TYPE_LONG_BOW 55
ITEM_TYPE_CROSSBOW 56
```

REPAIR SHOPS

A repairer, like shops, is on the mobile and not the room. Often but not always the repairer is also a shop. Lets use Hilmer of Waterdeep as our example. This would be his entry in the #REPAIRS section of the area file.

```
8131    ITEM_TYPE_ARMOR ITEM_TYPE_CONTAINER SUBSTANCE_METAL
        100   1         5 21    ; Hilmer
```

Lets break this up and look at it in detail:

`8131` - This is the vnum of the mobile (Brian) who is doing the repairs. In your area this number would be a QQ00 vnum.

`ITEM_TYPE_ARMOR ITEM_TYPE_CONTAINER` - The second and third fields are the item types that the repairer can repair. A repairer can repair up to two types of items. If you want to have your repairer just repair one item, then set the second field to 0. You can also have repairers that repair things like treasure, armour and furniture. For a list of item types refer to the Item Types List.

`SUBSTANCE_METAL` - The next field determines the material type that a repairer can repair. There is only one type of material that a repairer can repair. However, these types cover a group of materials. For a list of material types refer to Repair Shops Material Types List. Note that this field recently changed from MATERIAL_TYPE_METAL. Use of MATERIAL_TYPE in the repairs section of an area will crash the game now.

`100 1` - The first number is the profit modifier. At 100 it will cost 1/10th of the the cost of the item to have it repaired. The 1 indicates that it is a repairer. If this was a 2 then the mobile would recharge items like staves and wands. Instead of the 1 or 2 you can put `SHOP_FIX` or `SHOP_RECHARGE`. This is handy if you have many repairers in the area of the two different types.

`5 21` - As for shops this is the hours that the repair is open.

`; the guard smithy` - After the ; you can put a comment, ideally the mobile name that is the repairer.

Note that mobiles can be repairers AND shops. If the shop that sells an item is the one who made the item then it makes sense that they also repair the item they sell.

You must also make the mobile in the `#MOBILES` section of your area file and place the mobile in the `#RESETS` section of your area file. Here is the information for Hilmer of Waterdeep. You can see by his resets that he also sells items. He will have a `#SHOPS` listing as well.

```
#8131
hilmer master armourer~
{70}Hilmer~
{70}Hilmer the Master Armourer is polishing some armour here.
~
{70}He is dressed in fine chain mail so light it would be almost like
wearing clothes.  It is a fine display of his work.  His is a solid
well muscled man from many hours at the forge.  He is bald but for
a fringe of hair around the rim of his head.  His eyes are close
together and look rather mean.
~
U 45 CLASS_WARRIOR RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_CHAIN_MAIL MATERIAL_STEEL
d10+1 0
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
```

Hilmer is sentinel and he has been flagged no shove to prevent characters from shoving him out of his shop.

```
M 0  8131   1  8401; Hilmer in Halls of Hilmer - Master
G 1  8123   60 ;   fine chain mail armour
G 1  8124   60 ;   a fine chain mail coif
G 1  8125   60 ;a pair of fine chainmail
G 1  8126   60 ;  fine chain mail sleeves
G 1  8127   60 ;a pair of fine chainmail
G 1  8094   60 ; chain mail belt
```

If Hilmer was to be just a repairer and sell nothing then he would have no #SHOPS entry and he would have no G resets.

REPAIR MATERIALS

There are five types that repairers will repair. They cover a broad range of materials. Exotic repairers should be very rare and expensive and possibly hard to get to. Gemstone is ideal for jewellers. Leather covers cloth as well.

REPAIR MATERIAL TYPE	BIT VECTOR
SUBSTANCE_WOOD	1
SUBSTANCE_METAL	2
SUBSTANCE_GEMSTONE	4
SUBSTANCE_LEATHER	8
SUBSTANCE_EXOTIC	16

Please note that SUBSTANCE replaces MATERIAL_TYPE in the #REPAIRS section of an area.

RECHARGING MAGICAL ITEM MOBILES

Characters can get wands and staves recharge at places that are set up to recharge them. It works very much like repair shops, but for one field the difference. Make sure to read the lesson on setting up repairers first. Below is a sample mobile that recharges wands and the differences between repair and recharge is pointed out.

First make your mobile who will do the repairing.

```
#QQ24
wanda~
{70}Wanda~
{70}Wanda is showing a customer a wand here.~
She has very pointed features and her vivid red hair is a mass of unruly
curls.  Her body is rail thin, and her robes hang on them very loosely.
~
S 15 CLASS_WARRIOR RACE_HUMAN SEX_FEMALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_STAY_AREA|ACT_CITIZEN
>greet_prog 20~
sayto $n I can recharge your wands for a fee.
~
|
```

Then make sure the repairer is in resets in the room they are supposed to be in. And then add the following to the #REPAIRS section of your area file. It is like the other lines in #REPAIRS but for one aspect.

```
23124 ITEM_TYPE_WAND ITEM_TYPE_NONE SUBSTANCE_WOOD
        100   SHOP_RECHARGE  5 23    ; Wanda
```

Our recharger can recharge wands that are made of wood. Note that the field that is normally 1 in repairers has been changed to SHOP_RECHARGE.

PET SHOPS

Pet shops should be well planned in advance when designing the area, due to the fiddly way they work. A pet shop needs two consecutive room vnums to work right. (Else you will do as I did when I made my first petshop and sell wealthy lady's in Waterdeep as mounts! :) Unlike shops and repairers the pet shop is based on the room and not the mobile. A pet shop will still work even if there is no mob selling them in the room. You can list and buy without a mobile in the room. However it is important for roleplay that you make sure there is a mobile in the room. The mobile is also handy to restrict who can buy from the shop if you want restrictions in place.

Let us say we have allocated vnums `QQ50` and `QQ51` to our petshop.

`QQ50` is the room where the player types list to see what pets there are and buys the actual pet. This room needs to be flagged ROOM_PET_SHOP. This is the room that the mobile shopkeeper will load up into. Even though codewise he is not needed, he is needed roleplaywise.

`QQ51` is the storeroom for the pets that will be for sale in the pet shop. This room should be flagged `ROOM_NO_ASTRAL` and `ROOM_NO_SUMMON`. It should also have no way of being got into or ways out. Basically it is a room that players have no business being in and it is up to you the builder to prevent that from happening.

Next you need to make the mobiles of the pets/mounts you want to sell in your shop. You will also need to make your seller mobile. Make the pets the level that you want players to able to buy them from. Note that flying pets should be higher level. Lets say for this example we have the following mobs to be involved with the petshop.

- `QQ01` - Stable Master
- `QQ02` - A large pony
- `QQ03` - A small grey mule

You would have to set them up in resets as follows:

```
M 0  QQ01  1  QQ50; the stable master the Stables

M 0  QQ02  1  QQ51; a small pony in stables storeroom.

M 0  QQ03  1  QQ51; a large horse in stables storeroom.
```

Let us use a real example in the game , the Palace Stables in Waterdeep and put all the elements needed to make a working pet shop in below:

First the rooms. Note they are 2 consecutive vnums. The description of the storeroom warns players they should not be here and alert an imm. Note that only the first room is flagged petshop. It tells the code that the next vnum will be a pet storage room. Any mobiles loaded into the next vnum will be sold in the first room.

```
#8077
The Palace Stables~
{30}This is a reputable stable where a traveller can stable his
horse safely.  For a price of course. The stable master will
sell good quality mounts and teach one how to ride.
~
0 ROOM_INDOORS|ROOM_PET_SHOP SECT_INSIDE 0 0 0
DDIR_NORTH
~
~
0 -1 8122 1
DDIR_EAST
~
~
0 -1 8013 1
DDIR_SOUTH
~
~
0 -1 8113 0
S
#8078
Petshop storeroom.~
{30}This is the petshop storeroom.  You shouldnt be in here if you are.
There is pet poop all over the ground and it will rise up and get
you if you are in here without permission.
~
0 ROOM_NO_MOB|ROOM_INDOORS|ROOM_NO_SUMMON|ROOM_NO_ASTRAL SECT_INSIDE 0 0 0
S
```

Then we add our seller. This one has programs on him to stable characters mounts. That requires another room. See the lesson about stabling mounts for more information.

```
8077
stable master~
{30}the stable master~
{30}The stable master tends to his horses here.
~
He is dressed simply in cloth leggings and tunic with long leather boots.
```

```
He is a rider of some expertise and for a price he would be willing to
teach one how to MOUNT and ride a horse.
~
U 25 CLASS_FIGHTERS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 500
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
%8 1 mount~
>greet_prog 100~
sayto $n Can I interest you in a fine horse?
sayto $n Or I can teach you how to ride a horse?
sayto $n Or can I stable your mount for a small fee?
~
>intercept_prog sleep~
sayto $n You cannot sleep here.
sayto $n Go pay for a room in an inn.
~
>speech_prog yes aye~
say Which do you want?
~
>speech_prog stable~
sayto $n I can stable your mount safely
sayto $n for 4 gold pieces.
~
>bribe_prog 400~
mptransfer $n 8858 pet
mpat 8858 mptransfer $n 8077
sayto $n Here you go.
sayto $n Give me this bit of wood when you want your mount back.
mpechoat $n $I hands you a piece of painted wood.
mpechoaround $n $I hands $N a piece of painted wood.
mpechoat $n $I leads your mount off to a private stall.
mpechoaround $n $I leads $N's mount off to a private stall.
mpoload 8619
mpgive i8619 $n
~
>give_prog i8619~
sayto $n Rightothen, let me get your mount.
mpecho $I heads out to the back of the stables.
mptransfer $n 8858
```

```
mpat 8858 mptransfer $n 8077 pet
mpechoat $n $I leads your mount out to you.
mpechoaround $n $I leads $N's mount out to $m.
mpjunk i8619
~
|
```

Then we need our pets/mounts to sell. We have a pony and a large horse for sale here. Note that they poop!

```
#8037
small pony~
{F0}a small pony~
{F0}A small pony is here.
~
It is a small white pony with a black patch on its rump and over
one eye.
~
S 15 CLASS_MONSTER RACE_HORSE SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_MOUNTABLE|ACT_WIMPY
>rand_prog 1~
if sector($i) == 1
  mpecho $I cocks his tail and poops!
  mpoload 8580
  mpquiet on
  drop i8580
  mpquiet off
endif
~
|
#8038
large horse~
{30}a large horse~
{30}A large horse is here.
~
It is a brown large horse with a black tail and mane.
~
S 20 CLASS_MONSTER RACE_HORSE SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_MOUNTABLE|ACT_WIMPY
>rand_prog 1~
if sector($i) == 1
  mpecho $I cocks his tail and poops!
  mpoload 8580
  mpquiet on
```

```
  drop i8580
  mpquiet off
endif
~
|
```

Then we need to place the seller and mounts into resets.

```
M 0  8037   1  8078;             a small pony in Petshop storeroom.
M 0  8038   1  8078;            a large horse in Petshop storeroom.
M 0  8077   1  8424;   stable master in gentle rest stables
```

Banks

Unlike shops, setting up a bank is very simple. It is done on the mobile, and it is a simple ACT flag. ACT_BANK.

```
#8048
bankermob waterdeepbanker~
{70}the banker~
{70}A banker is here counting gold.
~
He is a very well dressed man with a portly build.  His hair is sparse
and limited to a fringe around his head.  He looks at you with piercing
blue eyes.
~
U 25 CLASS_WARRIORS RACE_HUMAN SEX_MALE POS_STANDING DEITY_NONE
ACT_SENTINEL|ACT_BANK|ACT_NOSHOVE|ACT_CITIZEN
0
ARMOR_TYPE_BANDED MATERIAL_BRASS
d15+15 1000
13 13 13 18 13 18 13
0 0 0 0 0
LANG_COMMON
LANG_COMMON
RIS_NONE RIS_NONE RIS_NONE
```

RESETS

MOBILE RESETS

When the game reboots every area is reset. Over time the areas are reset again. Time between resets varies from area to area, and is defined in the #FLAGS section of the area file. For a mobile you set where (what room) it loads into, how many of the mobile loads, objects it holds in inventory and equipment it is wearing.

Mobile placement

This is how you set where the mob is to load and how many of the mob can load in the game. The syntax would be as follows:

`M 0 mob-vnum limit room-vnum ; comment`

An example Mobile Placement reset would be:

```
M 0 QQ01 1 QQ27 ; dwarf in room 27
```

`M` - This defines that it is a mobile the game is dealing with.

`0` - This field isnt used but needs to be there to for the mud to load.

`QQ01` - This is the vnum of a mobile to load.

`1` - This is the limit of how many of this mobile may be present in the room. If for instance you put in 4 instead of 1 here, on boot up one dwarf would load. Providing he isn't killed, next repop another dwarf would load, continuing on until there are 4 all up of that vnum in that room.

`QQ27` - This is the vnum of the room where the mobile is loaded.

`; dwarf in room 27` - This is the comment. It is a good idea then to add comments after a ; Comments make it much easier to see what is happening in resets when reading over the area file. Especially as you are editing and adding to your area file. We require well commented resets on Forgotten Kingdoms Mud.

So in summary this reset says that it is a mobile placement reset for the mobile with the vnum of QQ01 into room QQ27 and only 1. The comment gives us the information that it is a dwarf.

Equiping a mobile

You place this line directly under the M line in resets. You must set the mobile in place before equiping it. The syntax is as follows:

```
E 0 obj-vnum 0 wear_loc ; comment
```

And expmple Equipment reset would be. Note that the M placement reset is here too. It is needed for the code to know which mob it is equipping. It makes the order that you place things in resets very important.

```
M 0 QQ01 1 QQ27 ; dwarf in room 27
 E 0 QQ04 10 WEAR_HEAD ; helmet on dwarf
```

`E` - This tells the game you are equipping the mobile before this line.

`0` - This field is not used so should be just set to 0.

`QQ04` - This is the vnum of the object that the mobile is to wear.

`10` - This is the total number it allows in the game before it does not equip the mobile with that item. Due to the fact equipment can be worth quite a bit we like to keep this reset number low so as to not fill an area with a lot of equipment that is not used and just leads to players making too much money on the sale of it.

`WEAR_HEAD` - This tells the game where on the mobile the object is to be worn. In this case on the head. Refer to the reset wear locations listing for more information on wear mobiles can wear objects. Note that they are different from the CAN_WEAR_ locations on objects

`; helmet on dwarf` - After the ; a comment should be placed.

Objects in mobiles inventory

This command gives a mobile the object specified and places it in their inventory. The command works for the mobile that is listed before it. For shop keepers you need to use this command to give them what they sell. The syntax is as follows:

```
G 0 obj-vnum 0 ; comment
```

The following is an example. Note that we have used the two resets from before of our dwarf wearing a helmet.

```
M 0 QQ01 1 QQ27 ; dwarf in room 27
 E 0 QQ04 10 WEAR_HEAD ; helmet on dwarf
```

```
G 0 QQ27 10 ; purse in dwarfs inventory
```

`G` - This tells the game that you are giving the mobile that preceeds it this object. You can give the mobile more than one object.

`0` - This unused by the game but the 0 must be put in place for the game to load right.

`QQ27` - This is the vnum of the object.

`10` - This is the maximum number of that object that can be in that game before it is loaded up in that mobiles inventory. If there are 11 in the game on reseting the area then the mobile will not load up with it. If the mobile is a shop he will load up as having it in his list command, but he will tell the player that they are out of stock when they go to buy the item.

`; purse in dwarfs inventory` - The comment should state what the name of the item is at the very least.

A mobile can be equiped with more than on item, providing it is a unique wear location. Mobiles cannot layer what they wear. Each item would have to have its own seperate E reset. They can have more than one G reset as well.

MOBILE RESET WEAR LOCATIONS

Please note that these are different to the CAN_WEAR locations on objects. A mobile can only have one item in each wear location. Mobiles cannot layer armour.

WEAR_FINGER_L
WEAR_FINGER_R
WEAR_NECK_A
WEAR_NECK_B
WEAR_BODY
WEAR_HEAD
WEAR_LEGS
WEAR_FEET
WEAR_HANDS
WEAR_ARMS
WEAR_WAIST
WEAR_WRIST_L
WEAR_WRIST_R
WEAR_LEFT_HAND
WEAR_RIGHT_HAND
WEAR_BOTH_HANDS
WEAR_EARS
WEAR_EYES

OBJECT RESETS

Objects put into a mobiles inventory and worn by mobiles are covered in Mobile Resets Lesson.

Loading an object into the room

This defines the room that object is loaded into and how many. The syntax to do this follows:

```
O 0 obj-vnum 0 room-vnum ; comment
```

The following is a sample of this:

```
O 0 QQ00 1 QQ78 ; Chest into room 78
```

`O` - This tells the game that an object is to be loaded.

`0` - This is unused by the game but needs to be here in order for the game to load.

`QQ00` - This is the vnum of the object to be loaded.

`1` - This is how many of this object can be in the game before it can loaded.

`QQ78` - This is the room that the object will be loaded into.

`; Chest into room 78` - This is the comment.

The object is not loaded if the target room already contains any objects with this vnum. The object is also not loaded if any players are present in the area.

Place an object into a container

This reset can be fiddly. Be sure to test this part in your area when it is loaded onto the testport to make sure you have it correct.

```
P 0 obj-vnum 1 obj-vnum ; comment
```

Our sample places a gem into the chest that we loaded into the room sample above.

```
O 0 QQ00 1 QQ78 ; Chest into room 78
 P 0 QQ01 10 QQ00 ; Placing gem into chest
```

P - This tells the game that this is going to be a 'PUT' reset.

0 - 0 is the default state. If this field is set to 1, then the object will be hidden inside the container, and will need to be searched for to be found, or seen with detect hidden.

QQ01 - This is the vnum of the object being placed into the chest.

1 - This is the number of gems that can be in the game before it wont load.

QQ00 - This is the vnum of the container that the gem is going to be put into.

; Placing gem into chest - This is the comment.

For the P reset, the second obj-vnum is the vnum of a container object where the object will be loaded. The actual container used is the most recently loaded object with the right vnum; for best results, there should be only one such container in the world. The object is not loaded if no container object exists, or if someone is carrying it, or if it already contains one of the to-be-loaded object.

Hiding an object in a room

The object will be hidden untill the player "searches" for it. Our sample below is a key that is hidden in the room.

H 0 obj-vnum 0 ; comment

```
H 0 QQ04 5 QQ03 ; Key Hidden in the room
```

H - This tells the game that this is a Hidden reset.

0 - This is unused, but needs to be there in order for the game to load.

QQ04 - This is the vnum of the object to be hidden.

5 - This is total number of the object that can be in the game before it stops loading.

QQ03 - This is the vnum of the room that the object will be hidden in.

```
; Key Hidden in the room
```

Burying an object in a room

The object will be buried untill the player "digs" for it. Our sample below is a key that is buried in the room.

```
U 0 obj-vnum 0 ; comment
```

```
U 0 QQ04 5 QQ03 ; Key Buried in ground
```

U - This tells the game that this is a Buried reset.

0 - This is unused, but needs to be there in order for the game to load.

QQ04 - This is the vnum of the object to be buried.

5 - This is total number of the object that can be in the game before it stops loading.

QQ03 - This is the vnum of the room that the object will be buried in.

```
; Key Buried in the room
```

DOOR RESETS

When you create an exit in the #ROOMS section you can set a door to be closed and locked or whatever state you wish. Characters can come along and open and close and lock doors as they wish, and unless a reset is defined the door will be left as the character left the door. If you want your door to shut you can either write a program on a mobile to shut and or lock it (a more IC approach), or you can use resets to set the door state to what you want it to be. This will reset the door state each copyover reguardless of what characters have done with the door.

Door states

This will determine if a door is locked and other states whenever an area resets. The syntax is as follows:

```
D 0 room-vnum door state ; comment
```

The following is a sample of a door that is to be locked each area repop.

```
D 0 QQ00 1 3 ; Door to the east is locked
```

`D` - This tells the game that this is going to be a door reset.

`0` - This is not used by the game but is needed in order for the area to load.

`QQ00` - This is the vnum of the room with the door.

`1` - This is the direction of the door.

`3` - This is the state of the door using the bit vector.

`; Door to the east is locked` - This is the comment.

The QQ00 is the vnum of a room. The next number is a door number from 0 to 6, door direction numbers. See the table below for door direction numbers. The final number indicates how to set the door by default. Door states are: 0 open unlocked, 1 closed unlocked, 3 closed locked.

Room exits must be coherent: if room 1 has an exit to room 2, and room 2 has an exit in the reverse direction, that exit must go back to room 1. This doesn't prevent one-way exits; room 2 doesn't HAVE to have an exit in the reverse direction.

Random rooms

Resets can be used to make a maze, that will change constantly.

```
R 0 room-vnum last-door ; comment
```

```
R 0 QQ08 4 ; two dimensional maze in room 8
```

For the 'R' command, the room-vnum is the vnum of a room. The last-door is a door number. When this command, the doors from 0 to the indicated door number are shuffled. The room will still have the same exits leading to the same other rooms as before, but the directions will be different. Thus, a door number of 4 makes a two-dimensional maze room; a door number of 6 makes a three-dimensional maze room.

DOOR STATES

DOOR STATES	BIT VECTOR
DOOR_OPEN_UNLOCKED	0
DOOR_CLOSED_UNLOCKED	1
DOOR_CLOSED_LOCKED	2
DOOR_NONE	3

DOOR DIRECTION BIT VECTORS

DOOR DIRECTION	BIT VECTOR
DIR_NORTH	0
DIR_EAST	1
DIR_SOUTH	2
DIR_WEST	3
DIR_UP	4
DIR_DOWN	5

ROOM RESETS

You can use the B reset to set bits on exit flags, room flags or mob affects. It allows you to set, remove, or toggle a room flag on a room

Syntax: `B 0 room# bitType SET|REMOVE|TOGGLE`

Sample: `B 0 8030 BIT_RESET_ROOM|BIT_RESET_TOGGLE ROOM_NO_ASTRAL`

TRAP RESETS

See the lessons on making traps, trap triggers and trap types for more information on using traps in your area. A current list of traps can be seen by typing `showtraps` on the testport with your builder character. More information about the trap can be seen by typing `showtrap #`.

