

Vehicles in CWG Buddha and Sun-Tzu Releases

Ken Ray

<kenr86@hotmail.com>

October 2003

Abstract

Version, Copyright and License Information

The code described in this document is based on the Circle “cwg-buddha” release, dated August 09, 2003. Refer to that for the various software levels and versions. The “cwg” versions of the CircleMUD code have been put together by Mark Garringer, zizazat@hotmail.com.

Enhancements to that code documented here are available under the same terms as the standard CircleMUD license, as documented in the CircleMUD code. The three basic requirements are:

- 1) You must not use CircleMUD to make money or be compensated in any way.
- 2) You must give the authors credit for their work.
- 3) You must comply with the DikuMUD license.

This documentation has been provided to help CircleMUD administrators, builders and coders better understand how to use the Vehicle and Window enhancements in the cwg releases, and it is the intent of the author that this receive widespread distribution through the CircleMUD community. While all care has been taken in the design and development of the code enhancements, and in the writing of this documentation, no warranty expressed or implied is associated with either the code described herein or this documentation. You use it entirely at your own risk.

This is Version 1.1 of the documentation, dated October 22 2003.

This documentation is Copyright Kenneth J. Ray, October 2003.

Contents

VEHICLES IN CWG BUDDHA AND SUN-