```
T [1|0] TTYPE_ # TRIGGER|TRIGGER
```

If placed after a D reset it is applied to the exit. If placed after an object (O,G,E,P,H) reset its applied to the object.

`[1|0]` 1 is reset after repop. 0 is never reset after the first time.

`TTYPE_` is the trap type. We can add these as easily as spells or races.

`#` is the number of times it can be triggered per reset.

`TRIGGER` is the things that trigger it. See the Trap Triggers lesson for a list of Trap Triggers. If two of the triggers are used it will only trigger once. So when coded, the lowest level trigger should be used, ie no need to use TRIGGER_PICK and TRIGGER_UNLOCK because anything that triggers PICK will also trigger unlock.

Example Trap

```
T 1 TRAP_MINOR_SPIKE TRIGGER_GET|TRIGGER_OPEN
```

The T reset can be placed after a door reset, and it will be on that door, or it can be placed after a O reset and the trap will be on that object.

Refer to the door and object resets lessons for more information.

The trap will trigger after anything is done with the door or object: get, open, shove etc. No door needed for door resets so will trigger on move, open door, unlock etc.

The trap will reload once.

Thieves and perhaps some other guilds are able to set traps in the game with the detrap skill.

Builders can make traps for sale in the game. See the other trap lessons for more information.

If you as a builder cannot find a trap that you want in the game, you can submit a trap design to the builders admins in the following format. Use the example as your base.

```
Name a strong negative energy trap~
Type TTYPE_NEG_ENERGY_STRONG
DamageType SD_HEALING
DNum 25
DType 8
Target TTARGET_CHAR
Learnmod 8
Expmod 60
Level 40
Skill blindness~
```

Use either the dnum (number of damage dice) and dtype (number of sides on the dice) or the skill (spell or skill coded into the game). A trap will either do specific damage or utilise a coded spell or skill in the game to do its damage/affects.

MOBILES OBJECTS AVAILABLE FOR USE IN ANY AREA

Below is a link to a list of vnums of objects that are in limbo.are and waterdeep.are. They can be used as spell components or as objects in quests. They can be used to stock your shops with general items. It should be noted that recall potions and scrolls are only to be of the vnum listed here. Area builders are NOT to create their own recalls. Please note you can only load objects from limbo.are and waterdeep.are into resets in your areas. If you try to use objects from other areas in the game you will crash the game. We have insured that these two areas always load up before any others. Please use this list to stock your area and not the list that is generated by olist on the testport. There are some quest items etc in the Waterdeep area, and some specialist objects for hard code that we do not want for sale etc in other areas. We did have an incident where a builder sold quest rewards from Waterdeep quests in her area and thereby ruined the value of the quest items.

The list also lists mobiles from limbo.are that are available to be used in any area. Please make sure to use only mobiles from this list, so that you are not accidently using mobiles that have been coded for familiars and companions.

Mobs and Objects List The list is up to date as of the date last edited on the bottom of this page. The objects in this list can be used in resets and loaded up in programs.

To follow is a list of objects that can ONLY be loaded up in programs. If they are put in resets, they will crash the game. But you can freely use them in programs. The list was compiled by Dalvyn into handy categories. There are a few vnums in here that are from limbo that can be used in resets, but check the other listing before putting anything in resets.

HUMANOIDS

```
11018 - goblin toe
11019 - hobgoblin toe
11369 - rakshasa eye
80    - sea elven elf ear ears
44    - elf ear
```

GIANTS

```
11366 - fomorian giant's ear
11367 - cyclops eye
11368 - ettin nail
```

ABERRATIONS/DRAGONS

```
11022 - vial of xorn saliva
11459 - naga scale
12330 - purple naga scale
```

```
11331 - black naga scale
11443 - beholder eyestalk
12013 - carrion crawler tentacle
12014 - multifaceted hook horor eye
11713 - blue umberhulk chitine
11023 - brown umberhulk chitine
12016 - black umberhulk chitine
11371 - bulette chitine
29469 - large bulette chitinous carapace
29470 - aranea leg
29473 - cold thoqqua tail
17068 - piece of phaerimm's skin
297   - wyvern tail
7503  - flask of dragon blood
```

MONSTROUS HUMANOIDS

```
11329 - piece of greenish yuan-ti skin
11330 - greenish yuan-ti scale
11331 - yuan-ti claw
11332 - yuan-ti eye
11333 - tip of a yuan-ti tail
15163 - sahuagin fin
12002 - piece of illithid brain
12015 - handful of quaggoth hair
70    - A harpy's voice
289   - minotaur horn
296   - ettercap fang
2347  - a kuo-toa webbed foot
```

UNDEAD

```
11020 - skeleton knuckle
11021 - long ghoul tongue
11242 - skeletal guardian's skull
11457 - bonebat's dessicated wing
11460 - lich skull
11541 - a piece of mummy wrappings
11542 - vampire fang
290   - piece zombie flesh
53    - bone
13960 - dracolich claw
13961 - handful of white wight hair
```

ANIMALS/BEASTS/VERMINS/INSECTS/MAGICAL ANIMALS

11146 - black bat wing
11211 - piece of spider flesh
11458 - piece of hairy red bat leather
11543 - vampiric bat fang
11755 - water beetle chitine
29471 - piece of brain spider
29 - bat guano droppings
38 - fur
63 - feather
65 - scale
284 - adders stomach
288 - scorpion tail
291 - piece beetle chitine
292 - snake tongue
293 - crab meat
294 - bear claw
2617 - wolf tail

CONSTRUCTS

11136 - mummified finger (from a crawling claw)

OUTSIDERS

11137 - piece of brown dao skin
11370 - piece of efreeti skin
11953 - steam mephit arm
11954 - magma mephit leg
11955 - ice mephit head
11956 - lightning mephit torso
11522 - piece of lemure flesh
11523 - torn imp wing
11525 - piece of hellcat skin
11526 - barbazu's pointed ear
11527 - cornugon scale
17011 - dretch tooth
17012 - quasit spiky horn
17013 - succubus eye
17014 - bebilith fang
17015 - retriever claw
17016 - vrock feather
17017 - hezrou hand
17018 - glabrezu head
17019 - nalfeshnee wing

17020 - marilith scale
17058 - balor claw

OOZES

11138 - piece of grey slime
11239 - piece of crystal ooze
11240 - piece of brown slime
11215 - piece of green slime
11241 - piece of gelatinous cube

COIN RESETS

When you make a mobile an unique mobile they will not load up with coins automatically. You need to give the mobile its coins in resets. Sometimes you simply may choose for a mobile to not have coins. You will also need to make sure it is IC for the mobile to carry coins. For instance most animals in the game will not carry coins, as coin does not have any meaning to them.

```
M 1 4020 2 4025; city guard
C 1 COIN_SILVER 2 6
```

Above is an example of the coins resets for an unique guard mobile. T he coin reset has to go directly after the mobs resets. It is similar to giving it an object for its inventory. Let us break this command up:

`C` - The `C` denotes that this is a coin reset.

`1` - This next number is not used by the game but must be here for the area to load.

`COIN_SILVER` - This is the type of coin. Refer to the table below to see the coin type.

`2 6` - These next two numbers (are the dice used to work out the amount of coins. In this case its 2d6 of this coin type.

If you wished you could have the mobile load up different coin types.

COIN_COPPER	0
COIN_SILVER	1
COIN_ELECTRUM	2
COIN_GOLD	3
COIN_PLATINUM	4

SPECIALS

MOBILE SPECIALS

This section defines special functions (spec-fun's) for mobiles. A spec-fun is a hard coded function which gives additional behavior to all mobiles with a given vnum, such as the justice implementing guard or the beholder casting spells in combat. Special functions are a holdover from older mud versions (like Diku). In FKMud mob programs are often better to use than the spec_functions, as they give you much more flexibility and customisation.

However spec-funs are a simple shortcut if you want your mob for instance to cast certain spells in combat, or use a breath weapon.

The spec_guard function is used by the Justice system to have mobiles with this function to carry out the punishments etc of the justice system.

Setting a mobiles class to a specific guild or deity will have the mobile act in a way that is common to that guild or priest of that deity, when standing around and when in battle.

One of the common uses of these special functions is the healers in the training temples. They have the spec_cast_adept function. Types of functions that can be used are:

spec_fido	Eats corpses
spec_cast_adept	For constantly healing mobs
spec_breath_fire	Uses fire breath weapon
spec_breath_frost	Uses frost breath weapon
spec_breath_acid	Uses acid breath weapon
spec_breath_gas	Uses gas breath weapon
spec_breath_lightning	Uses fire lightning weapon
spec_breath_any	Uses a random breath weapon in battle
spec_poison	Poisons foe with a bite
spec_guard	For guards in the justice system

spec_cast_cleric	General Cleric in battle
spec_janitor	Cleans up trash and drinks
spec_cast_undead	Casts curse, drain type spells in battle
spec_executioner	Old justice for dealing with killers and thieves. Do not use.
spec_judge	For judges in the justice system
spec_pet_gen	Looks for master, rests when master does
spec_paladin_warhorse	For a paladins warhorse, looks for master, aids, rescues, alerts to evil, doesn't tire
spec_pet_hawk	Will work for all pet birds
spec_pet_dog	Does spec_pet_gen plus also rescues master, race echos, sniffs invis chars
spec_pet_panther	For companion large cats, see dog
spec_pet_bear	For companion bears, see dog
spec_pet_wolverine	For companion wolves, see dog

The last 7 specials listed are made for PC pets. They will seek out their master, will land when their master does and often step in front of their master in battle if they feel their master is being threatened. The paladins warhorse is sensative to alignment of those PC's around it.

You would set up a mobile in the #SPECIALS section like follows.

```
#SPECIALS M QQ00 spec_thief
M QQ06 spec_cast_cleric
S
```

QUESTS

QUEST LOG

This section goes before the #MOBILES section in your area file, after all the other headers.

The questlog command in the game shows players what quests their character has completed, or is in the process of doing. The information for this command is generated by the #QUESTS section of the area file. This information is also handy for builders to see at a glance what their quests do and what the quest bits are. It is also used by game administrators to help reset quest bits for characters affected by bugged quests and so on.

```
#QUESTS
8000 10 4 1 7 {E0}You are helping Jonathon the Armourer in Waterdeep.~
8000 10 4 2 2 {E0}Jonathon needs a polished metal shield.~
8000 10 4 3 3 {E0}Jonathon needs a hardened leather helm.~
8000 10 4 4 4 {E0}Jonathon needs a tapestry vest.~
8000 10 4 5 5 {E0}Jonathon needs topaz earrings.~
8000 10 4 6 6 {E0}Jonathon needs fine chainmail armour.~
8000 10 4 7 7 {E0}Jonathon needs a wizards hat.~
8000 10 4 8 8 {A0}You have helped Jonathon the Armourer of Waterdeep stock his shop.~
-1
```

This is the quest journal for a quest with Jonathon the armourer in Waterdeep. Lets break it up.

`8000` - This is the vnums of the area, that the quest bits are checked in.

`10 4` - These are the quest bits that are being checked. This quest is using quest bits from 10 to 4. For more information on how this works refer to the quest bits tutorial.

`2 2` - This is the step. Step 2 of quests bits 10 to 4, this is where the PC has to find a polished metal shield. The first line uses 1-7 and while the bits equal that, it will show that line as well.

`{E0}Jonathon needs a polished metal shield.~` This is the comment that the PC sees when they look at their quest log. The colour is important as this is what the questlog command uses to work out what is current and so on. Each comment must be finished with a tilda.

Colour codes - Event codes

- `{F0}` - Special Event or Misc.
- `{E0}` - Quest in progress.
- `{A0}` - Completed quest.

`{90}` - Failed quest.

- `{20}` - Knowledge Geography etc.
- `{30}` - Trade learned.
- `{B0}` - Quest not yet started.

MUD PROGRAMS

MOBILE, OBJECT AND ROOM PROGRAMS

Otherwise known as mob progs, these programs are used to give life to your mobiles and your area in general. In the lessons to follow is a lesson for each type of program we have available for use on Forgotten Kingdoms. Many of the samples with real programs from the game. We will never release a whole area for download, but we will release parts to aid builders to learn how to code mob progs.

This particular lesson is not a lesson in how to write programs. This is a listing of what programs are available and a brief explanation of what they do. We will link to a lesson for each mud program type to give more detail on how to use them in your area.

Some programs will only work on mobiles, others only on objects and so on. The following tables list which programs will work for each of the three things that programs can be placed on, mobiles, objects and rooms.

Progs that work on mobiles

The following is a list of programs that will work on mobiles.

Trigger	Triggering Variable	Explanation
act_prog	keyword / phrase	Works for emotes socials actions bamfs
speech_prog	keyword / phrase	Works on says from same room as mob
rand_prog	percentage	randomly triggered based on percentile
fight_prog	percentage	random within a fight - percentile
hitprcnt_prog	percentage	percent is percentage of mob's max H.P.
greet_prog	percentage	entry that mob can see - by mob/player
entry_prog	percentage	when the mob itself enters a room
bribe_prog	amount of gold	when a player gives the mob money
death_prog	percentage	when the mob dies
script_prog		loops by line Hour triggers start

time_prog	hour and minutes	script prog - runs once on hour
hour_prog	hour	loops as Script for an hour from start
intercept_prog	keyword	when a player types a command
give_prog	item vnum ivnum	when character gives mobile certain object
arrival_prog	percentage	activates when mobile who is using mpwalkto arrives at destination
buy_prog	item vnum	activates when the mobile sells that vnum
injure_prog	percentage	activates when the mobile is injured

Progs that work on objects

The following list of programs will work on objects.

Trigger	Triggering Variable	Explanation
wear_prog	percentage	when a player wears the object
remove_prog	percentage	when a player removes the object
speech_prog	keyword/phrase	says or tells from same room as object
rand_prog	percentage	randomly triggered based on percentile
sac_prog	percentage	when a player sacrifices the object
zap_prog	percentage	when the player is zapped - alignment
get_prog	percentage	when a player gets the object
drop_prog	percentage	when a player drops the object
damage_prog	percentage	when the object is damaged
repair_prog	percentage	when the object is repaired

greet_prog	percentage	when a mob/player enters the room
exa_prog	percentage	when the object is Examined or Looked
push_prog	percentage	when a player pushes an object
pull_prog	percentage	when a player pulls an object
use_prog	percentage	See below for more information
intercept_prog	keyword	when a player types a command

Notes on the use_prog program

The use_prog will be executed when the item is used. To define what will trigger the use_prog, here is a list:

- Wands - when you zap
- Staves - when you brandish
- Food/pills - when you eat
- Blood/fountains/drink containers - when you drink
- Lights/armour/weapons - when you wear/wield/hold
- Potions - when you quaff

When the use_prog is executed you will not see the standard 'use message' (ie You quaff a violet potion). What you will will be any mpechoes placed in the program. Also, any mobprog commands can be used in use_prog.

NOTE The use_progs on lights, armour, and weapons are executed BEFORE the player wears/holds/wields any item. Therefore, a mpforce to remove the item will not work in the use_prog. You will have to use a wear_prog to accomplish this.

Room Programs

The following is a list of programs that will work in rooms.

act_prog	emotes socials actions bamfs
speech_prog	says or tells from same room as mob
rand_prog	randomly triggered based on percentile
fight_prog	random within a fight percentile

greet_prog	entry that mob can see by mob/player
entry_prog	when the mob itself enters a room
leave_prog	when the PC leaves a room
death_prog	when the mob dies
script_prog	loops by line. Hour triggers start
time_prog	script prog runs once on hour
hour_prog	loops as Script for an hour from start

IF CHECKS

We are often adding new if checks as we or one of our builders finds the need for one. So if you are wanting to check for something and it doesn't appear to be on this list, please post to the builders discussion forum, and the coders will consider it.

If check	Function and Syntax
`actorhasobjnum`	checks if the PC has the vnum in parenthesis worn or in inventory `if actorhasobjnum(8000)`
`actorotypewear`	checks if the PC is wearing the type of item in the parenthesis `if actorotypewear(1)` Use the bit vector not the word.
`align`	checks $whatevers alignment with rhs `if align($n) > 1000`
`canhire`	checks if $whatevers can hire an employee for a dwelling `if canhire($n)` This is only used in the dwellings areas, not in normal areas.
`canpkill`	checks if $whatevers can pkill `if canpkill($n)` All characters on FKMud can pkill so this check is not used.
`cha`	checks $whatevers charisma with rhs `if cha($n) > 15`
`charinroom`	checks that the string $1 can be seen in the room by $n `if charinroom($n) == $1`
`clan`	checks $whatevers clan with rhs Organisations are not hard coded in FKMud. Check for the specific organisation objects instead. `if clan($n) == Clanname`
`class`	checks $whatevers class with rhs `if class($n) == Wizard`

	`if class($n) == Warrior`
	`if class($n) == Priest`
	`if class($n) == Rogue`
	Only checks the 4 base classes, no guilds.

`con`	checks $whatevers constitution with rhs `if con($n) > 15`
`day`	checks day number with rhs `if day() == 1`
`deity`	checks $whatevers deity with rhs `if deity($n) == Mystra`
`dex`	checks $whatevers dexterity with rhs `if dex($n) > 15`
`doingquest`	checks $whatevers questbits with rhs Not used. Use if quest or if questr instead.

`economy`	checks the current economy with rhs
	`if economy(7500) > 1000`
	Value is in copper. Start Vnum of area is in parenthesis.
	`if economy() > 1000`
	If no vnum is specified it will check the area that the prog is being activated in.

`favor`	checks $whatevers favor with rhs `if favor($n) > 400`
`feat`	checks if $whatever has featname trained rhs times `if feat(featname, $n) > 0`
`formation`	checks if $whatever is in a certain position of the formation `if formation($n) == 1`
`getcurrentuses`	Checks how many uses $whatever of an ability currently has `if getcurrentuses(shapechange,$n) > 1`
`goldamt`	goldamt counts the greater of what is in $whatever's inventory or money pouch `if goldamt($n) > 500` [amount in copper coins] It checks what has the greatest amount, inventory or money pouch.

goldamtroom	goldamtroom compares rhs with amount of gold in the room that $whatever is in `if goldamtroom($n) > 500` [amount in copper coins]
goldamtroomc	goldamtroomc compares rhs with amount of gold in the room and containers in the room that $whatever is in `if goldamtroomc($n) > 500` [amount in copper coins] If the check is on $i it will look through the money pouch on $i
glory	checks the amount of CURRENT glory $whatever has with the rhs `if glory($n) > 20`
glory_total	checks the TOTAL amount of glory $whatever has with the rhs `if glory_total($n) > 20`
group	checks the number of people in $whatever's group with the rhs `if group($n) > 2`

guild	checks $whatevers guild with rhs
	`if guild($n) == Necomancers` [plural form]
	For priests use the prefix Clerics of
	`if guild($n) == Clerics of Tyr`
	Exception is
	`Druids of Chauntea`

hashorse	checks if $whoever has a horse `if hashorse($n)`
haspet	checks if $whoever has a pet `if haspet($n)`
hitamt	checks $whatevers hps with rhs `if hitamt($n) > 150`
hitprcnt	checks $whatevers hp percentage with rhs `if hitprcnt($n) > 50`
hometown	checks hometown of $whatever with rhs `if hometown($n) == Waterdeep` or `if hometown($n) == 8030` [vnum]
inarea	checks if $whatever is in the area

	`if inarea($n) == 7500` [Area start vnum]
`inroom`	checks if $whatever is in the rhs room `if inroom($n) == 8030` [vnum]
`int`	checks $whatevers int with rhs `if int($n) > 15`
`isaffected`	checks if $whatever is affected by rhs `if isaffected($n) == Detect_magic`
`isburied($o)`	checks if $o (object) is buried or not `if isburied($o)`
`ischaotic`	checks if $whatever is of chaotic alignment `if ischaotic($n)` Is true for CG CN and CE.
`ischarmed`	checks if $whatever is affected by charm `if ischarmed($n)`
`isclanned`	checks if $whatever is in a guild `if isclanned($n)` Clans are not currently hard coded on FKMud.
`isdevoted`	checks if $whatever is devoted to a deity `if isdevoted($n)`
`isemployer`	checks if $whatever is the mobs employer `if isemployer($n)` Used by dwellings only.
`isevil`	checks if $whatever is < -350 align `if isevil($n)` Is true for CE NE and LE.
`isfamilar`	checks if $whatever is a familiar `if isfamiliar($i)`
`isfight`	checks if $whatever is fighting `if isfight($n)`

`isfollow`	checks if $whatever is following `if isfollow($n)`
`isfullmoon`	checks to see if there is a fullmoon `if isfullmoon()`
`isgood`	checks if $whatever is one of the three good alignments `if isgood($n)` Is true for LG NG and CG.
`isguilded`	checks if $whatever is in a guild `if isguilded($n)`
`ishelled`	checks if $whatever is in hell `if ishelled($n)`
`isimmort`	checks if $whatever is immortal `if isimmort($n)`
`isindoors`	checks if $whatever is indoors `if isindoors($n)`
`islawful`	checks if $whatever is of lawful alignment `if islawful($n)` Is true for LG LN LE.
`ismobinvis`	checks if $whatever is mobinvis `if ismobinvis($n)`
`ismounted`	checks if $whatever is mounted `if ismounted($n)`
`isneutral`	checks if $whatever is of one of the three neutral alignments `if isneutral($n)` Is true for CN TN NG.
`isnpc`	checks if $whatever is a NPC `if isnpc($n)`
`isordered`	checks if $whatever is in an order `if isordered($n)`

	Orders are not hard coded on FKMud. Do not use.
`ispc`	checks if $whatever is a PC `if ispc($n)`
`ispet`	checks if $whatever is a pet `if ispet($i)`
`ispkill`	checks if $whatever is a pkiller `if ispkill($n)` This feature is not used by FKMud. Do not use.
`isunconcerned`	checks if $whatever is of neutral alignment `if isunconcerned($n)` Is true for NG TN and NE.
`isundead`	checks if $whoever is undead `if isundead($n)`
`iswanted`	checks if $whatever is wanted by the justice system as a criminal `if iswanted($n)`
`killer`	checks if $whatever is a killer `if killer($n)` This feature is no longer used by FKMud. Do not use.
`kismet`	checks the current kismet in the account of $whatever with rhs `if iskismet($n) > 100`
`language`	checks the language that $whatever is speaking with the rhs `if language($n) == Common`
`lck`	checks $whatevers luck with rhs `if lck($n) > 15`
`level`	checks $whatevers level with rhs `if level($n) > 40`
`manaamt`	checks $whatevers mana with rhs `if manaamt($n) > 100`
`manaprcnt`	checks $whatevers mana percentage with rhs

	`if manaprcnt($n) > 50`
`material`	checks if $whatever is made of material on rhs `if material($o) == 22` Use the bit vector number and not the word.
`memorised`	Checks how many times $whatever has memorised a spell `if memorised(magic missile,$n) > 2`
`mobinroom`	checks if mob is in rhs room `if mobinroom(QQ01) > 0`
`mobinvislevel`	checks $whatevers level of mobinvis with rhs `if mobinvislevel($i) > 53`
`month`	checks month number with rhs `if month() == 1` Months Listing
`moveamt`	checks $whatevers move with rhs `if moveamt($n) > 200`
`moveprcnt`	checks $whatevers move percentage with rhs `if moveprcnt($n) > 50`
`name`	checks name of $whatever with rhs `if name($n) == Drizzt`
`norecall`	checks if $whatever is in norecall room `if norecall($n)`

`number`	checks vnum of mob or object or gold on mob depending on $whatever
	`if number($i) > 10`
	if number uses $i then it counts the money on $i in inventory AND in money pouch
	`if number($n) == 7500`
	if number is a char other than $i (ie $n, $f, $c) AND $whatever is an NPC, it compares rhs with the vnum.
	`if number($o) == 7500`
	if number is an object (ie $o) then it compares rhs with the obj vnum

`numinarea`	checks number of players in area with rhs

	`if numinarea(7500) > 1` The start vnum of the area is in parenthesis. The rhs number is the number of PC's in the area.
`objininv`	checks that $# can be seen in the inventory of $whoever `if objininv($n) == $1`
`objinroom`	checks that $# can be seen in the room with $whatever `if objinroom($n) == $1`
`objtype`	checks object type with rhs `if objtype($o) == 1`
`objval0`	checks object value0 with rhs `if objval0($o) == 1`
`objval1`	checks object value1 with rhs `if objval1($n) == 1`
`objval2`	checks object value2 with rhs `if objval2($n) == 1`
`objval3`	checks object value3 with rhs `if objval3($n) == 1`
`objval4`	checks object value4 with rhs `if objval4($n) == 1`
`objval5`	checks object value5 with rhs `if objval5($n) == 1`
`objisworn`	checks that $whatever is worn by $n `if objisworn($o) == $1`
`order`	checks $whatevers order with rhs Orders are not used on FKMud. Do not use. `if order($n) == Ordername`

`otypecarry`	checks if $whatever has rhs number of objects of this type in inventory or worn
	`if otypecarry(1) > 1`
	`if otypecarry(armor) > 1`

otypehere	checks if itemtype is in room or on $whatever `if otypehere(1) > 1`
otypeinv	checks if $whatever has rhs item type in inv `if otypeinv(1) > 1`
otyperoom	checks if room has rhs item type `if otyperoom(1) > 1`
otypewear	checks if $whatever has rhs item type worn `if otypewear(1) > 1`
ovnumcarry	checks if $whatever has rhs vnum carried `if ovnumcarry(7500) == 1` Carry means in inventory or is worn.
ovnumhere	checks if how many vnum are in room `if ovnumhere(30) > 0`
ovnuminv	checks if $whatever has rhs vnum in inv `if ovnuminv(1) > 1`
ovnumroom	checks if room has rhs vnum `if ovnumroom(1) > 1`
ovnumwear	checks if $whatever has rhs vnum worn `if ovnumwear(1) > 1`
ownsmark	checks if $whatever is the marked owner of an object `if ownsmark($n)`
pcinroom	compares rhs to number of PCs in room with $whatever `if pcinroom($n) > 2`
perm_cha	compares rhs to number of PCs permanent charisma `if perm_cha($n) > 12`
perm_con	compares rhs to number of PCs permanent constitution `if perm_con($n) > 12`
perm_dex	compares rhs to number of PCs permanent dexterity

	`if perm_dex($n) > 12`
`perm_int`	compares rhs to number of PCs permanent intelligence `if perm_int($n) > 12`
`perm_lck`	compares rhs to number of PCs permanent luck `if perm_lck($n) > 12`
`perm_str`	compares rhs to number of PCs permanent strength `if perm_str($n) > 12`
`perm_wis`	compares rhs to number of PCs permanent wisdom `if perm_wis($n) > 12`
`position`	checks $whatevers position with rhs `if position($n) == 8`
`practice`	checks $whatevers number of pracs with rhs Practice points are the same as stat points. `if practice($n) > 0`
`quality`	checks the quality of an object with the rhs `if quality($o) < 2` Use the bit vector and not the word.
`quest`	checks if $whatever has questbits in area with rhs `if quest(0,1,$n) == 0`
`questr`	checks if $whatever has questbits in area with vnum r with rhs `if questr(8000,0,1,$n) == 0`
`race`	checks $whatevers race with rhs `if race($n) == Human`
`rand`	checks random percent with that in () `if rand(50)`

`resistance`	checks if $whatever has rhs resistance to resistancetype(0)
	`if resistance(0, $n) > 25`
	`if resistance(cold,$n) > 50`
	`if resistance(fire,$n) <= 25`

`sector`	checks sector of room with rhs `if sector($n) == 13` Use the bit vector number, not the word.
`sex`	checks $whatevers sex with rhs `if sex($n) == 1` for males `if sex($n) == 2` for females `if sex($n) == 0` for neutral
`skilllevel`	checks if $whatever has skilllevel of skill with rhs `if skilllevel(second attack, $n) > 5`
`skillcheck`	checks if $whatever passes the skill level check with the rhs `if skillcheck(second attack, $n)`
`str`	checks $whatevers strength with rhs `if str($n) > 15`
`string`	checks $# for the input word `if string($1) == smell` See the String If Check Function lesson for more information.
`stringprefix`	checks $# for the input word, including shorthand input `if stringprefix($1) == smell` See the String If Check Function lesson for more information.
`temp`	checks the temperature of the room that $whatever is in, in Fahrenheit. `if temp() > 15`
`thief`	checks if $whatever is a thief `if thief($n)` No longer used by FKMud since the implementation of the new Justice system.
`time`	checks hour number with rhs `if time() == 1` Time ranges from 0 - 23
`timeskilled`	checks $whatevers times killed with rhs This is no longer used by FKMud.
`value5bits`	checks the quest bit of value5 on an object

	`if value5bits(0,5,$o) == 1`
`wasinroom`	checks room $whatever was last in with rhs `if wasinroom($n) == 7500`
`wear_loc`	checks where object is currently worn with rhs `if wear_loc($o) != -1` Example checks for if it is not worn.

`weather`	checks if weather is cloudless/cloudy/rainy/lightning
	`if weather() == cloudy`
	`if weather() == cloudless`
	`if weather() == rainy`
	`if weather() == lightning`

`wis`	checks $whatevers wisdom with rhs `if wis($n) > 15`

FOR SOCIALS, SPELLS AND MOB PROGS

$whatevers	**capital of each includes the title if after /**
`$n/N`	char that triggers prog or ch in hardcode $n refers to the PC by name, $N refers to the PC by their adjective
`$i/I`	self mob with the prog
`$t/T`	victim(used in hard code) - Do not use in area code. It crashes the game.
`$r/R`	random char in room
`$e`	he/she of $n
`$m`	him/her of $n
`$s`	his/her of $n
`$E`	he/she of victim
`$M`	him/her of victim

$S	his/her of victim
$j	he/she of $i
$k	him/her of $i
$l	his/her of $i
$J	he/she of $r
$K	him/her of $r
$L	his/her of $r
$o/O	object with the prog
$c/C	character carrying $o
$p/P	object in hard code
$a/A	article for $o
$f/F	character that $n is fighting
$w/W	The group leader (use W)
$b	room name
$v	room vnum
$0 to $9	strings to be checked for

$TARGETS

The following is a list of all the $targets or $whatevers that can be used in progs.

$whatevers	capital of each includes the title if after /
$n/N	char that triggers prog or ch in hardcode $n refers to the PC by name, $N refers to the PC by their adjective
$i/I	self mob with the prog
$t/T	victim(used in hard code) - Used in give away progs for the person the item is being GIVEN TO.
$r/R	random char in room
$e	he/she of $n
$m	him/her of $n
$s	his/her of $n
$E	he/she of victim
$M	him/her of victim
$S	his/her of victim
$j	he/she of $i
$k	him/her of $i
$l	his/her of $i
$J	he/she of $r
$K	him/her of $r
$L	his/her of $r

$o/O	object with the prog
$c/C	character carrying $o
$p/P	object in hard code
$a	article for $o
$A	player account
$f/F	character that $n is fighting
$w/W	The group leader (use W)
$b	room name
$v	room vnum
$0 to $9	strings to be checked for

MP COMMANDS

The following is a list of all the mpcommands that can be used in mobile/object/room programs.

mpadvance	This command will advance a PC in levels. This should never ever be used.
mpaffect	This will cast a spell without echos. There is no chance of failure but there is a chance for resisting the spell. This is ideal for when you want to have mobiles who have special attacks that are fire based etc, allow for resisting. `mpaffect poison $r`
mpapply	This command is never to be used. It makes the PC apply for authorisation. It is only used in the character creation process. `mpapply $n`
mpapplyb	This command is never to be used. It makes the PC apply for authorisation. It is only used in the character creation process. `mpapplyb $n`
mpareaecho	This will do an echo that will echo to the WHOLE area. `mpeareacho You can hear the screams of a banshee nearby.`
mpasound	This will echo a chosen phrase to the rooms immediately adjacent and not in the current room. `mpasound You hear the sound of a bell ringing nearby.`
mpat	This does a command in another location. `mpat 8030 yell Help! I am being attacked in Selunes Temple.` This will mean that despite the fact the mobile is in another area they can yell in 8030 which is Waterdeep. This is ideal for small areas hanging off another area where people are more likely to be to call for help etc.
mpcast	This command will cast a spell without a chance of failure. It can be resisted, but echos are still seen. `mpcast poison $r`
mpclearability	Reduces the current number of uses for an ability `mpclearability 'shapechange' $n`

mpclearspell	Removes one prepared spell `mpclearspell 'magic missile' $n`
mpclosepassage	This closes the exit in a certain room in a certain direction. It only works for EX_PASSAGE exits. `mpclosepassage roomvnum exitdirection`
mpdamage	This will damage a PC or a mobile for a certain number of hps without killing them. We prefer that mpmadd currhp's is used rather than this. It does not take hit locations into account. `mpdamage $n 100` This will damage the PC for 100 hitpoints.
mpdegrade	This changes the condition of equipment of the specified material type. `mpdegrade $n 17 -1` The 17 refers to the material type. Refer to the objects lesson on material types for the bit number of each material type. You must use the bit vector number and not the word in this situation. The -1 degrades the object by 1.
mpdeposit	This deposits some gold into the current area's ecomomy. You would use it in situations like when you use bribe progs for boat trips etc, put that money back into the economy. `mpdeposit 1000`
mpdrain	This does the opposite of mprestore. If an amount is not specified it will take the target down to 1 hp/mana/stamina and mentalstate to -99. It does not take into account hit locations and is better handled with mpmadd and mpmset percenthp. `mpdrain $n`
mpdream	It echos to a sleeping target only. `mpdream $n You dream about ghosts and trolls.`
mpecho	This echoes to the room only. `mpecho The young elf grimaces in pain before doubling over.`
mpechoaround	This echoes to everyone in the room except for the target. It is often used in conjunction with mpechoat. `mpechoaround $n The ghoul glares at $n.`
mpechoat	This echoes only a specified target. `mpechoat $n The ghoul glares at you.`

`mpfavor`	This raises a players favour by the amount given. `mpfavor 100` Favour ranges from -1000 to 1000.
`mpforce`	This forces a mobile or a PC to perform an action. `mpforce $n dance` This forces the PC to dance.

`mpgive`	This command gives an object to a PC or a mobile. It is better than give as it does not worry about if they PC can hold the object or not based on encumberance etc. There is no echo with mpgive so echos if needed will have to be made using mpechoat and mpechoaround.
	`mpgive knife $n`
	`mpgive i12004 $n`

`mpgoto`	Sends the mobile to a given location.
	`mpgoto 8030`
	This will send the mobile to room 8030.
	`mpgoto 3`
	Going to room vnum 3 will purge the mob from the game.
	`mpgoto 99`
	Room 99 is a waiting room for mobiles to sit in while they are not being used. For instance mobiles that go home for the night would go to 99.

`mphireemployee`	This command is used by the dwellings system to allow PC's to hire NPC's to work at their dwellings. Do not use it for normal areas. `mphireemployee $n`
`mpinfect`	This will infect the target with disease. `mpinfect $n`

`mpinvis`	This sets the mobiles invis level or allows them to go wizinvis. This means they will not be viewable by PC's under a certain level.
	`mpinvis`
	This command without argument will send the mobile wizinvis to a default level of 53 meaning that only immortals can only see this mobile now.
	`mpinvis 60`
	This command will make the mobile invisable to all PCs and IMMs below level 60.

`mpjunk`	This will remove an object from the game. `mpjunk $o` This will junk the given object and is used on the object itself to be junked.

	`mpjunk i12004`
	This will junk a given object name (i12004 could also be knife or whatever other names the object has). It is best to use the vnum unless you do not care which vnum of items with the same name you are junking.
	`mpjunk i12004 $n`
	This will junk a given object name (i12004 could also be knife or whatever other names it has) on a specific target (namely the PC).
	`mpjunk all`
	This will junk all items in inventory of the mobile.
	`mpjunk all.iQQ00 $n`
	This will junk all items of that vnum in the inventory of the PC.
	`mpjunk all.iQQ00`
	This will junk all items of that vnum in the inventory of the mobile.

`mpkill`	This force the mobile to attack a given target. You can specify which rank in the group formation will be attacked.
	`mpkill $n`
	`mpkill $r middle`
	If there is a character in the middle rank of the group, it will attack them first.

`mpkilllist`	This modifies a PCs kill list.
	arguments: add(add mob to list), clear(clears list), remove(removes mob from list)
	`mpkilllist add $n`
	`mpkilllist clear $n`
	`mpkilllist remove $n`

`mplag`	This command does not work and impossible to code according to the coders.
`mplog`	This will record something to the log file and also echo to any immortals in the game at the time. See the Logging Events list for a list of events we want logged. `mplog EVENT: $n has entered Elminster's Tower.` This tells the immortals on at the time that someone is in the tower and if at all possible Elminster will be loaded for roleplay.
`mpmadd`	This command will add to a given field on a PC or mobile. `mpmadd victim field amount` See the lesson on mpmadd for more information on things that can be mpmadded.
`mpmakecash`	This will enable the mobile/object to make coins. It cannot make coins onto a PC. `mpmakecash 10 platinum` This will make 10 platinum coins.

	`mpmakecash 5 gold` This will make 5 gold coins.
`mpmload`	This loads up a given mobile into that room. `mpmload 8023` This loads up mobile vnum 8023 into the room.
`mpmset`	This command will set given fields on a PC or a mobile. `mpmset victim field amount` See the lesson on mpmset for more information on what things can be mpmset.
`mpnothing`	This does nothing and is used for scripts. Do not use without very good reason. `mpnothing`
`mpoadd`	This command will add to a given field on an object. `mpoadd object field amount` See the lesson on mposet for more information on things that can be mpoadded.
`mpoload`	This will load up a specified object. The level of the object can be specifed providing the level required is not higher than the level of the mobile doing the loading. If the level is not required to be specified then the field would be left off. `mpoload 12004 5` This will load up the specified object to level 5.
`mpopenpassage`	This opens an EX_PASSAGE exit. `mpopenpassage roomvnum exitdirection`
`mposet`	This command will set a given field on an object. `mposet object field amount` See the lesson on mposet for more information on things that can be mposet.
`mppeace`	This will stop a fight. It should stop agressive mobiles as well because it forces them to sit. `mppeace`
`mppkset`	This command is never to be used. It makes the PC apply to be able to pkill. It is only used in the character creation process. `mppkset $n`
`mppractice`	This command practices a given skill. This is used to give skill levels in trades etc. Please get permission from the area admins before using this comand. `mppractice $n skill-level skill` Skill level is between 1 and 25.

mppunish	This command deducts experience from the PC.
	`mppunish $n`
	`mppunish $n some`
	`mppunish $n lots`

mppurge	This will purge a given item, mobile or room. The difference between mppurge and mpjunk is that mppurge looks in the room and gets rid of mobiles as well as object. Mpjunk is mainly used for getting rid of objects on a mobile or PC.
	`mppurge`
	Without argument like this it will purge everything in the room. It does not purge anything in the players inventory. It does not purge itself either.
	`mppurge m8023`
	This will purge the mobile of the vnum 8023 which is a Waterdeep Soldier. The word soldier could have been used in this case or any other key words that the mobile has. The use of m8023 means there is little chance for confusion.
	`mppurge i12004 $n`
	This will purge the object i12004 on the target $n. It is best to use the iVNUM in this case. If other key words like knife is used then it may cause the wrong object to purge.

mpquiet	This when turned on makes a mobile ACT_SECRETIVE so that any actions performed by this mobile are not echoed. When turned off the flag is removed.
	`mpquiet on`
	`mpquiet off`

mpregoto	This will send the mobile back to the vnum from where they did a mpgoto from. For instance mobiles in room 99 would often use mpregoto to go back to the room they were in before they went home for the night. `mpregoto`
mpresize	This will resize given object to the size of the being on the RHS. For instance, if you have a quest item made especially for the PC, you can have it made to their size. `mpresize iVNUM $n`

mprestore	Full restore of mana/hps/move and mentalstate. If you specify it will do just the one. This command is to be used sparingly. This can also be handled by mpmadd and mpmset percenthp.
	`mprestore $n hit`
	Restores all of the targets hit points.
	`mprestore $n mana`
	Restores all of the targets mana points.

	`mprestore $n move`
	Restores all of the targets stamina points.
	`mprestore $n mentalstate`
	Sets the targets mentalstate back to 0.

`mpreward`	This command gives experience from the PC.
	`mpreward $n`
	`mpreward $n some`
	`mpreward $n lots`

`mprset`	This command edits room fields.
	flag(sets bit), flags(toggles bit), sector
	`mprset sector 13`
	`mprset flags 2`

`mpsetclan`	This command will join the PC into a guild or a clan. `mpsetclan $n Illusionists`
`mpsetfeat`	This command will give the PC the specified feat. `mpsetfeat $n 'blind-fight' 1` The 1 is how high the level of the feat is, if the feat has more than one level. If only one level of feat then just use the 1.
`mpsetheight`	This will set the height of the PC. It is only used in character creation. `mpsetheight $n tall` This takes a range within the tall range of that height for the PC's race. Short can also be set. If the command does not specify a height then it sets the height within the average range. This is only used in the character generation process.
`mpsetsong`	This command will allow a PC to know the specified bardsong. `mpsetsong $n 'song of heroism' 4` The number is the number of lines that the PC is to know of the song.
`mpsettrap`	This command sets up a trap on a room exit or on an object. `mpsettrap <obj|exit> <type> <reloads> <trigger>`
`mpsetweight`	This will set the weight of the PC. This is only used in the character creation process. `mpsetweight $n thin` This works as for mpsetheight. You can also set fat.
`mpslay`	This makes the object or mobile slay a given target whether it be a mobile or a PC.

	`mpslay $n`

`mptakecash`	This will take a specified amount of gold from the specified target. The value is in copper. It doesn't matter if the PC has other currency in inventory, the code will take the equivalent in copper and give back change. The PC must have the money in inventory, not in a container for it to work.
	`mptakecash $n 100`
	This will take the equivalent of 100 copper from the PC.
	`mptakecash $n 100 [haggle]`
	If you add haggle after the command it will haggle and echo what the cost was haggled down to.

`mptakecashroom`	This will take a specified amount of gold from the room. The value is in copper. It doesn't matter if the room has other currency in the floor, the code will take the equivalent in copper and give back change. The coins must be outside of containers etc for this to work.
	`mptakecashroom 100`
	This will take the equivalent of 100 copper from the room.
	`mptakecashroom 100 [container]`
	If you add container to the command it will check containers for the cash.

`mptrain`	This command will train a statistic without costing gold. It still requires a stat point. This command should be used rarely if at all. `mptrain $n wisdom` This will raise the PC 1 wisdom.

`mptransfer`	This will transfer the target to a given location.
	`mptransfer $n 8030`
	This will transfer the PC to that room vnum. It will also transfer the PC's pet.
	`mptransfer all 8030`
	This will transfer all in the room to that room vnum.
	`mptransfer $n 8030 pet`
	This will transfer the PC and pet to that room vnum.

`mpunintercept`	This command is used to break an intercept prog to make the command that it is intercepting acting as normal from that point on. `mpunintercept`
`mpwalkto`	This command will make a mobile walk to a given vnum following the best path that they can find. Mobiles cannot go through locked doors when using mpwalkto. `mpwalkto 8030` Specify the vnum of the room to walk to.

`mpwithdraw`	This withdraws money from the economy of the area. The value is in copper. If you do mpmakecash to give coins to a player as part of a quest you should take that amount from the economy. `mpwithdraw 1000`

MPMSET AND MPMADD FIELDS

For builders with high level god testing characters on the testport, many of these commands can be used with mset for testing and for OLC. Some fields cannot be mpmadd or cannot be mpmset. If a sample is not given for mpmadd or mpmset, then that particular field cannot be mpmset or mpmadd 'd.

affected	Sets what a mobile or PC is affected by. Refer to Mobiles Affect Flags Lesson for a list of affected flags. `mpmset $n affected blindness`

align	Sets the permanent alignment of a PC and the alignment of a mobile. Do not set a PC's alignment in quests but rather set their lawful and their good. See Mobile Alignments Lesson for more information on alignments.
	`mpmset $n align 1000`
	`mpmadd $n align 100`

armor	Sets the virtual armour type. See Object Armour Types Lesson for a list of armour types. `mpmset $n armor 10` Use the bit vector, not the word.

blood	Blood is required for vampires.
	`mpmset $n blood 100`
	`mpmadd $n blood 10`

cartowner	Sets the owner for a cart
	`mpmset $n cartowner cartobj`
	`mpmset $n cartowner none`

cha	Sets the charisma of target
	Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
	`mpmset $n cha 19`
	`mpmadd $n cha 1`

class	Sets the class of the PC or mobile. PC's can only be 1 of four classes; rogue, wizard, warrior and priest. There should be no reason for changing a PC's class. See Mobiles Class Lesson for a list of possible mobile classes. `mpmset $n class rogue`

con	Sets the constitution of target
	Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
	`mpmset $n con 19`
	`mpmadd $n con 1`

currhp	Sets the current hitpoints of a PC or mobile. It does not affect the maximum hitpoints.
	`mpmset $n currhp 100`
	`mpmadd $n currhp 10`
	`mpmadd $n currhp -10`

currmana	Sets the current mana of a PC or mobile. It does not affect the maximum mana.
	`mpmset $n currmana 100`
	`mpmadd $n currmana 10`
	`mpmadd $n currmana -10`

currmove	Sets the current move/stamina of a PC or a mobile. It does not affect the maximum move.
	`mpmset $n currmove 100`
	`mpmadd $n currmove 10`
	`mpmadd $n currnove -10`

damroll	Sets the damroll on a mobile. You should not use this field without permission.
	`mpmset $i damroll 10`
	`mpmadd $i damroll 1`

defpos	This sets the position that mobiles revert to after a fight. This field is no longer used by FKMud. `mpmset $i defpos 1`
deity	Sets the deity of a PC or a mobile. Never set the deity of a player without asking permission from the Area Admins. See Mobile Deities Lesson for a listing of deities. `mpmset $n deity mystra`

dex	Sets the dexterity of target
	Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
	`mpmset $n dex 19`
	`mpmadd $n dex 1`

drunk	Sets the drunk level of a PC. Ranges from 0 to 50.
	`mpmset $n drunk 40`
	`mpmadd $n drunk 5`
	`mpmadd $n drunk -1`

`emotion`	Sets the emotion state of the PC. This field is not utilised in the game.
	`mpmset $n emotion 20`
	`mpmadd $n emotion 10`

`favor`	Sets the favor of a PC. Favor goes from -1000 to 1000. In most cases it would be best to use mpmadd.
	`mpmset $n favor 500`
	`mpmadd $n favor 50`

`feats`	Sets the number of feat points for a PC.
	`mpmset $n feats 19`
	`mpmadd $n feats 1`

`flags`	Sets the act flags of a mobile. Refer to Mobiles Flags Lesson for a list of flags. You cannot set flags on a PC. `mpmset $n flags noassist`

`full`	Sets the hunger level of a PC. Ranges from 50 (full) to -50 (starving).
	`mpmset $n full 40`
	`mpmadd $n full 10`
	`mpmadd $n full -10`

`good`	Sets the PC's hidden good alignment field. It ranges from 1000 to -1000. In general it is best to use mpmadd for this field.
	`mpmset $n good 500`
	`mpmadd $n good 50`
	`mpmadd $n good -50`

`height`	Sets the height of a PC to a specific height in inches.
	`mpmset $n height 72`
	`mpmadd $n height 1`

`hitroll`	Sets the hitroll on a mobile. You should not use this field without permission.
	`mpmset $n hitroll 10`
	`mpmadd $n hitroll 1`

`hometown`	Sets the hometown of a player. This is only done as part of character creation. `mpmset $n hometown waterdeep`
	Sets the maximum hitpoints of a PC or a mobile. There is no reason to set a PC's maximum hitpoints.

hp	`mpmset $n hp 300` `mpmadd $n hp 1`
immune	Sets the mobile or PC's immunities. This field is no longer used by FKMud.

int	Sets the intelligence of target
	Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
	`mpmset $n int 19`
	`mpmadd $n int 1`

kismet	Sets the current kismet of the PC's account. Most areas would not set kismet. It is mostly used in character creation.
	`mpmset $n kismet 100`
	`mpmadd $n kismet 10`

lawful	Sets the PC's hidden lawful alignment. It ranges from -1000 to 1000. In most instances it is best to use mpmadd for this field.
	`mpmset $n lawful 500`
	`mpmadd $n lawful 10`
	`mpmadd $n lawful -10`

lck	Sets the luck of target Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
leader	This sets the leader on a PC or mobile, who they are following. `mpmset $i leader $n`

level	Sets the level of a mobile. Level can range from 1 to 50. Do not use this on PCs.
	`mpmset $i level 50`
	`mpmadd $i level 1`

long	Sets the long description of a mobile. It should never be used on a PC. `mpmset $i long {70}A large soldier stands on guard here.`
lycanthrope	Sets the race number for cursed weres or 1000 plus the race number for pure or true lycanthropes. See the Mobile Races Lesson for race numbers or check showrace on the test port. `mpmset $n lycanthrope racenum`

mana	Sets the maximum mana or a PC or a mobile. There is no reason to set a PC's maximum mana.
	`mpmset $i mana 100`
	`mpmadd $i mana 10`

material	This sets the material of the virtual armour that the mobile is wearing. See Object Materials Lesson for a listing of material types. `mpmset $i material 10` Use the bit vector and not the word.
master	Sets the master of a mobile. The mobile becomes the pet or mount for the PC. `mpmset $i master $n` - Sets the mobile into the pet slot. `mpmset $i master $n horse` - Sets the mobile into the mount slot.

mentalstate	Sets the mentalstate of the PC. It ranges from 100 to -100.
	`mpmset $n mentalstate -100`
	`mpmadd $n mentalstate -20`

move	Sets the maximum move/stamina of a PC or a mobile. There is no reason to set a PC's maximum move.
	`mpmset $i move 500`
	`mpmadd $i move 50`

name	Sets the keywords of a mobile. `mpmset $i name mQQ01 large tall soldier` It is best to make sure you use the mVNUM in there so that the game can find it easily.
percenthp	Sets the percentage of the current hitpoints of a PC or mobile. It does not affect the maximum hitpoints. `mpmset $i percenthp 50`
percentmana	Sets the percentage of the current mana of a PC or mobile. It does not affect the maximum mana. `mpmset $i percentmana 50`
percentmove	Sets the percentage of the current move/stamina of a PC or mobile. It does not affect the maximum move. `mpmset $i percentmove 50`
pos	This sets the position of the mobile. Do not use on PC's, use force. In fact you can use force for mobiles as well. See Mobile Positions Lesson for a list of mobile positions. `mpmset $i pos 1`
practice	Sets the amount of stat points that a PC has. Do not use this without permission from the Area Admins. `mpmset $n practice 10`

	`mpmadd $n practice 1`

`qp`	Sets the quest/glory points of a PC. Use mpmadd rather than mpmset for giving glory to a PC. Using mpmset could strip away hard earned glory points from a PC.
	`mpmset $n qp 10`
	`mpmadd $n qp 1`

`quest`	See quest bits lesson for more information.
	`mpmset $n quest 0 4 10`
	`mpmadd $n quest 0 4 1`

`quest_number`	This field is the same as quest, and it is no longer used by FKMud.

`questr`	As for quest bits but sets the bits in another area. Use this when a mobile or object is likely to set bits on a PC outside of the area that the quest is in.
	`mpmset $n questr 8000 0 4 10`
	`mpmadd $n questr 8000 0 4 1`

`race`	Sets the race of the PC or mobile. If you are ever to change the race or class of a PC you should consult with your Area Admin first. For a list of races refer to the Mobiles Race Lesson. `mpmset $i race elf`
`randquest`	This randomises quest bits. Do not use.
`ris`	This field is no longer used by FKMud.

`resist`	Sets the mobile or PC's resistance to the specified resistance. See Mobile Resistances Lesson for a list of resistances. This field sets the degree of resistance, ranging from 0% to 150%.
	`mpmset $n resist typenum amount`
	`mpmadd $n resist typenum amount`

`resistant`	This is no longer used by FKMud.

`sex`	Sets the sex of the PC or mobile. Value 0 for neuter, value 1 for male and value 2 for female.
	`mpmset $n sex 1`
	`mpmadd $n sex 1`
	Use the bit vector number and not the word.

`short`	Sets the short description of a mobile. It should never be used on a PC. `mpmset $i short {70}a short fat soldier`

`size`	Sets the size of a mobile. This is determined by the game by the race of the mobile. This field is not used on FKMud.
`speaks`	This sets the PC or mobile to a language that they know and understand. Do not use this on PC's without permission. `mpmset $n speaks common`
`speaking`	This sets the PC or mobile to a language that they are currently speaking. `mpmset $n speaking common`
`spec`	Sets the special of a mobile. See the Specials Lesson for a list of special functions. `mpmset $i spec spec_guard`

`str`	Sets the strength of target
	Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
	`mpmset $n str 20`
	`mpmadd $n str 1`

`susceptible`	Sets the mobile or PC's susceptabilities. This field is no longer used by FKMud.

`thirst`	Sets the thirst of PC. Ranges from 50 (full) to -50 (thirsty).
	`mpmset $n thirst 20`
	`mpmadd $n thirst 10`
	`mpmadd $n thirst -10`

`title`	Sets the title of the PC. (What shows on the who list). `mpmset $n title of Waterdeep`
`weight`	Sets the weight of a PC to a specific weight in pounds. `mpmset $n weight 150`

`wis`	Sets the wisdom of target
	Value must be between 3 and 18 for PC's and 3 to 25 for Mobiles.
	`mpmset $n wis 20`
	`mpmadd $n wis 1`

MPOSET AND MPOADD FIELDS

For builders with high level god testing characters on the testport, many of these commands can be used with mset for testing and for OLC. Some fields cannot be mpoadd or cannot be mposet. If a sample is not given for mpoadd or mposet, then that particular field cannot be mposet or mpoadd 'd. Note that when the object is being carried by the PC, you need to use the "on $n" syntax.

`actiondesc`	This is the fourth line on an object after the long description. This field is not used by FKMud.

`affect`	Sets affects on an object.
	`mposet on $n iQQ00 affect luck 3`
	`mposet $o affect luck 3`

`capacity`	Sets the capacity field (value0) on containers, carts, sheathes and quivers.
	`mposet on $n iQQ01 capacity 30`
	`mposet $o capacity 30`

`cflags`	This is the container flags, for containers, carts, sheaths and quivers.
	`mposet on $n iQQ00 cflags 1`
	`mposet $o cflags 1`

`charges`	Sets the charges (value2) on Wands and Staves.
	`mposet $o charges 10`
	`mposet on $n iQQ01 charges 10`
	`mpoadd $o charges 5`
	`mpoadd on $n iQQ01 charges 5`

`condition`	Sets the condition of the object.
	`mposet $o condition 1`
	`mposet on $n $o condition 1`
	`mpoadd $o condition 1`
	`mpoadd on $n $o condition 1`

`cost`	Sets the cost value of the object.
	`mposet $o cost 2000`
	`mposet on $n $o cost 2000`
	`mpoadd $o cost 100`
	`mpoadd on $n $o cost 100`

	Cost value is in copper.
delay	This field is no longer used by FKMud.
doses	What is this?

ed	Sets the extra description on an object. It does this by adding extra lines to the description.
	`mposet iQQ11 ed addline 'pink gem hilted longsword' {D0}There is a pink orb set into the hilt of the sword.`
	`mposet on $n iQQ11 ed addline 'pink gem hilted longsword' {D0}There is a pink orb set into the hilt of the sword.`

flags	Sets the flags on an object.
	`mposet on $n iQQ10 flags magic`
	`mposet $o flags magic`

identify	Sets an identify field on an object for objects who's magic lies within object programs rather than applies.
	`mposet $o identify This object does something big!`
	`mposet on $n iQQ01 identify This object does something big!`

key	Sets the key vnum needed to open containers, carts, sheaths and quivers.
	`mposet $o key QQ01`
	`mposet on $n iQQ03 key QQ01`

level	Sets the level of an object.
	`mposet $o level 1`
	`mposet on $n iQQ01 level 1`

long	Sets the long description of an object.
	`mposet $o long {A0}A green wand is lying here.`
	`mposet on $n iQQ01 long {90}A red wand is lying here.`

mark	Sets the ownership mark on an object.
	`mposet $o mark $n`
	`mposet on $n iQQ01 mark $n`

material	Sets the material type of an object.
	`mposet $o material 11`
	`mposet on $n iQQ01 material 11`
	Use the bit vector number rather than the word.

maxcharges	Sets the maximum charges on a wand or staff. Does it do more types than this?
maxdoses	What is this?

name	Sets the keywords on an object.
	`mposet $o name iQQ01 long black robe`
	`mposet on $n iQQ01 name iQQ01 long black robe`
	It is important to but the iVNUM back into the keywords when giving an object new or different keywords.

quality	Sets the quality of an object.
	`mposet $o quality 3`
	`mposet on $n iQQ01 quality 3`
	`mpoadd $o quality 1`
	`mpoadd on $n iQQ01 quality 1`
	Use the bit vector number rather than the word.

quest	This field is no longer used by FKMud.
randquest	This field is no longer used by FKMud.

rmaffect	This field removes an affect from the object in order. For instance if you want the second affect from an object removed, then you will specify 2.
	`mposet $o rmaffect 2`
	`mposet on $n iQQ01 rmaffect 2`

short	Sets the short description of an object.
	`mposet $o short {80}a black robe`
	`mposet on $n iQQ01 short {80}a black robe`

size	Sets the size of an object.
	`mposet $o size 4`
	`mposet on $n iQQ01 size 4`
	`mpoadd $o size 1`
	`mpoadd on $n iQQ01 size 1`
	Use the bit vector number rather than the word.

slevel	This is the spell level of wands, scrolls, potions, pills and staves.
	`mposet $o slevel 30`
	`mposet on $n iQQ01 slevel 30`

spell	Sets the first spell on an object, value1. Syntax needed
spell1	Sets the second spell on an object, value2 Syntax needed
spell2	Sets the third spell on an object. Syntax needed
spell3	Sets the fourth spell on an object. Syntax needed
tflags	This sets the Trigger Flags (value0) on buttons and levers etc. Syntax needed

timer	Sets the timer on an object.
	`mposet $o timer 100`
	`mposet on $n iQQ01 timer 100`
	`mpoadd $o timer 10`
	`mpoadd on $n iQQ01 timer 10`

type	Sets the item type of an object.
	`mposet $o type 1`
	`mposet on $n iQQ01 type 1`
	Use the bit vector number rather than the word.

value0	Sets the value of value0.
	`mposet $o value0 5`
	`mposet on $n iQQ01 value0 5`
	`mpoadd $o value0 1`
	`mpoadd on $n iQQ01 value0 1`
	For many object types this value sets a specific thing on the object. Change with care.

value1	Sets the value of value1.
	`mposet $o value1 5`
	`mposet on $n iQQ01 value1 5`
	`mpoadd $o value1 1`
	`mpoadd on $n iQQ01 value1 1`
	For many object types this value sets a specific thing on the object. Change with care.

value2	Sets the value of value2.
	`mposet $o value2 5`
	`mposet on $n iQQ01 value2 5`
	`mpoadd $o value2 1`
	`mpoadd on $n iQQ01 value2 1`

	For many object types this value sets a specific thing on the object. Change with care.

value3	Sets the value of value3.
	`mposet $o value3 5`
	`mposet on $n iQQ01 value3 5`
	`mpoadd $o value3 1`
	`mpoadd on $n iQQ01 value3 1`
	For many object types this value sets a specific thing on the object. Change with care.

value4	Sets the value of value4.
	`mposet $o value4 5`
	`mposet on $n iQQ01 value4 5`
	`mpoadd $o value4 1`
	`mpoadd on $n iQQ01 value4 1`
	For many object types this value sets a specific thing on the object. Change with care.

value5	Sets the value of value5. Value5 is not a specified field in any object type. This makes it ideal for using in object programs to check values and usage of an item.
	`mposet $o value5 5`
	`mposet on $n iQQ01 value5 5`
	`mpoadd $o value5 1`
	`mpoadd on $n iQQ01 value5 1`

value0bits	Sets the quest bits on an object in value0.
	`mposet $o value0bits 0 5 1`
	`mposet on $n iQQ01 value0bits 0 5 1`
	`mpoadd $o value0bits 0 5 1`
	`mpoadd on $n iQQ01 value0bits 0 5 1`
	Note that this value is often used by objects to set specific fields. Use this with caution. It is recommended that value5bits be used in most instances.

value1bits	Sets the quest bits on an object in value1.
	`mposet $o value1bits 0 5 1`
	`mposet on $n iQQ01 value1bits 0 5 1`
	`mpoadd $o value1bits 0 5 1`
	`mpoadd on $n iQQ01 value1bits 0 5 1`
	Note that this value is often used by objects to set specific fields. Use this with caution. It is recommended that value5bits be used in most instances.

	Sets the quest bits on an object in value2.

value2bits	mposet $o value2bits 0 5 1 mposet on $n iQQ01 value2bits 0 5 1 mpoadd $o value2bits 0 5 1 mpoadd on $n iQQ01 value2bits 0 5 1 Note that this value is often used by objects to set specific fields. Use this with caution. It is recommended that value5bits be used in most instances.
value3bits	Sets the quest bits on an object in value3. mposet $o value3bits 0 5 1 mposet on $n iQQ01 value3bits 0 5 1 mpoadd $o value3bits 0 5 1 mpoadd on $n iQQ01 value3bits 0 5 1 Note that this value is often used by objects to set specific fields. Use this with caution. It is recommended that value5bits be used in most instances.
value4bits	Sets the quest bits on an object in value4. mposet $o value4bits 0 5 1 mposet on $n iQQ01 value4bits 0 5 1 mpoadd $o value4bits 0 5 1 mpoadd on $n iQQ01 value4bits 0 5 1 Note that this value is often used by objects to set specific fields. Use this with caution. It is recommended that value5bits be used in most instances.
value5bits	Sets the quest bits on an object in value5. Value5 is not used by any objects to set any specific fields, so this value is safe to use for quest bit checking. mposet $o value5bits 0 5 1 mposet on $n iQQ01 value5bits 0 5 1 mpoadd $o value5bits 0 5 1 mpoadd on $n iQQ01 value5bits 0 5 1
weapontype	Sets the weapon type of an object. mposet $o weapontype 1 mposet on $n iQQ01 weapontype 1 Use the bit vector number rather than the word. Syntax check needed.
wear	Sets the wear location of an object. mposet $o wear neck mposet on $n iQQ01 wear neck
	Sets the weight of an object.

weight	mposet $o weight 10 mposet on $n iQQ01 weight 10 mpoadd $o weight 1 mpoadd on $n iQQ01 weight 1

QUEST BITS LISTING

This file contains many of the final quest bits for many of the quests in the game. It might be that you want to check that a PC has done a certain quest before starting yours. This file is not yet complete but is being added to constantly. If there is a set of quest bits you are needing that is not on the file contact the builders admins.

Quest Name	Area Name	File Name	Area Vnum	Final Quest Bits	Knowledge	Glory	Exp	Other rewards	Comments
Waterdeep Murder	Waterdeep	limbo.are	100	0 3 1					Obsolete due to justice system
Berdusk Murder	Berdusk	limbo.are	100	3 3 1					Obsolete due to justice system
Berdusk Attempted Murder	Berdusk	limbo.are	100	3 3 2					Obsolete due to justice system
Smoon Murder	Silverymoon	limbo.are	100	6 3 1					Obsolete due to justice system
Smoon Attempted Murder	Silverymoon	limbo.are	100	6 3 2					Obsolete due to justice system
MH Murder	Mithril Hall	limbo.are	100	9 3 1					Obsolete due to justice system
MH Attempted Murder	Mithril Hall	limbo.are	100	9 3 2					Obsolete due to justice system
									Obsolete

Daggerford Murder	Daggerford	limbo.are	100	12 3 1					due to justice system
School of Combat	Fighters Guild		600	0 1 1					
Orcs fighters guild	Fighters Guild		600	1 1 1					
Dwarves Fighters Guild	Fighters Guild		600	2 1 1					
Gnome Fighters Guild	Fighters Guild	fightersguild.are	600	3 1 1					
Fetch items for fighter in Wdeep	Fighters Guild	fightersguild.are	600	4 3 7		1	some	a guild weapon	
School of Won attack	School of Wonder		700	0 2 1					
School of Won death	School of Wonder		700	0 2 2					
School of Won Grad	School of Wonder		700	2 7 102					
Mages Guild S of Won	School of Wonder		700	9 1 1					
Enchanters Guild SofW	School of Wonder		700	10 1 1					
Wizards Apraisal Quest	School of Wonder		700	11 4 8		2			
Rangers Guild Join	Rangers Guild		900	0 7 75					
Axe prog rangers guild	Rangers Guild		900	7 2 1					
Elven chain greed prog	Rangers Guild		900	11 1 1					
Elven blades greed prog	Rangers Guild		900	12 1 1					
Fletching Quest	Rangers Guild		900	13 5 9					
ZK necroguild join	Necro		1000	0 1 1					
Faerdale necro guild jo	Necro		1000	1 1 1					

Feebov necro guild join	Necro		1000	2 1 1					
Draconic golem created	Necro		1000	10 2 2					
Moradin Pray Waterdeep	Pantheon		1200	0 1 1					
Kill wyvern	Wyverns Tower		1600	0 4 7					
Pathfinding Quest	Wyverns Tower		1600	4 4 8					
Kill any centaurs etc	Wyverns Tower		1600	4 4 9					
Alaron evil dragon quest	Alaron		1700	0 3 7					
Alaron squire quest	Alaron		1700	3 5 20		3	20000	2 armour 1 quilted tunic	
Fetch crabs	Alaron		1700	9 2 2					
Armour appraisal quest	Alaron		1700	11 4 10					
Murder in Alaron	Alaron		1700	15 1 1					Used for killing dwarves in Alaron
Simon Stonebreaker	Shadowdale		1800	0 2 3					
Rescue Simon dwarf	Shadowdale		1800	0 2 3					
Get wand from drow	Shadowdale		1800	2 1 1					
Kill Darkwalker	Gwynneth		2000	0 4 6					
Get Keans help	Gwynneth		2000	14 1 1					
Get Alicias help	Gwynneth		2000	15 1 1					
Shining Mirror Quest	Gwynneth		2000	4 3 3					
Spy on Ascarle	Gwynneth		2000	9 2 2					
Elven artifact Tristan	Gwynneth		2000	11 2 2					

Fugative Quest	Gwynneth		2000	7 4 10					
Brackish Herbs	Gwynneth		2000	13 1 1					
Corwell Attempted Murder	Corwell		2000	16 2 1					
Corwell Murder	Corwell		2000	16 2 2					
Coin prog room 2229	Storm Keep		2200	15 1 1					
Dig prog	Storm Keep		2200	16 1 1					
Find shield quest	Storm Keep		2200	17 5 31					
Pig pendant prog	Storm Keep		2200	22 2 2					
High Priest quest	Maos		2300	0 3 4					
Aurazin/Lacedons	Maos		2300	23 4 9					
Rescue Mistress Thann	Howling Peak		2600	0 5 9					
Skin Mistress Thann	Howling Peak		2600	0 5 5					
Kill Mistress Thann	Howling Peak		2600	0 5 10					
Snow Queen	Frozen Wastes		2700	0 4 10					
Wand Dropping	Frozen Wastes		2700	4 1 1					
Camp activity in orc camp	Orc Village	orcvill.are	2800	0 2 1					
Elf ear quest	Orc Village	orcvill.are	2800	2 3					
Elf conquerer quest?	Orc Village	orcvill.are	2800	5 3					
Kill an ogre	Ogres Den		2850	18 2 1					
Kill halfling quest	Ogres Den		2850	16 2 2					
Dornal's quest	Dornal's area		2900	7 3 6					
Mage Fair Attemp Murder	Mage Fair		2900	20 1 1					
Elandra Greed prog	Mage Fair		2900	21 1 1					
Micklebottom Greed prog	Mage Fair		2900	22 2 3					
Willandra Greed			2900	24 2 2					

Prog	Mage Fair								
Antrea Greed Prog	Mage Fair		2900	26 1 1					
Wilsconin Greed Prog	Mage Fair		2900	27 2 2					
Stanley Greed Prog	Mage Fair		2900	29 2 2					
Callie Greed Prog	Mage Fair		2900	16 4 15					
Menzo First House	Menzo		3100	0 3 1					
Menzo Second House	Menzo		3100	0 3 2					
Menzo Third House	Menzo		3100	0 3 3					
Menzo Fourth House	Menzo		3100	0 3 4					
Priestess of Lloth	Tier Breche		3600	0 5 31					
Fail whip quest	Tier Breche		3600	5 5 9					
Fail Serpent quest	Tier Breche		3600	5 5 26					
Whip quest	Tier Breche		3600	5 5 28					
Find Petronins tiger	Ardeep	ardeep.are	3800	0 4 14		1	100	5 gold or necklace of willow bark	
Find the traps quest	Ardeep	ardeep.are	3800	4 4 12		2	500	holy leaf bracelet	
Killed Tamina	Ardeep	ardeep.are	3800	4 4 15					
Petronins tiger - dead or alive	Ardeep	ardeep.are	3800	8 3					
Borrow Petronins shovel	Ardeep	ardeep.are	3800	16 3 2/3					
Berdusk Dungeon Innoc	Berdusk		4000	0 4 0					
Berdusk Dungeon	Berdusk		4000	0 4 14					
Helms Watch Song quest	Berdusk	bersusk.are	4000	4 4 13		2	some	Song of Helms Watch Skill	

Leather Armour Making Quest	Berdusk	bersusk.are	4000	8 5 10		1		3 points leatherarmour making	
Casting quest	Silverymoon	silverymoon.are	4500	0 1 1					Allowed to cast in Silverymoon
Griffon Riding Quest	Silverymoon	silverymoon.are	4500	1 1 1					Paid for the Griffon
Victor the wizard metal quest	Silverymoon	silverymoon.are	4500	2 5 17		2	little	Choice of staff or kophesh	
Vampire quest	Summerfield	summerfield.are	4600	0 4 7		2			
Items in chest in pub	Summerfield	summerfield.are	4600	5 1 1					
Find Lorelie's Books(0-7used)	House Moon		4700	6 2 3					
Make dragon armour	Temple of Malar		4900	0 4 9					
Make dragon skull	Temple of Malar		4900	4 1 1					
Song of Heroism	Daggerford	daggerford.are	5200	0 4 13		2	some	Song of Heroism Song Skill	
Tanning Quest	Daggerford	daggerford.are	5200	4 3 6		1		3 points in tanning skill	
Steal the gold box	Daggerford	daggerford.are	5200	7 4 4		1	some	random thieves reward	
Save Queen Brianna	Hartsvale	hartsvale.are	5700	0 4 8					
Betray Queen for Titan	Hartsvale		5700	5 4 7					
Kill Tavis for Titan	Hartsvale		5700	10 1 1					
Kill Avner for Titan	Hartsvale		5700	11 1 1					
Avners Appraisal Quest	Hartsvale		5700	12 4					

Help the queen	Hartsvale		5700	0 4 8					
Betray the queen	Hartsvale		5700	5 4 7					
Got ugly weapon	Hartsvale		5700	0 4 10					
Killed in Hartsvale (G)	Hartsvale		5700	0 4 11					
Killed Tavis	Hartsvale		5700	10 1 1					
Killed Avner	Hartsvale		5700	11 1 1					
Gotten ice diamond	Hartsvale		5700	12 1 1					
Rescue Gurden's son	Tantras sewers		5900	0 4 6					
Aid Spider King	Tantras sewers		5900	0 4 9					
Ascarle drow quest	Ascarle		4000						
Cleaned mushroom caves	Silv Sewers		6100	0 2 2					
Used (mushroom quest)	Silv Sewers		6100	2 5 *					
Melchin's quest	Silv Sewers		6100	8 3 7					
Paid for boat trip to Ascarle	Ascarle	ascarle.are	6200	0 1 1					
Killed Neried in Ascarle	Ascarle	ascarle.are	6200	1 2 2					
KIlled Giant Reef in Ascarle	Ascarle	ascarle.are	6200	3 2 2					
Killed Giant Octopus Ascarle	Ascarle	ascarle.are	6200	5 1 1					
Killed Melenti in Ascarle	Ascarle	ascarle.are	6200	6 1 1					
Killed Aboleth in Ascarle	Ascarle	ascarle.are	6200	7 1 1					
Completed Ascarle Quest	Ascarle	ascarle.are	6200	1 8 127		2	2000	A hardwood staff +2	
Learn to mine quest	Mines One		6600	0 4 11					
Recover and fix crown	Mines Two		6600	8 4 12					

Shore up mines quest	Mines Three		6600	4 4 7					
Help dwarven ghost	Mines Four		6600	12 4 12					
Learn to smelt quest	Undercity		6700	0 4 7					
Learn to make weapons	Undercity		6700	4 4 5					
Learn to make armour	Undercity		6700	8 4					
Weyr Rat Head	Waterdeep Sewer	wdsewers.are	7000	0 1 1					
Magician Errand Quest	Zhentil Keep	zhent.are	7500	0 3 7		1	some	potion of prot, pink gem, 5 plat	
Kill the Red Dragon	Zhentil Keep	zhent.are	7500	8 2 3				Gems	
Black Dagger Quest	Zhentil Keep	zhent.are	7500	15 4 9					
Death Mark Quest	Zhentil Keep	zhent.are	7500	19 3		1	2000	A diamond	
Sword Quest	Zhentil Keep	zhent.are	7500	24 4 15		1			
Allowed to use forge in keep	Zhentil Keep	zhent.are	7500	28 1					
Imp Mirror Usage	Zhentil Keep	zhent.are	7500	29 2 3				Imp Mirror breaks	
Kill the lich in ZK	ZK sewers		7600	0 2 3					
Free prisoners in keep	ZK sewers		7600	2 2 3					
Shove prisoners in keep	ZK sewers		7600	2 2 2					
Cyric quest	ZK sewers		7600	2 2 3					
Load wand in lich hoard	ZK sewers		7600	4 1 1					
Join the Zhentarim	Zhentil HQ		7700	0 5					

Get Zhent Pay	Zhentil HQ		7700	6 1					
Save Mrs Thann Quest	Waterdeep	waterdeep.are	8000	0 5 13		3	3000	Merchant guild eq and admission	
Brackish herbs to Blackstaff	Waterdeep	waterdeep.are	8000	5 1 1					
Golden Thorn to Blackstaff	Waterdeep	waterdeep.are	8000	6 1 1					
Dragons Blood to Blackstaff	Waterdeep	waterdeep.are	8000	7 1 1					
Ardikan's Death	Waterdeep	waterdeep.are	8000	8 2 1					No longer used, justice system handles it.
WD Armoury	Waterdeep	waterdeep.are	8000	10 4 8					
Hear a story from Volo	Waterdeep	waterdeep.are	8000	14 1 1					
Paid for Room	Any(questr)	waterdeep.are	8000	15 1 1					Used by most inn areas
Waterdeep Dungeon	Waterdeep	waterdeep.are	8000	16 5 *					
Meriki's Apraise Quest	Waterdeep	waterdeep.are	8000	21 4 13		2		4 points of appraise	
Timberuk - Tantras	Waterdeep	waterdeep.are	8100	0 3 1					
Timberuk - Shadowdale	Waterdeep	waterdeep.are	8100	0 3 2					
Dung sweepers Guild	Waterdeep	waterdeep.are	8100	3 5 17		1			
Leishan quest	Waterdeep	waterdeep.are	8100	8 3 1					
Chimney Sweep Quest	Waterdeep	waterdeep.are	8100	11 4 10		1	1000		
Griffon Riders Quest	Waterdeep	waterdeep.are	8100	15 5 **					This quest needs writing

Watch Pay Day	Waterdeep	waterdeep.are	8100	20 1 1					
Learn to Tan Quest	Waterdeep	waterdeep.are	8100	21 3 5					
Learn Weapon Smithing	Waterdeep	waterdeep.are	8100	24 5 11					
Waterdeep rp policy breech	Waterdeep	waterdeep.are	8100	29 1 1					
Ghost of Wdeep Quest	Waterdeep	waterdeep.are	8200	0 4 7					
Keep Locket ghost quest	Waterdeep	waterdeep.are	8200	0 4 9					
Fortune Hall Gambling	Waterdeep	waterdeep.are	8200	4 1 1					
Pregnant wife watch soldier	Waterdeep	waterdeep.are	8200	5 3					
Bostave Component Quest	Waterdeep	waterdeep.are	8200	8 4 15				Allowed to shop for magic items	
Vampire in Waterdeep	Waterdeep	waterdeep.are	8200	12 5 30		3	lots	choice of ioun stone	31 failed
Lords of Waterdeep code qust	Waterdeep	waterdeep.are	8200	17 5		2	some	heavy gold ring with ac	
Suspicious Merchant Smuggle	Waterdeep	waterdeep.are	8200	22 4		1	some	gloves steal or braclet detect evil	good and evil path
Sepentil Oghma Quest	Waterdeep	waterdeep.are	8200	26 5					
Play Drums berdusk	Waterdeep	waterdeep.are	8300	0 4 15					
Paladin quest checks	Waterdeep	waterdeep.are	8400	WHOLE BLOCK					
Rope 8160 Fray Prog	Waterdeep	waterdeep.are	8500	0 3 7					Changed to value5

Rope 8161 Fray Prog	Waterdeep	waterdeep.are	8500	3 3 7					Changed to value5
Arcana for Ondils Quest	Waterdeep	waterdeep.are	8600	0 1 1					
Geog High Wiz Faerdale		waterdeep.are	8600	1 1 1					
Sword of Heroes	Shilmista	shilmista.are	9300	0 4 5					
Sea elf necklace	Shilmista	shilmista.are	9300	4 4 10					
Shadows Cloak Song	Westgate	westgate.are	9400	0 4 13		2	some	Shadows cloak song skill	
Paid for a room in lucky drunk	Lucky Drunk	taverns.are	9800	0 1 1					
Barmaid in love	Lucky Drunk	taverns.are	9800	1 3 6		1	little	shortsword +1 and ring+1 cha if done right	
Mephit quest	Undermountain	um	11900	21 6					
Paladin Squire quest	HoJ	hallsofjustice.are	13800	0 4 15					
Paladin Help Thrimba	HoJ	hallsofjustice.are	13800	4 7 5					
Paladin Generosity	HoJ	hallsofjustice.are	13800	4 7 8					
Paladin Valour	HoJ	hallsofjustice.are	13800	4 7 13					
Paladin Honesty	HoJ	hallsofjustice.are	13800	4 7 18					
Paladin Fealty	HoJ	hallsofjustice.are	13800	4 7 33					
Paladin Law	HoJ	hallsofjustice.are	13800	4 7 40					
Paladin Honour	HoJ	hallsofjustice.are	13800	4 7 44					
Paladin Humility	Hoj	hallsofjustice.are	13800	4 7 48					
Paladin									

MagicItem WealthTithe	HoJ	hallsofjustice.are	13800	4 7 71					
Paladin Completed lessons	HoJ	hallsofjustice.are	13800	4 7 80					
Paladin Join squires Guild	HoJ	hallsofjustice.are	13800	11 3 7					
Rescue Sir Treavor	HoJ	hallsofjustice.are	13800	14 4 13		2		shield or bracer 2 ac	
Destroyed Feebov's appa	Feebov		13900	0 4 9					
Weapons Appraise quest	Abbey Sword		14100	0 4 10					
Started Collegium Quest	Highcastle	highcastle.are	14200	0 3 1					
Paladin mount quest	Highcastle	highcastle.are	14200	3 4 9					
Bardsong quest	Highcastle	highcastle.are	14200	7 4 13					
Healer Quest	Highcastle	highcastle.are	14200	11 4 7					
Noble Quest	Highcastle	highcastle.are	14200	15 4 8					
Given willowbark tea	Highcastle	highcastle.are	14200	19 1 1					
Eat food in the dining room	Highcastle	highcastle.are	14200	21 3					
Given bandages	Highcastle	highcastle.are	14200	20 1 1					
Save Tethir Forest	Tethyr	tethyr.are	15100	0 5 16					
Macumail give weapon	Tethyr	tethyr.are	15100	0 5 17					
Paid for bath at foaming sands	Tethyr	tethyr.are	15100	6 2 1					
Paid for room in Zass	Tethyr	tethyr.are	15100	8 1 1					
Wild elves aid foxfire	Tethyr	tethyr.are	15100	9 1 1					
Mistwalker Ship Destinations	Tethyr	tethyr.are	15100	10 3					
Paid for									

Mistwalker	Tethyr	tethyr.are	15100	13 1 1					
Find Golden Thorn	Tethyr	tethyr.are	15100	14 1 1					
Fletching quest	Tethyr	tethyr.are	15100	15 4					
Join the School of Stealth	Tethyr	tethyr.are	15100	19 5 15					
Join the wild elves	Tethyr	tethyr.are	15100	24 5 29					
Free Jill the Dwarf	Tethyr	tethyr.are	15100	29 2 2					
Alaundo manuscripts	Tethyr	tethyr.are	15200	0 5 19		2	some	melodic armour	using second vnum block in area
Alchemist quest	House of Wonder	housewonder.are	15400	0 3 5					
Help the researching wizards	House of Wonder	housewonder.are	15400	5 5					
Berry Dropping	House of Wonder	housewonder.are	15400	10 1					
Learn about Mystra	House of Wonder	housewonder.are	15400	11 5	2 religion	2	some		
Has joined spies guild	Spies Guild		15900	0 1 1					
Started the spies quest	Spies Guild		15900	1 1 1					
Training in spies guild	Spies Guild		15900	2 6 39					
Quest in spies guild	Spies Guild		15900	8 4 14					
Used (gambling hall)	Fortune Hall		15900	20 9 *					
Altar of Tymora	Fortune Hall		15900	29 1 1					
Joined gnome ills guild	Gnome Ills Gd		16100	0 1 1					
Non-gnome questing	Gnome Ills Gd		16100	1 1 1					

Quest 'hat of invis'	Gnome Ills Gd		16100	2 4 14					
Altar of Lathander	Roseportal Hse		16200	0 1 1					
Paid for Wave Runner	Shipping		16300	0 1 1					
Killed a sailor	Shipping		16300	31 1 1					
Killed sailors	Sword Coast Shipping		16300	31 1 1					
Paid for Waverunner	Sword Coast Shipping		16300	0 1 1					
Paid for Wave Reaver	Sword Coast Shipping		16300	1 1 1					
Kill giant leaders	Hartsvale Town		17100	0 5 25					
Murder for Cyric	God Temples	godtemples.are	19000	0 4 29		2	some	If Cyric follower, weapons +2	31 fail
Find Selantha's Husband	Skullport - Cent Trade	centradelanesp.are	21300	0 4 13		2	some	silk gloves with a pocket	
Lucky Drunk Vat	Skullport - Cent Trade	centradelanesp.are	21300	4 3 5		2	some	ring of coins - luck +1	
Dig Clay for Vizen	Skullport - Cent Trade	centradelanesp.are	21300	-			100	2 gold	
Find food for recipe for wurli	Skullport - Cent Trade	centradelanesp.are	21300	7 4 15		2	some	Random shield of resistance	
Bat fighting	Skullport - Cent Trade	centradelanesp.are	21300	11 2 1				Coins if they win	
Griffon quest	Longsaddle	longsaddle.are	22700	1 4 8		1		20 plat	
Paid to stable horse	Longsaddle	longsaddle.are	22700	5 1 1					
Manticore familiar problem	Longsaddle	longsaddle.are	22700	6 4 14		3	some	20 plat and a gem	15 failed, killed fuzzby
Brew Potion Quest	Longsaddle	longsaddle.are	22700	10 4					
	House of								

Lapidary Quest	Midnight	houseofmidnight.are	22800	13 4 11					
Shapechange Quest	House of Midnight	houseofmidnight.are	22800	9 4 12					
Moonbow Quest	House of Midnight	houseofmidnight.are	22800	5 4 11					
Oracle Quest	House of Midnight	houseofmidnight.are	22800	0 5 19					
Chakram Quest	House of Midnight	houseofmidnight.are	22800	0 5 29					
Shadow Walk Quest	House of Shadows	houseofshadows.are	22000	11 4 12					
Bracer Quest	House of Shadows	houseofshadows.are	22000	7 4 11					
Thief Guild Quest	Nightmasks	nightmasks.are	22100	0 5 16					
Daromar exotics	Darromar	darromar.are	23100	5 1 1					
Cluggan the gnome	Darromar	darromar.are	213100	6 5		2	some	stone skin ring	
Deck of Fate Tyr	Deck of Fate		23400	0 1 1					
Deck of Fate Mystra	Deck of Fate		23400	1 1 1					
Deck of Fate Mask	Deck of Fate		23400	2 1 1					
Deck of Fate Ilmater	Deck of Fate		23400	3 1 1					
Deck of Fate Moradin	Deck of Fate		23400	4 1 1					
Deck of Fate Gruumsh	Deck of Fate		23400	5 1 1					
Deck of Fate Chauntea	Deck of Fate		23400	6 1 1					
Deck of Fate Kelemvor	Deck of Fate		23400	7 1 1					
Deck of Fate Sune	Deck of Fate		23400	8 1 1					
Deck of Fate Mielikki	Deck of Fate		23400	9 1 1					

Deck of Fate Tempus	Deck of Fate		23400	10 1 1					
Deck of Fate Cyric	Deck of Fate		23400	11 1 1					
Deck of Fate Lathander	Deck of Fate		23400	12 1 1					
Deck of Fate Malar	Deck of Fate		23400	13 1 1					
Deck of Fate Gond	Deck of Fate		23400	14 1 1					
Deck of Fate Selune	Deck of Fate		23400	15 1 1					
Deck of Fate Tymora	Deck of Fate		23400	16 1 1					
Deck of Fate Loviatar	Deck of Fate		23400	17 1 1					
Deck of Fate Helm	Deck of Fate		23400	18 1 1					
Deck of Fate Beshaba	Deck of Fate		23400	19 1 1					
Deck of Fate Oghma	Deck of Fate		23400	20 1 1					
Deck of Fate Lloth	Deck of Fate		23400	21 1 1					
Deck of Fate Corellon	Deck of Fate		23400	22 1 1					
Deck of Fate Torm	Deck of Fate		23400	23 1 1					
Deck of Fate Yondalla	Deck of Fate		23400	24 1 1					
Deck of Fate Garl	Deck of Fate		23400	25 1 1					
Deck of Fate Talos	Deck of Fate		23400	26 1 1					
Deck of Fate Shar	Deck of Fate		23400	27 1 1					
Fletching Quest for Corellon	Leuthilspar	leuthilspar.are	23700	0 5 26		2	some	4 points of fletching skill	
Join Wings of	Wings of		25400	0 5 18				Become an	

Raven Enchant	Charm	wingsofcharm.are				2		enchanter	
Enchanters Transport quest	Wings of Charm	wingsofcharm.are	25400	5 5 31		3	reward	Scroll of transport	
Kill the Grick for the dwarf	Under Alaron	underalaron.are	25600	0 3 3				+1 con ring	
Lathanders Temple	Faerdale	faerdale.are	28400	18 4 12					
Elemental Globe	Faerdale	faerdale.are	28400	10 4 11	Nature 1	3	3000	Elemental globe	
Bandits Pass	Faerdale	faerdale.are	28400	0 4 8		2	2000	Faerdale item	
Crystal Chain	Faerdale	faerdale.are	28400	4 2 3				Crystal Chain	
Joined Moonclaws	Faerdale	faerdale.are	28400	6 4 2		1	1000	Join THieves guild	
Lead Moonclaws	Faerdale	faerdale.are	28400	6 4 4		1	1000	Soft leather	
Joined Velvet Hands	Faerdale	faerdale.are	28400	6 4 7					
Lead Velvet Hands	Faerdale	faerdale.are	28400	6 4 9					
Mystras Mantle	Faerdale	faerdale.are	28400	14 4 14	Arcana 1	2	3000	Ondils spellbook	8600 0 1 0 for knowledge arcana
High Wizards Apprent	Faerdale	faerdale.are	28400	20 5 25	Geog 1	3	5000		Quest to join necs guild
Kill High Wizard	Faerdale	faerdale.are	28400	20 5 26				Robes	
Character Generation									
Stat's Generation	Character Gen	chargen.are	30000						
Stat's Generation	Character Gen	chargen.are	30100						
Stat's Generation	Character Gen	chargen.are	30200	0 10					
Kismet Usage	Character Gen	chargen.are	30200	10 6					
Race Selection etc	Character Gen	chargen.are	30500	15 5					
Hometown			30500	20 5					

Selection	Character Gen	chargen.are							
Brewery Licence	Dwellings	dwellings.are	40000	0 1 1					
Giantspire Mountains Know G	Rashemen	wilds46.are	46000	0 2 3	Geog 1				
Rawlingswood Know Geog	Rashemen	wilds46.are	46000	2 2 3	Geog 1				
Umber Marshes Know Geog	Thay	wilds48.are	48000	0 2 3	Geog 1				
Thaymount Know Geog	Thay	wilds48.are	48000	2 3 3	Geog 1				
Sea of Moving Know Geog	The Far North	wilds50.are	50000	0 2 3	Geog 1				
Icewind Dale Know Geog	The Far North	wilds50.are	50000	2 2 3	Geog 1				
Spine of the World Know Geog	The Far North	wilds50.are	50000	4 2 3	Geog 1				
Lurkwood Know Geog	The Far North	wilds50.are	50000	6 2 3	Geog 1				
Moonwood Know Geog	The Far North	wilds50.are	50000	8 2 3	Geog 1				
Anauroch Know Geog	The Far North	wilds50.are	50000	10 2 3	Geog 1				
High Forest Know Geog	The North	wilds51.are	51000	0 2 3	Geog 1				

Neverwinters Wood Know Geog	The North	wilds51.are	51000	2 2 3	Geog 1				
Evermoors Know Geog	The North	wilds51.are	51000	4 2 3	Geog 1				
Nether Mountains Know Geog	The North	wilds51.are	51000	6 2 3	Geog 1				
Kryptgarden Know Geog	Sword Coast	wilds52.are	52000	0 2 3	Geog 1				
Meer Dead Men Know Geog	Sword Coast	wilds52.are	52000	2 2 3	Geog 1				
Westwood Mountains	Sword Coast	wilds52.are	52000	4 2 3	Geog 1				
Greypeak Mountains	Sword Coast	wilds52.are	52000	6 2 3	Geog 1				
Ardeep Knowledge Geography	Anauroch	wilds53.are	53000	0 2 3	Geog 1				
High Moor Knowledge Geog	Anauroch	wilds53.are	53000	2 2 3	Geog 1				
Weathercote Wood Geog	Anauroch	wilds53.are	53000	4 2 3	Geog 1				
Misty Forest Know Geog	Trackless Sea	wilds54.are	54000	0 2 3	Geog 1				
Marsh of Chelimber Geog	Trackless Sea	wilds54.are	54000	2 2 3	Geog 1				
Forgotten Forest Geog	Trackless Sea	wilds54.are	54000	4 2 3	Geog 1				
Border Forest Geog	Trackless Sea	wilds54.are	54000	6 2 3	Geog 1				
Dragonspine Forest Geog	Trackless Sea	wilds54.are	54000	8 2 3	Geog 1				
Great Grey Land Thar Geog	Trackless Sea	wilds54.are	54000	10 2 3	Geog 1				
Galena Mountains Know Geog	Trackless Sea	wilds54.are	54000	12 2 3	Geog 1				
Trollbark Forest Know Geog	Sea of Swords	wilds55.are	55000	0 2 3	Geog 1				

Serpent Hills Know Geog	Sea of Swords	wilds55.are	55000	2 2 3	Geog 1				
Desertsmouth mountains Geog	Sea of Swords	wilds55.are	55000	4 2 3	Geog 1				
Cormanthyr Know Geog	Sea of Swords	wilds55.are	55000	6 2 3	Geog 1				
Moonshaes Know Geog	Sea of Swords	wilds55.are	55000	8 2 3	Geog 1				
Forest of Wyrms Know Geog	Dalelands	wilds56.are	56000	0 2 3	Geog 1				
Earthspur Mountains Geog	Dalelands	wilds56.are	56000	2 2 3	Geog 1				
Flooded Forest Know Geog	Dalelands	wilds56.are	56000	4 2 3	Geog 1				
Reaching Woods Know Geog	Heartlands	wilds57.are	57000	0 2 3	Geog 1				
Trielta Hills Know Geog	Heartlands	wilds57.are	57000	2 2 3	Geog 1				
Fields of the Dead Geog	Heartlands	wilds57.are	57000	4 2 3	Geog 1				
Sunset Mountains Know Geog	Corymyr	wilds58.are	58000	0 2 3	Geog 1				
Storm Horns Know Geog	Corymyr	wilds58.are	58000	2 2 3	Geog 1				
Kings Forest Know Geog	Corymyr	wilds58.are	58000	4 2 3	Geog 1				
Thunder Peaks Know Geog	Corymyr	wilds58.are	58000	6 2 3	Geog 1				
Vast Swamp Know Geog	Corymyr	wilds58.are	58000	8 2 3	Geog 1				
Woods of Sharp Teeth Geog	Sembia	wilds59.are	59000	0 2 3	Geog 1				
Cloak Wood Know Geog	Sembia	wilds59.are	59000	2 2 3	Geog 1				
Cloud Peaks Know Geog	Tethyr	wilds60.are	60000	0 2 3	Geog 1				
Snowflake Forest Know Geog	Tethyr	wilds60.are	60000	2 2 3	Geog 1				

Snakewood Know Geog	Tethyr	wilds60.are	60000	4 2 3	Geog 1				
Troll Mountains Know Geog	Tethyr	wilds60.are	60000	6 2 3	Geog 1				
Forest of Tethir Know Geog	Amn	wilds61.are	61000	0 2 3	Geog 1				
Snowflake Mountains Geog	Amn	wilds61.are	61000	2 2 3	Geog 1				
Forest of Mir Know Geog	Amn	wilds61.are	61000	4 2 3	Geog 1				
Purple Hills Know Geog	Amn	wilds61.are	61000	6 2 3	Geog 1				
Calim Desert Know Geog	Calimshan	wilds62.are	62000	0 2 3	Geog 1				
Marching Mountains Geog	Calimshan	wilds62.are	62000	2 2 3	Geog 1				
Spider Swamp Know Geog	Calimshan	wilds62.are	62000	4 2 3	Geog 1				
Dwarven Mining Tunnels Geog	Calimshan	wilds62.are	62000	6 2 3	Geog 1				
Dark Dominion Know Geog	Calimshan	wilds62.are	62000	8 2 3	Geog 1				
Eastways Know Geog	Calimshan	wilds62.are	62000	10 2 3	Geog 1				
Clean the shrines	Medusa's Castle	medusascastle.are	101100	0 4 8					
Prayed at Myrkles shrine	Medusa's Castle	medusascastle.are	101100	4 1 1					
Prayed at Tempus shrine	Medusa's Castle	medusascastle.are	101100	5 1 1					
Prayed at Talos shrine	Medusa's Castle	medusascastle.are	101100	6 1 1					
Prayed at Talonas shrine	Medusa's Castle	medusascastle.are	101100	7 1 1					

Prayed at Banes shrine	Medusa's Castle	medusascastle.are	101100	8 1 1					
Enchanted Medusa's head	Medusa's Castle	medusascastle.are	101100	9 1 1					
Weave disruption quest	Medusa's Castle	medusascastle.are	101100	10 3 5					
Found medusa's spellbook	Medusa's Castle	medusascastle.are	101100	13 1 1					
Found witches spellbook	Medusa's Castle	medusascastle.are	101100	14 1 1					

QUEST BITS TUTORIAL

Quest bits run from 0 to 32. In order to understand quest bits it is good to be able to understand the binary number system. Each bit has an on and off position, just as binary digits do. In order to determine the step number for a bit, you would calculate the decimal number from the binary number or vise versa.

So for instance you have an 8 step quest. You would need to use 3 bits, assuming that the quest is done in order. So the quest bits would start at 0 and run for 3 bits. So to check if a characters quest is completed with the final eigth step you would do:

```
if quest(0, 3, $n) == 8
```

For further explanation, here is a brief set of possible quest sets.

```
if quest(0, 1, $n) == 0
if quest(0, 1, $n) == 1
```

```
if quest(0, 2, $n) == 0
if quest(0, 2, $n) == 1
if quest(0, 2, $n) == 2
if quest(0, 2, $n) == 3
```

```
if quest(0, 3, $n) == 0
if quest(0, 3, $n) == 1
if quest(0, 3, $n) == 2
if quest(0, 3, $n) == 3
if quest(0, 3, $n) == 4
if quest(0, 3, $n) == 5
if quest(0, 3, $n) == 6
if quest(0, 3, $n) == 7
```

You can see how the number of possible steps climbs quickly, so you don't need many bits to do a full quest.

If you have a second quest in the area and you have used (0, 3, $n) as your first quest set, the second quest set can begin at (3,x,$n) where x is the number of bits you need to complete your second quest. Notice that (3,x,$n)==1 has no spaces. The spaces are not necessary for the code to work, but can be helpful when looking for possible problems in your code.

Quest Bits Tutorial by Grimace

Grimace (Area Creator from Mortal Realms Mud) has wrote the following tutorial for MRMud. We use the same quest bits system, so it applies to here as well:

Think of quest bits as a set of 32 switches that can be on or off. Let's take the simplest case. If you have ONE switch, it can either be off or on. You could write a simple quest where once the user has said "Grimace forever!", the mob changes the switch from 0 to 1 (off to on). Since you have 32 such switches to use, just use the first one (offset = 0, ie: at the beginning): mpmset $n quest 0 1 1. In this case the offset is 0, the quest is using only 1 bit of width (1 switch) and the

value is 1 ("ON"). Pretty easy, eh? It might look like this:

```
>speech_prog p Grimace forever!~
say Damned straight.
mpmset $n quest 0 1 1
~
```

Now say you want a quest with FOUR steps. Well, binary tells us that we can get 4 unique numbers from TWO switches (A off, B off; A on, B off; A off, B on; A on, B on). So... let's tack this quest on after the first. Since we've already used 1 switch for the first quest, we'll need an offset of 1 so we start at the second bit. Since we want 4 unique positions, we need 2 switches, so the width will be 2. The values will range from 0 to 3, so it'll look like this: mpmset $n quest 1 2 1 or mpmset $n quest 1 2 2, etc., all the way up to mpmset $n quest 1 2 3. It'll screw it up if you try to set the value greater than 3. Here's an example:

```
>give_prog i9700~
if quest(1,2,$n) == 0
  say Aha.  That's a nice... item... you have there.  Bring me a few more.
  mpmset $n quest 1 2 1
else
  if quest(1,2,$n) == 1
    say Two items.  That's spiffy.  I need another.
    mpmset $n quest 1 2 2
  else
  if quest(1,2,$n) == 2
    say THREE wonderful items.  Mu  ha  ha  ha.  Here's your reward.
    cast 'change sex' $n
    mpmset $n quest 1 2 3
  else
    if quest(1,2,$n) == 3
      say I don't need any more of those.  Go away.
    endif
  endif
 endif
endif
~
```

Getting the hang of it? You can divide the 32 switches into whatever pieces you want. You can have 32 1-bit quests (on/off) or you can have 1 32-bit quest (2^32 = billions of values) or anything in between. Simply decide how many steps you need, then figure out how many switches you'd need to group together to get that number of values, then pick a non-overlapping offset value to set aside a chunk of your 32 switches for this particular quest. (By the way, the equation to

figure out how many bits you need for a certain number of steps is this: 2^#bits = #steps. It's better to set aside way too many and not use them than to set aside not enough. I tend to make my quests 4 bits wide as a rule. 16 steps is a nice round number unless you want to do silly stuff like count mobs killed, like in mist city... So you'd have three 16-step quests like a 0 4 quest, and a 4 4 quest, and a 8 4 quest, with values from 0 to 15, and then maybe a couple flags like 9 1 and 10 1, with values of 0 and 1 only.

Other notes:

1. All bits for every char start in the "off" position (value = 0).
2. Every pc and each mob has a unique set of 32 switches for each area. If you set or check quest bits, it will automatically use the bits from the area the mob or pc is from, even if it's for an item or mob from another area. Objects however only check the area they are currently in so they should use questr.
3. Generally a mob_prog will have a bunch of nested "if else" statements (remember an "endif" for EACH "if") which check the quest bits, perform some cool slight of hand, and then set the quest bits. It's a generic formula.

Feel free to ask any member of the area administration if you have any questions. Quest bits are cool. Use them everywhere.

- Grimace

QUEST BITS ON OBJECTS

Builders now have the ability to set quest bits on items. Quest bits can be set on ALL of the values, but it is recommended in most instances that bits only be set on value5. The other values are used by objects for different settings, value5 is not used by any object.

```
mposet $o value5bits
mpoadd $o value5bits
if value5bits(,,$o) ==
```

They work just like

```
mpmset $n quest
mpmadd $n quest
if quest(,,$n) ==
```

Up to now, we could store (easily) only one value on objects, in value5. That works well for, say, a charged item with 4 charges of a given spell. In this case, the object's value5 represented the number of charges left. Yet, it was quite tricky to code objects with charges of more than one spell (for example, a ring of spell storing for cure light wounds, levitate, and bless).

The new commands allow us to slice value5 into several parts. In the example of the ring of spell storing for cure light wounds, levitate and bless, we could use 1 bit to indicate whether or not the cure light wounds charge is available or not, another bit for the levitate charge, and a third bit fot the bless charge. If we number those bits 0, 1, and 2, we can now use the following programs:

```
>intercept_prog curelighttrigger~
if wear_loc($o) != -1
  if value5bits(0,1,$o) == 1
    cast 'cure light' $n
    mposet on $n $o value5bits 0 1 0
  else
    mpechoat $n Nothing happens.
```

```
  endif
else
  mpechoat $n You should wear the ring first.
endif
~
>intercept_prog blesstrigger~
if wear_loc($o) != -1
  if value5bits(2,1,$o) == 1
    cast bless $n
    mposet on $n $o value5bits 2 1 0
  else
    mpechoat $n Nothing happens.
  endif
else
  mpechoat $n You should wear the ring first.
endif
~
>intercept_prog curelightrecharge~
if wear_loc($o) != -1
  if value5bits(0,1,$o) == 1
    mpechoat $n The ring does not need to be recharged.
  else
    if skilllevel($n,cure light) < 6
    or manaamt($n) < 40
      mpechoat $n That does not work.
    else
      mpmadd $n currmana -40
      mposet on $n $o value5bits 0 1 1
      mpechoat $n You recharge the ring.
    endif
  endif
else
  mpechoat $n You should wear the ring first.
endif
~
```

Another application for these new commands: objects with "memory", that is, objects that should react to a sequence of actions. An in-game example: the harp in Shadowdale. The "reaction" of the objects depends on the last five notes that have been played.

Coding such an object without the new commands required to use base-7 representation of numbers (after numbering the strings/notes from 0 to 6). Value5 was used to store a number representing the strings that have been plucked. That yielded programs that are hard to read and hard to modify.

```
0,3 : number of first string plucked
3,3 : number of second string plucked
6,3 : number of third string plucked
9,3 : number of fourth string plucked
12,3 : number of fifth string plucked
(eventually: 15,2 : number of strings already plucked)
```

Initially, we would have

```
value5bits 0,3 == 0
value5bits 3,3 == 0
value5bits 6,3 == 0
value5bits 9,3 == 0
value5bits 12,3 == 0
value5bits 15,2 == 0
```

If someone plucks string 4, the values become

```
value5bits 0,3 == 4
value5bits 3,3 == 0
value5bits 6,3 == 0
value5bits 9,3 == 0
value5bits 12,3 == 0
value5bits 15,2 == 1
```

If someone then plucks strings 3, 2 then 5, the values become

```
value5bits 0,3 == 4
value5bits 3,3 == 3
value5bits 6,3 == 2
value5bits 9,3 == 7
value5bits 12,3 == 0
value5bits 15,2 == 4
```

At this point, if someone plucks a fifth string, we can trigger an effect if it is needed.

The object could have intercept progs for each string (plucka, ..., pluckg for example). If the A string is coded as 0, B as 1, ..., D as 3, ..., G as 6, the programs will look like this:

```
>intercept_prog pluckd~
if value5bits(15,2,$o) == 0
  mposet value5bits $o 0 3 3
endif
if value5bits(15,2,$o) == 1
  mposet value5bits $o 0 6 3
endif
if value5bits(15,2,$o) == 2
  mposet value5bits $o 0 9 3
endif
if value5bits(15,2,$o) == 3
  mposet value5bits $o 0 12 3
endif
if value5bits(15,2,$o) == 4
  mposet value5bits $o 0 15 3
endif
mpoadd $o value5bits 15 2 1
if value5bits(15,2,$o) == 5
  (check for effect)
  (then reset the values:)
  mposet $o value5bits 0 17 0
endif
~
```

The same could be used for any object to which several sequences of actions can be applied.

SAMPLE QUEST

With this I will attempt to explain how quest bits work using an example of a small 3 bit quest. Mobile 1 wants you to kill Mobile 2. In this quest sample we will show how it works for a p layer doing the quest solo. As in they HAVE to get the death blow.

Program on mobile one would read:

```
>greet_prog 100~
if quest(0,2,$n) == 0
  sayto $n Go kill Mobile 2 for me.
  sayto $n Go present yourself to Mobile 3 when you have done so.
  mpmset $n quest 0 2 1
endif
~
|
```

Player wanders off and kills Mobile 2. The program on Mobile 2 would be.

```
>death_prog 100~
if quest(0,2,$n) == 1
  sayto $n Tell Mobile 3 I will haunt them forever!
  mpmset $n quest 0 2 2
endif
~
|
```

Player goes to Mobile 3 as instructed and says to Mobile 3 that Mobile 2 will haunt them forever. The word haunt triggers a speech program. The program on Mobile 3 would be.

```
>speech_prog haunt~
if quest(0,2,$n) == 2
  sayto $n Ho! So he thinks to haunt me!
  sayto $n Here is your reward.
  mpoload QQ01
  mpgive QQ01 $n
  mpechoat $n Mobile 3 hands you Object Number 1.
```

```
    mpechoaround $n Mobile 3 hands $n Object Number 1.
    mpmadd $n exp 1000
    mpmadd $ qp 1
    mpmset $n quest 0 2 3
  endif
  ~
  |
```

So PC walks away with 1000 experience, 1 glory point and a nifty little object. If they were asked to kill someone evil then perhaps you would raise their lawful and their good. If they were asked to kill someone good then you would lower it.

Now this quest was written using $n. This means it is pretty much a solo quest. They must get the death blow. Most of the quests on FK are orientated towards groups doing the quest. To make it doable by a group, where it would not matter who in the group gets the death blow you would replace the $n with $W. This means the quest bits are set on the group leader.

GREET PROGRAMS

A greet program is activated when a PC walks into the room. You can set the percentage chance of the program activating. Note that the mobile must be able to see the PC in order to activate. For instance if the room is dark, and the mobile cannot see in the dark the program will not activate.

The following prog is from Danilo in Waterdeep (when there was a mob Danilo).

```
>greet_prog 100~
if sex($n) == 2
  mpechoat $n You have entered an all male domain!
  mpechoat $n $I gives you an astonished look.
  mpechoaround $n $I stares in amazement at $N.
  mpechoat $n $I takes you gently by the arm and escorts you to to door.
  mpechoaround $n $I escorts $n to the door.
  sayto $n I am sorry my dear, but this is not
  sayto $n the place for a Lady such as yourself.
  mptransfer $n 8061
else
  if sex($n) == 1
    sayto $n Welcome Sir, what can we do for you?
    sayto $n Would you like to buy some of my fine wares?
  else
    sayto $n Well I surely don't know what you
        sayto $n are but you are not welcome here.
    mpechoat $n The guards come to escort you away.
    mpechoaround $n The guards come to escort $N away.
    mptransfer $n 8001
    say Well I sure don't know what IT was!
  endif
endif
~
|
```

This program activates EVERY time a PC walks in. Danilo then checks the sex of the player. 1 = Male and 2 = Female. If the PC is female there is a series of echos at and around the player before they are transfered to outside the shop. If they are male Danilo offers to serve them. If they are neutral (yes it does happen occasionally with spells like change sex) they are transferred to the southern gates of Waterdeep.

Note the use of $I. This will take the mobiles short descripton and out put it. It is quicker and easier than typing the

mobile's short description. And if the short description was to change for some reason, then the program will remain accurate. $N was also used in the mpechoarounds, so that it echos the adjective if the character is ungreeted.

The following is *part* of a prog on a Waterdeep Soldier.

```
>greet_prog 40~
if iswanted($n)
  yell I have found a criminal.  Its $N!
  mpechoat $n The soldier grabs you and drags you off to the dungeon.
  say You shall pay for your crimes $N.
  mptransfer $n 8115
  mpecho $n is dragged off to the dungeon.
  mpgoto 8199
endif
~
|
```

This program only activates 40 percent of the time. The soldier makes a check for if the character is wanted by the justice system, yells that they have found the character and then transfers the character to the dungeon in Waterdeep. Note that in using $N it means they will not yell out the players name UNLESS it is known by other players. If they don't know the player they should see the players adjective.

You can also specify a direction in greet progs. This is the direction that the PC has arrived from.

```
>greet_prog 100 south~
mpechoat $n You arrived from the south.
mpechoaround $n $N arrived from the south.
~
|
```

DEATH PROGRAMS

Death programs activate on the death of a mobile. You can set the percentage chance that they have of activating.

```
>death_prog 100~
if questr(15100, 19, 5, $n) == 6
  mpoload 8070
  mpmset $n questr 15100 19 5 7
  mpechoat $n You have killed one of the secret Lords of Waterdeep!
  mpechoaround $n $n has killed one of the secret Lords of Waterdeep!
endif
~
|
```

The above program is on an undisclosed mobile in Waterdeep. It happens 100 percent of the time. However what it does will not happen unless the conditions are met. This particular program checks for quest bits in ANOTHER area. It is looking at the area with the vnums that start with 15100. You can check quests in different areas using the questr method. If these bits are correct in that area the prog loads up an object (head of the victim), sets the bits up one step on the PC and then echos at the player and around the player.

```
>death_prog 100~
if quest(4, 2, $n) == 2
  mpoload 28433
  mpoload 28434
  mpmadd $n exp 500
  mpmadd $n lawful -100
  mpmadd $n good -100
  mpmset $n quest 4 2 3
endif
~
|
```

The above program is the death prog on mobile who is a citizen of an area. It checks to see if the PC has those quest bits set right before loading up two objects. It gives them 500 experience points before adjusting their lawful and good down.

Most of the time because death progs are associated with quests you will want them to happen 100% of the time. Occasionally there might be an item on a mobile that you want to load only a small chance of the time. So a rare and special object could be loaded this way. Of course there are other uses for such a prog.

```
>death_prog 1~
mpoload 28431
~
|
```

In the above program there is a 1% chance of this object being loaded on the death of this mobile.

GIVE PROGRAMS

Give programs are activated when a mobile is given a certain object.

```
>give_prog i8140~
if quest(10, 4, $n) == 2
  mpecho Jonathon grabs a rag and starts to polish the shield.
  sayto $n This will do nicely!
  sayto $n While you were gone I had another order.
  sayto $n An adventurer in town for a short time is
  sayto $n in need of a hardened leather helm.
  sayto $n I hear Rebeleigh stocks it.
  mpecho Jonathon turns back to polishing his shield assuming you will get him what he
asks.
  mpmset $n quest 10 4 3
  mpjunk i8140
endif
~
|
```

This is one of the give programs on Jonathon the Armour in Waterdeep. Note that there is an `i` before the vnum of the item that he wants. This is important or the give program will not work. This program activates when the player gives Jonathon a polished shield, it then checks that the quest bits on the player are right before proceeding with the rest of the program. He talks to the player a little before setting their quest bits up another step. He then junks the item. Most of the time it is necessary for items to be junked. For instance the mob could then be killed for an item, or they would sell it in the shop and so on. Leaves too much room for cheating not to junk the item in most cases.

```
>give_prog i28428~
if quest(6, 4, $n) == 1
  mpjunk i28428
  sayto $n Ahhhhhhhhhhhhhhhh.
  mpecho The Moonclaw Leader carresses the gem reverently.
  sayto $n This makes you worthy of becoming a member of our guild.
  mpoload 28423
  mpgive badge $n
  mpechoat $n The guild leader hands you a badge.
  mpechoaround $n The guild leader hands $N a badge.
  sayto $n Wear this with pride as we are the superior Theives
  sayto $n Guild in the city of Faerdale.
```

```
    mpmset $n quest 6 4 2
    if class($n) == Rogue
      if guild($n) != Bards
        if guild($n) != Thieves
          mpmset $n guild Thieves
          mpechoat $n You have joined the guild of thieves.
        endif
      endif
    endif
    mpmadd $n lawful -100
    mpmadd $n good -100
    mpmadd $n exp 1000
    mpmadd $n qp 1
  endif
  ~
```

This give program is a bit more complicated. This is the give program on the Moonclaws leader in Faerdale. The PC gives the leader a gem hes been asked to steal. The program checks that the quest bits are correct and then executes the rest of the program. Note that if the quest bits are not right the leader will just keep the gem and nothing will happen. If you wish you could put an else before the last endif of the prog handing the gem back to the PC, or keep it saying something to the effect of Thanks for the goodies. If the quest bits are correct the program junks the gem, talks the player and loads the badge. Note that we use `mpgive` to give the badge to the player. If you use normal give, and the players hands are full then they cannot give the item to the player. Mpgive bypasses this. The player gets the item reguardless of if they are maxed out on what they can hold and their carry weight. However there is NO echo for mpgive so you have to do the echo, which you can see we have done here. It is often good to do a echo around the player getting the object too, so other players in the room can see that the PC was given an object. The prog then sets the players quest bit up.

After this point the program gets a little more complicated and starts checking the base class of the PC, and if they are NOT in the bards or the thieves guilds. If they are not in any of these guilds it then sets them into the thieves guild. If you are wanting to place a PC in one of the guilds as a result of your quest RUN IT BY US first. Do not do so without asking permission. The program then goes on to give experience, change the players lawful and good and give them a glory point.

SPEECH PROGRAMS

Speech programs are triggered by a keyword or a key phrase that is used in says or sayto.

```
>speech_prog ondil ondils ondil's~
if quest(14, 4, $n) == 1
  mpecho The lich throws back his head and laughs insanely.
  sayto $n So you seek Ondil's Book of Spells?  You will have
  sayto $n to battle me for that information, and if you win
  sayto $n I may tell you.
  mpmset $n quest 14 4 2
  mpkill $n
endif
~
|
```

In this prog a PC saying any of the words, ondil, ondils or ondil's will trigger this prog. The prog then checks the quest bits on the PC before doing some sayto's, an echo and setting up the quest another bit. After this the mobile attacks the PC with the mpkill command. You will notice that often normal commands in mob progs are preceeded by an mp. You must use the mp version of the command in mob progs. If there is variations on saying a particular word then it is good practice to put in those variations as has been done with the word ondil.

The following prog is not triggered on a keyword but rather a phrase. When using a phrase, it is preceeded by a **p**. This means the PC must say the exact phrase, including punctuation and case. This is a much harder quest to do, so at some point your quest should have given a good clue of what to say.

```
>speech_prog p Arilyn sent me~
if quest(0, 5, $n) == 7
  nod $n
  sayto $n Arilyn wants the help of the Lythari?
  sayto $n I will take you to the Lythari elders.
  mptransfer $n 15263
  mpgoto 15263
  mpechoat $n You feel odd as you go between worlds.
  mpforce $n look
endif
~
|
```

This is the prog from the lythari in the elf quest in Tethir Forest. The player must say the phrase *Arilyn sent me* exactly. No change in case, no change in punctuation. I actually have this prog repeated with a few different variations on the phrase to make it easier for players. Once the phrase is said this quest checks the quest bits of the PC. Talks to the PC and then transfers them to a room in the lythari den. Mptransfer transfers the PC only. If you want $n's pet to go as well you need to add the argument pet. To do that it would have to be `mptransfer $n pet`. It will not transfer whole groups. To do that it would have to be `mptransfer all`. The mob then follows the player with an mpgoto and forces the player to look. This is important. You can actually do a mptransfer and the player has no idea that it has happened. Helpful in some quest situations, but if you want the PC to know they have moved you have to mpforce them to look. If you are ever testing a quest with an imm character on the test port, bare in mind that mobs cannot force an immortal. Therefore testing is best done with a mortal.

BRIBE PROGRAMS

Bribe progs are triggered by the giving of coins to the mobile.

```
>bribe_prog 1000~
sayto $n Fare Thee Well $n.
sayto $n I hear that in Faerdale a young hero is needed.
mpecho The priest utters a few strange words.
mpecho The priest reaches into a pouch and throws something onto the ground.
mpechoat $n You cough and splutter at the smoke that rises up from the ground.
mpechoat $n You find yourself elsewhere!
mpechoaround $n The smoke produced makes you cough and splutter.
mpechoaround $n You look up and $N is gone!
mptransfer $n 28464
mpat 28464 mpforce $n look
~
|
```

This prog WAS on the Priest of Ilmater in Waterdeep. If he is given the equivalent of 1000 copper he will perform his program of several echos and transfering the PC to the Master of Faerdale. The amount of the bribe prog is always in copper. What you tell the players is up to you. For instance in this case, 10 gold works, or 5 platinum, or 1000 copper, or 100 silver, or 50 electrum. However, despite this it is a good idea to give players a definative amount to work with.

```
>bribe_prog 50~
mpechoat $n The captain guides you aboard his ferry.
mpechoaround $n The Captain guides $N aboard his ferry.
mpecho The Captain pushes off across the river.
mpechoat $n Before you know it you are at the other side of the river.
mptransfer $n 15102
mpat 15102 mpforce $n look
~
|
```

This program is on the ferry master on one side of the river in Zazesspur. For 50 copper or the equivalent he will transfer the player from one side of the river to the other. Notice that he forces the player to look at the other side.

INTERCEPT PROGRAMS

Intercept programs intercept a command that is inputed into the game. They work on mobiles in the room and objects in inventory and on the ground and in rooms themselves. They are used to make up commands or change what would happen with an existing command.

```
>intercept_prog shape~
if inroom($n) == 15254
  mpechoat $n You carve the stick into an arrow shaft.
  mpoload 15172
  mpgive i15172 $n
  mpmset $n quest 15 4 1
  mpjunk i15167 $n
else
  mpechoat $n You really should do this with the wild elf elder's supervision.
endif
~
|
```

This program is on an object, that is used in the arrow making quest in Tethir Forest. With this I created the command *shape*. The PC would type shape into the game and if the program would activate. First it checks to see that the player is in a certain room (in this case with the wild elf elder in Tethir Forest), and then it loads up another object and gives it to the player. There is no echo saying that the PC is given the object as we do not want them to see such an echo. It sets the bit up and junks the object that the program is on. Do NOT junk objects that progs are on until the last line of the program. Junking them before the end means the program will not continue after the junk line. In this case if the PC is in the wrong room, it gives them a message saying so.

```
>intercept_prog sleep~
sayto $n Hey! What do you think you doing sleeping in my shop!
sayto $n Go pay for a room in an Inn.
~
|
```

```
>intercept_prog request~
if deity($n) == Mystra
  mpunintercept
else
  sayto $n Such armour is reserved for the followers of Mystra.
endif
~
|
```

This program intercepts the request command. It is on the priest of Mystra in her temple in Waterdeep. We didnt want anyone getting the armour he wears, and only followers of Mystra. So this program does a check for if the PC follows Mystra, if they do the command continues as normal. This is what the `mpunintercept` does. If the PC does not follow Mystra the n it echos at them, and the command does not execute.

RAND PROGRAMS

Rand progs continually activate over and over and over when someone is in the area. They are a resource intensive prog so they should only be used if no other prog type will be able to do the job.

```
>rand_prog 1~
mpecho The moonclaw trips over his own feet.
mpecho It makes you wonder what sort of thief he makes!
~
|
```

This program is on the Moonclaw Guild Member in Faerdale. This one is a fairly simple program that will have a 1% chance of happening every tick. For things like that that happen over and over while a player is in the room, keep the percentage very low so as to not be spammy.

```
>rand_prog 10~
if wear_loc($o) != -1
  if manaprcnt($r) < 95
    if questr(28500, 7, 6, $r) < 60
      mpechoat $r You feel a surge of power.
      mpmadd $r currmana 10
      mpmadd $r questr 28500 7 6 1
    else
      mpechoat $n You crystal chain of power shatters into a million pieces.
      mpjunk i28418
    endif
  endif
endif
~
|
```

This program is on a magical object in an area. It has a 10% chance of triggering each tick. It checks to see if the item is worn, and that the PC's mana percentage is less than 95%. It also checks the quest bits in a given area. If all these conditions are satisfied it gives back 10 mana to the wearer and adds 1 to their quest bits. After 60 uses the object is junked. Often items like this stop working or are junked after so many uses. This allows for a super powerful item to be in the game, without disturbing game balance.

Another option instead of using quest bits in the area, would have been to use checks on value5. This would have saved quest bits.

```
>rand_prog 100~
if rand(40)
  if quest(0, 5, $r) == 6
    mpechoat $n Blackstaff looks up at you.
    mpechoaround $n Blackstaff looks up at $n.
    sayto $r You must be hungry.
    emote utters some strange words and moves his hands.
    mpquiet on
    mpoload 8154
    drop i8154
    mpoload 8111
    drop i8111
    mpquiet off
    mpechoat $n Blackstaff places a fine feast of food before you.
    mpechoaround $n Blackstaff places a fine feast of food before $n.
    mpechoat  Blackstaff returns to his studies.
    mpmset $r quest 0 5 7
  else
    if quest(0, 5, $r) == 7
      sayto $r Ahhh here we go.
      sayto $r I can cure her, but I will need some ingredients.
      mpecho Blackstaff turns the pages of the tome he had been reading through.
      mpmset $r quest 0 5 8
    else
      if quest(0, 5, $r) == 8
        sayto $r Now they wont be easy to find.
        sayto $r I need Brackish Herbs, Dragons Blood, and Golden Thorn.
        say Now let me think, where can these be found
        mpmset $r quest 0 5 9
      else
        if quest(0, 5, $r) == 9
          sayto $r Talk to Robyn Kendrick in Corwell about Brackish Herbs.
          sayto $r I understand dragons are rampaging in near Zhentil Keep.
          sayto $r Talk to Foxfire in Tethir Forest about the Golden Thorn.
          say Be quick.  I will prepare the rest of the spell while you are gone.
          mpmset $r quest 0 5 10
        endif
      endif
    endif
  endif
endif
if quest(0, 5, $r) == 10
```

```
    if quest(5, 1, $r) == 1
      if quest(6, 1, $r) == 1
        if quest(7, 1, $r) == 1
          mpecho Blackstaff adds the final ingredients to his potion.
          mpecho Blackstaff utters some strange words.
          mpecho Blackstaff swirls the potion and it shines an iridescent purple.
          mpoload 8048
          mpechoat $r Blackstaff hands you the potion.
          mpechoaround $r Blackstaff hands $r the potion.
          mpgive dragolish $r
          mpjunk all
          sayto $r Take this and give it to the Merchants wife.
          sayto $r It should cure her.
          mpechoat $n Blackstaff gives you the potion.
          mpechoaround $n Blackstaff gives $n the potion.
          mpmset $r quest 0 5 11
          mpmset $r quest 5 1 0
          mpmset $r quest 6 1 0
          mpmset $r quest 7 1 0
        endif
      endif
    endif
  endif
  if position($r) == 4
    wake $r
    sayto $r Hey! What do you think you doing sleeping here!
    sayto $r Go pay for a room in an Inn.
  endif
  ~
  |
```

Rand progs can be used to slow down events happening. For instance this program stops Blackstaff s peaking too quickly when he had a lot of information to give to the player. It gives the player a chance to read the information before it scrolls off the screen. It is a very long and complicated program. It actually does more than one part of the quest. The first part is Blackstaff talking to the player when the first visit, the second is where they have handed him the ingredients and the final checks to make sure they have not slept in his study.

This who rand prog has a 100% chance of triggering - though I say it is 100% chance in actuality it does not activate that quickly I have found. After that in order to slow down the first part of the prog I have used an if rand(40) within the prog. This means that this part of the prog will activate 40% of the time while the rest will activate 100% of the time. You will notice that Blackstaff sets quest bits on the player as he talks. This is so he does not say the same thing each time the rand prog is activated. You must remember to set the bits up one to stop it repeating over and over again.

Another point to note is the use of $r in the rand prog. If you use $n the program will not work. You MUST use $r in rand progs.

FIGHT PROGRAMS

Fight programs activate a certain percentage of the time while the mobile is fighting. It should be noted than when there is no fight program on the mobile, it will fight according to its class. For instance if it has been set to CLASS_MAGES then it will cast spells from the mages guild file in battle. A fight program overwrites these default actions.

```
>fight_prog 30~
mpechoat $n You flinch at the stab of the Velvet Hands Leader.
mpechoaround $n $N flinches at the stab of the Velvet Hands Leader.
mpmadd $n currhp -10
~
|
```

This program is on the Velvet Hands Leader in Faerdale. It has a 30% chance of triggering each round of battle. It echos to the PC that the stab hurts them and then it deducts 10 hps from their current hps. This could kill them if their hps are low enough. This sort of program allows you to put in attacks that are not standard.

```
>fight_prog 20~
yell Guards! Guards!  Elves in the palace!
yell Vermin! Worse than bloody mice!
if rand(20)
  if quest(0, 2, $i) < 3
    mpmload 15125
    mpforce guard mpkill $r
    mpmadd $i quest 0 2 1
  endif
endif
~
|
```

Fight progs can be used to allow the mobile to call upon aid. This program is on Abrum in Zazesspur when he gets attacked in his castle. He calls upon his guards for aid. In this particular prog he does not load a guard every time he yells for the guards. The fight prog has a 20% chance of occuring each round, and within that there is a 20% chance of a guard being loaded. In order to stop too many guards being loaded, we have set quest bits on Abrum himself not the PC. $i refers to the mobile that has the program on it. You can set quest bits on mobiles just the same as PC's, and often it is a good economical way to use quest bits in an area. This one makes sure that Abrum cannot load more than 3 guards.

```
>fight_prog 70~
mpecho Blackstaff mutters an incarnation and heals his wounds.
mpmadd self currhp 40
if rand(20)
  cast fireball $r
else
  if rand(20)
    cast earthquake
    say I am the greatest of mages.  All shall quake at the sight of me!
  else
    if rand(20)
      cast 'acid blast' $r
      say Feel the strength of my spell $r
      mpechoat $r Ouch!
      mpechoaround $r $r winces in pain.
    else
      if rand(20)
        cast 'lightning bolt' $r
        mpecho Blackstaff calls upon the elements to aid him in battle.
      else
        cast 'flame strike' $r
        mpecho A candle flickers as Blackstaff draws upon the power of fire.
        mpechoat $r Ouch!  That hurt!
        mpechoaround $r $r whimpers as the flamestrike burns them.
      endif
    endif
  endif
endif
~
|
```

This program is one that allows Blackstaff to cast random spells in battle. First of all he "heals" himself. I could have him cast heal, BUT he would probably still be chanting when he went to cast his next spell. This means you cant give him two spells in the one round. He needs time to chant the spells.

The use of mpaffect or mpcast would work here, as there is no lag associated with these like there is for cast.

TIME PROGRAMS

Time programs are triggered by the time. This is game time. The game time works on a military style 24 hour clock.

```
>time_prog 21~
close gate
mpecho The Guard closes the gate for the evening so that the restless dead cannot escape.
~
```

This prog is on the guards outside the city of the dead in Waterdeep. He closes the gate at 9pm at night. Note that 9pm is 2100 hours in military time. We drop off the 00 though.

```
>time_prog 7~
if rand(50)
  if position($i) != 8
    mpecho A thief stealthily slinks off to sell his nights takings.
    mpgoto 99
  endif
endif
~
>time_prog 21~
mpregoto
~
|
```

This is TWO time progs here. This is the programs on the thieves in Waterdeep. MOST of the thieves disappear during the day in Waterdeep. The first prog happens at 7 am. It has an `if rand(50)` in it to make it only happen 50% of the time. It also checks to see if the thief is fighting at the time with the if position check. It then goes to room 99. Now this room is a mobile waiting room. If you need to send a mobile away for a short time send it to this room. A tip for the wise, never go there with your imm char on the test port at night - SPAM! There is hundreds of mobs in there at night just from Waterdeep alone.

The next prog happens at 9pm at night. The `mpregoto` sends the mobile back to the room it came from. On another mob you might put an echo saying they start work for the day. But the thief doesnt really want his presence known in the room so I have no echo.

If you want a program to trigger every hour, then put -1 into the time field.

```
>time_prog -1~
mpecho You hear bells marking the end of an hour.
~
|
```

ENTRY PROGRAMS

None of these in the game seem to work. A coder needs to look into this.

```
Code will go here.
```

DROP PROGRAMS

This program is an object program that is activated when the object is dropped. It can be activated a certain percentage of the time.

```
>drop_prog 100~
mpecho $O disintegrates as they touch the ground.
mppurge $o
~
|
```

This prog is on the plain equipment from the temple. When a player drops the equipment it echos to the room ($O is the object's short desc), and then it purges the object. There is of course a lot more possibility for such a prog. For instance you could make a ball bounce away, or a magical object explode and so on.

EXA PROGRAMS

Examine progs are activated when the PC examines or looks at an object.

```
>exa_prog 100~
mpechoat $n You take a look at the Newspaper with interest.
mpechoaround $n $n takes a look at the Newspaper with some interest.
mpforce $n help waterdeepnews
~
|
```

This program is on the Waterdeep News Board. Each time a PC looks at the board they get those echos and they are forced to read a help file. As you cannot put help files in your own area at this point time, you will need to consult with us before doing this kind of command.

```
>exa_prog 100~
mpoload 10310
put i10310 chest
mpechoat $n You spy a key in a corner of the chest.
~
|
```

This is the prog on the chest near the first door in the training temple. It loads up a key and puts the key in the chest and then echos that there is a key in the chest. Unfortunately the echo before this prog is that the chest is empty.

WEAR PROGRAMS

Wear programs activate when an item is worn or held by the PC.

```
>wear_prog 100~
if deity($n) != Corellon
  mpforce $n remove i1243
  mpechoat $n The boots zaps you and disappear.
  mpechoaround $n The boots zaps $n and disappear.
  mpmadd $n currhp -20
  mppurge i1243 $n
else
  mpechoat $n Your footsteps can no longer be heard.
endif
~
|
```

This prog is on the supplicate object of Corellons followers. When worn it checks to see if the deity is Corellon and if not it hurts the player and they junk. If they do follow corellon it echos to them.

BUY PROGRAMS

Buy progs work similar to give progs in their format. They only work for the PC buying an object from a mob, not selling them to a mob. They cannot be used to stop a PC from buying the object. If you wish to restrict a PC from buying from that mob then you will have to use an intercept program (a sample of which I will include below). An intercept on buy means that it works for EVERYTHING that the mobile sells but a buy program is for an individual object that the mobile sells.

```
>buy_prog i7931~
sayto $n This fey longsword was magically enhanced
sayto $n by a Priestess of Corellon.
sayto $n With the fey shortsword they make well
sayto $n balanced weapons for dual wielding.
~
|
```

The above prog activates when item vnum 7931 is bought from the mobile. In this case its a fey longsword. As you can see the buy prog is useful for telling PC's information about the object they just bought. Buy progs activate 100 percent of the time. If you want to reduce the chance of the mobile saying something then use an if rand in the prog.

```
>intercept_prog buy~
if guild($n) != Rangers
  sayto $n I am sorry, but I only cater to rangers.
else
  mpunintercept
endif
~
|
```

This program here restricts all who are not rangers from buying from this mobile. It activates when the PC types buy. It cannot be used to restrict them from buying one item but rather from shopping with that mobile over all.

LEAVE PROGRAMS

Leave programs that are in rooms are activated when a PC leaves the room. Anything in the prog will happen as the PC leaves so the commands will activate in the room they are leaving from. Leave programs on a mobile activate when the mobile leaves the room not the PC.

```
>leave_prog 100~
connect up
~
|
```

This very simple program is in Howling Peak in the room with the burned out wagon. Quite simple as the PC leaves that room it closes the direction up that was created when the PC looked at the wagon. This makes sure that anyone coming after the PC to the area will have to find the path for themselves. Now this connection wont close until all of the PC's party leaves up.

```
>leave_prog 40~
if rand(30)
  mpechoat $n $O jingles as you move.
endif
mpechoaround $n $n leaves with a jingle of bells.
~
|
```

This leave prog is on an object, a set of bells. A percentage of the time, when the PC leaves the room, they will do so jingling.

OTHER LESSONS

GUILDS IN AREAS

Guilds for the purpose of this page are the ones PC's join to give them new skills and develop the character further. This includes rangers, fighters, thieves, bards, etc and the various wizard schools of magic. Your area can have a branch of any of these guilds providing it is appropriate for the area. In recent times, we have been getting long term roleplayed high level characters to make a guild as part of their roleplay. The player of the character either designs or builds the area. These are areas in their own right, that hang off existing areas.

We are trying to avoid putting any more guilds in Waterdeep, in an effort decentralise the system. If you are interested in building a guild, ASK FIRST as with any other area. At this point we are not just considering the builder, but also the character behind the roleplay of the guild. We expect a joining quest in a guild, and also quests to learn one or two of the higher level spells/skills available to that guild. Equipment can be for sale in the guild, and trainers of SOME of the skills/spells/feats. No guild should be a one stop shop for all skills.

The forum for areas that need builders, lists some of the guilds we are wanting to see built right now.

GUILD AND TEMPLE STOREROOMS

First, before even doing any of this, make sure it is OK with the area administrators. There can only be ONE storeroom per guild. Clerics of each faith are considered a guild in the hard code. The storerooms will be placed in what is considered to be the main branch or temple of the guild/faith.

Members of the guild can use the donate command to donate items to the storeroom. If the PC is of hero level they do not have to be in the room to donate, they have the magical ability to donate from a distance.

A builder does not do anything more than make the room in the area for a donation room, the rest must be done by an area administrator in the game, editing the guild file settings. It is suggested that you place the room in a place where only guild members can easily get to it. It should be possible for determined characters to steal from it, but not easy. This means put protections in place that are not impossible to over come.

QUEST REWARDS

On Forgotten Kingdoms we have two main types of quests. Immortal run quests and area driven quests. Rewards for doing quests depend on the difficulty of the quest. We have no set guide lines or rules, but we ask that common sense be used. If we find that the rewards for an area driven quest are too high we will get the builder to change them.

A player can recieve more than just objects as a reward. Possible rewards include:

- **Magical Objects** Magical objects are objects that have applies added to them or a special object program that gives the object a magical aspect. Magical objects are not very common and should be hard to find. Potions and scrolls are reguarded as magical objects.
- **High Quality Objects** These include objects made of rare and expensive materials such as mithril and adamantite. We do not want objects made of these materials very common in the game so they are suitable reward for hard quests. Objects made from exotic materials should be from suitable quests. For instance a dwarven quest would yeild Mithril armour, and an elven quest elven chain and so forth. We want most of our mithril and elven chain items to come from the trades system, from players who work hard at the trades. It gives more value to the items. So these kinds of items should not be freely available in shops, but should be rare quest rewards.
- **Favor/Favour** If the quest is deity related favour can be given or taken away. It is important to use mpmadd to current favour rather than mpmset.
- **Glory Points** For most quests we would like to see glory points given. A simple two step quest will most likely not earn a quest point. The amount of quest points given depends on the difficulty of the quest. Players can then use accumulated quest points for special things like getting stat points at Aurora's. Generally quest points given range from 1 to 3 points depending on the difficulty of the quest.
- **Experience** Killing mobiles shouldn't be the only way to earn experience and quests should be a good source of experience, especially at the higher levels. You can either use mpmadd exp and give a specific amount or use mpreward to give exp that is calculated by hard code based on level and so on.
- **Gold** Our aim on Forgotten Kingdoms is to have gold mean something. So giving gold as a reward to a quest can be worthwhile. For instance escorting a VIP from one city to another will be a common way to give gold as a reward.
- **Alignment** You can alter an characters HIDDEN alignment. There are two fields that are visable to deities only. The lawful and the good field. This is what we check to see that a character is acting like its CHOSEN alignment. If a player does an evil quest then both their good and their lawful should be lowered. Only use mpmadd not mpmset with the lawful and good fields.
- **Skills** If the quest is teaching a trade skill, then you can reward the PC with a small amount of skill level in that skill. Some quests might teach a skill outside of a class and you can reward them accordingly. This kind of reward should be cleared with the Area admins first.

BUILDERS NOTES ON DWELLINGS

This lesson contains information needed for the making of mobs and objects specific for dwellings. Most builders will never need to use this information as most of the dwelling work is done by the builder admins.

Dwelling Mob Type Wages

Certain mob types in dwellings perform special functions. For instance a servant reduces the cost of repairs each month. In order for the code to know what type of mob is in the dwelling it looks at the wage rate. The wage rate of the mob needs to be set to that rate. The following is a list of mob types and the wage rates:

- Servants - 299 copper per month
- Cooks - 298 copper per month

Dwelling Special Mob Fields

Mobs that are for hire in dwellings have fields that must be set. All dwelling mobs must be unique as the code uses the line of 5 zeros after the mob stats line. These are defined here on after as Value0 to Value4 from left to right.

- Value0 == This is set by the game when the mob is hired to the treasury room of the player that has hired the mob.
- Value1 == This is the mobs wages each month in copper. Certain mob types like servants must have this value set to a predetermined amount in order for them to function correctly.
- Value2 == This is the vnum of their hire room. This is the room they will go back to if they have been dismissed or they quit for not being paid.

Dwellings use the 40000 to 41000 vnum block presently. Hire costs are set by progs on the mob itself.

Dwelling Book Shops

Bards will be able to purchase books from a publishing house at 25% of the value of the book. They can then place the book in their shop storeroom for sale and make around 25% profit. Each bard has their own publisher, most should be in Tym Waterdeep Limited on Book Street.

Bards are told they may only sell in their own shop, not to other shop-keepers. All sales of the book from the publisher are logged. If we find bards abusing this system the bard will no longer be able to sell their own books in their shop.

TRADES

Forgotten Kingdoms has a detailed and constantly expanding and evolving trades system. Area support is needed for the trades to work. Policy has made it that in order to learn a trade, a character must do a quest, rather than visit a trainer and use the train command like other skills and spells.

If you feel it is ICly fitting for your area to have a trades quest, ask for permission from the Area Administrators. The quest is coded just like any other quest. The difference is in how the trade skill is learned, using the mppractice command.

Refer the to Trades List for a listing of trades that FKMud offers. Some skills like brew, scribe, makewands, makestaves are treated the same way as trades for characters to learn them.

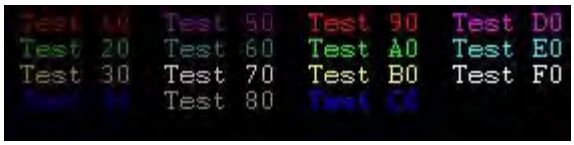
```
mppractice $n 5 mining
mppractice $n 5 mining
mppractice $n 5 mining
mplog TRADESKILLS: $n has been given 3 skill points in the mining skill.
```

The 5 in the above command is how high it will train up to. If the PC has a skill level over 5 then it will not teach them. The max skill level is 25. Only 1 point in the skill is given at a time, and in this instance, in order to give 3 points in the skill the command was repeated 3 times. The PC sees no echo for this. Make sure to put in a TRADESKILLS: log.

COLOURISING AN AREA

All mobiles, objects and rooms should be colourised. We want these items to follow a consistant standard, so that the game has a co-hesivive look and feel to it.

The syntax for applying colour to something as is `{00}`. The 00 is the colour number. It must be enclosed in the curley braces for the code to know that this is a colour code. The possible colours include (note that the graphic below does not show the braces):



The same information in the graphic above can be found in game by typing `help colour codes`.

- Colourising mobiles - Standards for colourising mobiles
- Colourising objects - Standards for colourising objects
- Colourising rooms - Standards for colourising rooms

When selecting the colour, you should be aware that the standard for most players is a blackground, not a white one. We work to the black standard. As a result `{00}` will not show up for most. This is an ideal code to use when you want to space out something with characters you do not want seen. It is often used for ascii art in the game.

PRESTIGE CLASSES

How we are going to handle prestige classes is still under discussion. The topic was opened to discussion to the who game in the game forums. You can read about it here. The game owners will then decide how they want to handle implementing this. Once done a lesson will be written for builders.

CHARACTER HOMETOWNS

When a character is created, the player chooses a hometown for the character. As a builder sometimes you might want to check for if a character is from a certain hometown. The following is a list of hometowns, and their recall vnums.

CREATION	30000
WATERDEEP	8030
ZHENTIL_KEEP	7557
ORC_VILLAGE	2819
MENZO	3174
MITHRIL_HALL	6705
GOLDEN_OAK	13664
BALABELL	14928
SHADOWDALE	1844
SILVERYMOON	4561
CALIMPORT	1024
TALL_TREES	2048
EVERMEET	4096
SKULLPORT	21062
WESTGATE	9492
CENTAURS_GROVE	13523

Please note that not all of these hometowns have been implemented in the game. Some of them are in planning.

NON KEY DOORS

There are several doors in the mud that cannot be open simply with a key. You might have to lift a bar, push a (hidden) button, turn a dial, say a keyword, knock on the door, or do some other action.

This can generally be done with a (more or less) simple intercept or speech program. But, if you want more precise echoes, it can get a bit trickier. Let's say you want to use an intercept turn program to unlock and open a door. The two rooms connected by this door are vnums A and B (A being south from B). You might use the following program in room A:

```
>intercept_prog turn~
mpecho As you turn the dial, the door opens.
mpat B mpecho The door to the south opens.
mpoload keyvnum
unlock north
open north
mpjunk ikeyvnum
~
```

Note: You could have an intercept_prog turn in room A and a speech_prog Open! in room B if you want the door to require different actions to be opened, depending on whether the character is in room A or room B. The two programs can be different.

This program is good enough to allow characters to open the door. But, if someone opens the door (by triggering the program) then someone else comes in room A and types 'turn', they will see the 'door opens' echo again, even though the door is already open. You might have the same problem if you want to enable characters to lock the door by using the same keyword ('turn' in the example).

The only way to solve this problem is to keep track of the state of the door: is it open or not. The most obvious way to do this is with quest bits... but, since quest bits can't be kept on rooms, we need to replace the room progs with mob programs, and thus, we need to have invisible mobs in each of the rooms.

Since the door can be either open or closed and locked, we only need quest bit 0,1. Let us assume that the door is initially closed and locked and that this value is quest bit 0,1 = 0. The programs could be something like:

```
>intercept_prog turn~
if quest(0,1,$n) == 0
  mpecho The door to the north opens.
  mpat B mpecho The door to the south opens.
```

```
  mpoload keyvnum
  unlock north
  open north
  mpjunk ikeyvnum
  mpmset self quest 0 1 1
else
  mpecho The door to the north closes.
  mpat B mpecho The door to the south closes.
  mpoload keyvnum
  close north
  lock north
  mpjunk ikeyvnum
  mpmset self quest 0 1 0
endif
~
```

Unfortunately, that does not solve all the problems yet. This program allows the code to remember if the door is open or closed, but we also need to make sure that the "state" of the door is the same in room A and in room B. That is, we need to make sure that, when someone opens the door from room A, the invisible mob in room B knows that the state of the door is now 'open' (and vice versa).

As far as I know, there are 2 ways to do that. The first one is perhaps simpler, but can be a problem if the programs are complex and long (since it will double the length of the programs). The second one is a bit more difficult, but it does not add much to the length of the programs.

The first solution consists in using the fact that, if you have 2 mobs with the same vnum (two copies of the same mob), when the quest bits are changed on one of the mobs, they are also changed on the other ones (and on any other mobs with the same vnum). We can thus make 1 mob and use 2 copies of this mob (one in room A and one in room B). As soon as one of them modifies its quest bits (when the state of the door is changed), the other one will automatically have its quest bits modified as well.

That means that both programs (program for room A and program for room B) must be placed on this mob, and that inroom checks must be added. For the example, I assume that the keyword to open/close the door is 'turn' for both rooms.

```
>intercept_prog turn~
if inroom(self) = A
  *same program as above*
else
  *same program as above*
  *but north becomes south*
  *and south becomes north*
endif
~
```

If the original program is already long, by using this solution (and thus doubling the length of the program), you might get a program that is too long (I'm not sure exactly what the limit on the number of lines in a program is, but there is a limit).

For the second solution, 2 different mobs are used and we need to modify the quest bits of both mobs each time the door is opened/closed. Now, there is an additional problem: invisible mobs are... invisible. And they can't see each other. So, if AMOB and BMOB are the vnums of the mobs in rooms A and B (respectively), we cannot simply use a command like 'mpat B mpmset mBMOB quest 0 1 1'.

There is a relatively simple solution to this though... even if mob BMOB cannot be seen, it can still intercept commands from any characters or mobs, even those it cannot see. The trick consists in having a command executed in room B, having mob BMOB intercept this command and react by changing its own quest bits.

So, if we add these intercept_progs on mob BMOB

```
>intercept_prog doorhasbeenopened~
mpmset self quest 0 1 1
~
>intercept_prog doorhasbeenclosed~
mpmset self quest 0 1 0
~
```

We can modify AMOB's program into

```
>intercept_prog turn~
if quest(0,1,$n) == 0
  mpecho The door to the north opens.
  mpat B mpecho The door to the south opens.
  mpat B doorhasbeenopened
  mpoload keyvnum
  unlock north
  open north
  mpjunk ikeyvnum
  mpmset self quest 0 1 1
else
  mpecho The door to the north closes.
  mpat B mpecho The door to the south closes.
  mpat B doorhasbeenclosed
```

```
    mpoload keyvnum
    close north
    lock north
    mpjunk ikeyvnum
    mpmset self quest 0 1 0
  endif
  ~
```

Basically, in this solution, we have mob AMOB trigger the intercept_prog of BMOB in order to have BMOB change its own quest bits.

Additional ways to handle doors is covered in the Buttons and Levers Lesson.

Months of the Game

The game has its own unique names for each of the months of the calendar year.

MONTH NAME	BIT VECTOR
Deepwinter	0
the Claw of Winter	1
the Sunsets	2
the Storms	3
the Melting	4
the Time of Flowers	5
Summertide	6
Highsun	7
the Fading	8
Leafall	9
the Rotting	10
the Drawing Down	11

Our players come from all parts of the globe, so it would be rude to equate a particular month in the game with a particular month of the year because for one part of the world, July is the middle of summer, and the other part July is the middle of winter. So below is a list that shows which 3 months go with which season. Above is the bit vector that matches the particular month, for use in the if month check.

(Winter)	Deepwinter	the Claw of Winter	the Sunsets
(Spring)	the Storms	the Melting	the Time of Flowers
(Summer)	Summertide	Highsun	the Fading
(Autumn/Fall)	Leafall	the Rotting	the Drawing Down

SECTOR RESTRICTIONS

What on earth did I intend for this lesson?

OLC COMMANDS

OLC COMMANDS

Main OLC commands

- `loadarea`: used to load your area, you will have to use it after a crash or a copyover
- `savearea`: remember to do that often enough! There is nothing worse than a crash and losing a heap of work.
- `instaroom`: puts the objects and mobs (and their inventory) of the room in the reset section of your area
- `rat`: allows you to execute a command in several rooms at once (you might want to warn the other builders online about the spam you are going to make though), e.g. : '`rat 2000 2050 redit flags indoors,`' will execute '`redit flags indoors`' in rooms 2000, 2001, 2002, ..., 2050.

Commands for rooms, mobs, and objects - Summary

List

- `rlist, mlist, and olist`: list the rooms, mobs, and objects (respectively) that have already been created in your area. You can also use vnums to get a partial list: `rlist 8030 8080`

Creation

- none for the rooms: simply use '`goto VNUM`' to create a new room VNUM, or create a connection to that room
- `mcreate VNUM keywords`: creates a new mob with the listed keywords
- `ocreate VNUM keywords`: creates a new object with the listed keywords

Examination

- `rstat`: gives you the stats of the current room
- `mstat keyword, ostat keyword`: allow you to get the stats of a mob or an object. For each mob, mVNUM is a keyword for mob VNUM and, for each object, iVNUM is a keyword for object VNUM. These keywords are not set immediately after the mob/object is created in OLC though, only after a copyover/crash.

Edit

- `redit`: for rooms (see below)
- `mset, oset`: for mobs and objects (see below)

Programs

- `rpedit, mpedit, opedit`: to add/remove/edit a program. Use them only if you are familiar with the program code.

Room editing

The command to edit rooms is `redit`. It's used in this way: `redit field value`. Here are the main 'field's you can set for a room in OLC.

- `redit name NAME`: allows you to set the name of the room. Do not use any colour code in the name of the room.
- `redit desc`: allows you to enter a writing buffer where you can set the room description (or edit it if the

room already has a description). This buffer uses the same system as the buffer for character descriptions. You can get a list of commands by typing /? while in the buffer. Room descriptions should be colourized.

- `redit ed KEYWORDS`: allows you to write an extra description for KEYWORDS in the room. Note that there are currently no way to edit or remove extra descriptions, so it might be easier to add them later in the area text file.
- `redit flags FLAGS`: allows you to set or remove room flags. If a flags is set twice, it will be removed (redit flags actually works as a toggle). You can set more than 1 flag with one command (e.g. `redit flags indoors noastral`). The command rat (see above) can be useful to set flags in several rooms at once. Some of the most used room flags include: nomob, noastral, indoors, dark, petshop.
- `redit sector SECTORNUMBER`: allows you to set the sector of the room. You can type '`help sector`' on the mud to get a list with all the sectors and their number. Using the word itself to set the sector type will not work, it has to be the bit number.
- `redit exit DIRECTION`: removes an exit (e.g., `redit exit east` will remove the east exit)
- `redit exit DIRECTION VNUM`: adds an exit in the specified direction, to room VNUM
- `redit bexit DIRECTION VNUM`: adds an exit in the specified direction, to room VNUM, and the corresponding exit in room VNUM (e.g., `redit bexit east 2000` creates an exit east to room 2000 and creates, in room 2000, an exit west to the current room).
- `redit exit DIRECTION VNUM FLAGS KEY KEYWORD`: adds an exit in the specified direction, to room VNUM, with a 'door'. This door will have the given KEYWORD, can be locked with object vnum KEY. If there is no key for this door, replace KEY with -1. The flags must be given in a numerical form. Some of the most used flags: 1 - is a door, 2 - is closed, 4 - is locked (they add up, so, for a closed door, the flag number is 1 + 2 = 3).
- `redit bexit DIRECTION VNUM FLAGS KEY KEYWORD`: same as above, but the connection is created in both rooms.
- `redit exflags DIRECTION`: lists the flags of the exit.
- `redit exflags DIRECTION FLAGS`: sets the flags of the exit. This is very useful for more complex flags (secret, hidden, pickproof, nopassdoor, ...) since the flags can be listed with their names here (and not the numerical value as in redit exit)

Mob editing

To edit mobs, use the command `mset MOB FIELD VALUE`. Its arguments are (a) a keyword for the mob that is being edited, (b) a field to be edited, and (c) a value for that field. A mob whose vnum is VNUM automatically has mVNUM as a keyword (m5358 for example). This automatical keyword is not set directly after the mob has been created in OLC though (but only after a copyover or crash).

- `mset MOB name KEYWORD`: sets the keywords of a mob. When created, a mob has the keywords listed in the mcreate command. This command can be used to change those keywords.
- `mset MOB short SHORTDESC`: sets the short description of a mob. You can colourize the short description. This short description is used for actions made by the mob, e.g., "SHORTDESC sits down and thinks deeply."
- `mset MOB long LONGDESC`: sets the long description of a mob. You can colourize the long description.

This long description is what people will see when they 'look' in a room where the mob is.

- `mset MOB sex SEXNUMBER`: sets the sex of the mob (0 for neutral, 1 for male, 2 for female). You must use the number and not the word in this case.
- `mset MOB race RACE`: sets the race of the mob. Type 'showraces' in the mud to see a list of all the available races.
- `mset MOB class CLASS`: sets the class of the mob (fighters, paladins, rangers, monster, thieves, bards, mages, illusionists, invokers, ...). Cleric mobs should have their class set to the name of the deity. For instance: mset MOB class chauntea will make the mob a priest/druid of chauntea. The mob will then cast spells that are of that faith. You should also set the deity field for priest mobs. See below.
- `mset MOB level LEVEL`: sets the level of the mob.
- `mset MOB deity DEITY`: sets the deity of the mob. You can use 'none' for no deity (that's the default value).
- `mset MOB flags FLAGS`: sets flags on the mobs. Some of the most used mob flags: sentinel, aggressive, healer (typing 'heal' in front of the mob will work), mobinvis.
- `mset MOB description`: allows you to enter a writing buffer where the mob's description can be written or edited.

The settings above are for simple mobs. If you use one of the commands described below, the mud will automatically convert the mob to a unique mob.

- `mset MOB affected AFFECT`: makes the mob affected by a spell or special affect (truesight, sanctuary, ...)
- `mset MOB str/int/wis/cha/lck/dex/con VALUE`: sets the strength/intelligence/wisdom/charisma/luck/dexterity/constitution of the mob.
- `mset MOB align ALIGN`: sets the mob's alignment. ALIGN should be a value amongst: 1000 (LG), 650 (NG), 450 (CG), 200 (LN), 0 (TN), -200 (CN), -450 (LE), -650 (NE), -1000 (CE).
- `mset MOB armor ARMOR`: sets the mob's default armor, e.g., 'mset MOB armor leather'. This is the armour that will be seen on the mob even when it is naked. What is termed the virtual armour.
- `mset MOB material ARMOR`: sets the mob's default armor material, e.g., '`mset MOB material bone`'. This is the armour that will be seen on the mob even when it is naked.
- `mset MOB speaks LANG`: sets the mob's known language(s), e.g., '`mset MOB speaks common dwarven`'.
- `mset MOB speaking LANG`: sets the mob's current language, e.g., '`mset MOB speaking dwarven`'.
- `mset MOB resistant/immune/susceptible RESIST`: sets the mob's resistances, immunities, susceptibilities, e.g., '`mset MOB resistant pierce`'.

Object editing

Objects are edited with the command `oset`. Its arguments are (a) a keyword for the object, (b) the field that is being edited, and (c) the value. The object vnum VNUM automatically has iVNUM (e.g., i937) as a keyword, but this keyword is not set immediately after the object is created in OLC (only after a copyover/crash).

- `oset OBJ name KEYWORDS`: allows you to set the keywords of an object. When the object is created, it has all the keywords listed in the ocreate command that created it.
- `oset OBJ short SHORTDESC`: sets the short description of the object. This is, for example, what can be seen when you look at someone wearing the item. The short description can be colourized.
- `oset OBJ long LONGDESC`: sets the long description of the object. This is what is seen when someone 'looks' in a room where the object is (on the ground). The long description can be colourized.
- `oset OBJ wear WEARLOC`: sets the wear locations for the objects. Several locations can be set in one command. If a location is set twice, it will be removed (oset wear works like a toggle). Some wear locations: take (this is needed if the object can be taken from the ground), hold, both, arms, legs, finger, head.
- `oset OBJ ed KEYWORD`: adds an extra-description for the keyword KEYWORD on the object. Currently, extra-descriptions cannot be removed or edited in OLC. It might be easier to add them later, in the area text file.
- `oset OBJ quality QUALITY`: sets the quality of the object (outstanding, superior, high, average, low, inferior, worthless)
- `oset OBJ condition CONDITION`: sets the condition of the object (perfect, superb, very good, good, usable, bad, very bad, awful, terrible)
- `oset OBJ material MATERIAL`: sets the object's material
- `oset OBJ flags FLAGS`: sets the flags of the object. More than one flag can be set in one command. If a flag is set twice, it is removed. Some object flags: magic, evil, antilaw, glow, hum.
- `oset OBJ affect AFFECT`: sets a magical affect on the object, e.g., 'oset OBJ affect dex -2' for a -2 dexterity affect when the object is worn.
- `oset OBJ rmaffect AFFECTNUMBER`: removes an affect from an object, e.g., 'oset OBJ rmaffect 1' removes the first affect. Ostat the object to see a list of its affects.
- `oset OBJ type TYPE`: sets the type of the object (light, scroll, potion, weapon, projectil, armor, food, fountain, container, sheath, quiver, drinkcon, ...)

Some information about objects are stored in 5 special values (called value0, ..., value4, see the Builders' Lesson Pages). These values can be set with `oset OBJ value0 VALUE`, ..., `oset OBJ value4 VALUE`.

The exact value is sometimes hard to find (e.g., what is the value for a short sword?). They can be found in 2 ways: (a) look them up on the Builders' Lesson page or in the fkbit.lst file that comes with the area check, (b) find a similar non-magical object (a short sword in this case) in Waterdeep and ostat it. If you do not want to do that, you might prefer to set these values later in the area text file, where you will be able to use the flags (WEAPON_TYPE_SHORT_SWORD, ...)

Here is an (incomplete) list of the values that can be set:

- weapon: value3 - weapon type
- armor: value3 - armor type
- light: value2 - hours of light left
- scroll/potion: value0 - level; value1, value2, value3 - spell numbers
- wand/staff: value0 - level; value1 - max number of charges; value2 - current number of charges; value3 - spell number

- container/quiver/sheath: value0 - weight capacity; value1 - flags (1 for closeable, 2 for pickproof, 4 for closed, 8 for locked, these values add up); value2 - key vnum
- drinkcon: value0 - capacity; value1 - current quantity; value2 - liquid number
- food: value0 - hours of food
- fountain: value1 - number of drinks, value2 - type of liquid

Shopkeepers and repairers

To create a shopkeeper (not a pet shopkeeper), create the mob and give him all the items he is supposed to sell and use instaroom in the room. Then use the command makeshop VNUM, where VNUM is the vnum of the shopkeeper.

You can set some values about the shopkeeper with the command `shopset`:

- `shopset VNUM open/close/buy/sell VALUE`: sets the opening hour, the closing hour, the buy value, and the sell value of the shopkeeper VNUM.

To create a repairer, create the mob then use the command makerepair VNUM, where VNUM is the vnum of the shopkeeper.

- `repairset VNUM open/close VALUE`: sets the opening hour and the closing hour.

Resets

Set up the room as you would like it to look in the game. You can create mobs or invoke existing ones from your area. Use `minvoke mVNUM` to load your mobs. Use `oinvoke oVNUM` to load up objects. You can give objects to mobs and `force` them to wear the objects. If they are to sell the objects then they must have them in inventory. Mobs will not sell what they are wearing.

- `instaroom`: Use this command once you have set up the room as you like it. It sets the resets for the room only, not the whole area.

General Area Commands

You can set some fields of your area in general using the `aset` command.

In general the syntax is `aset areaname.are field argument`.

Fields that can be set are (note some have been removed as they are not suitable for builders to set):
`author resetmsg resetfreq flags humidity climate mine wood judge courtroom dungeon laws temp`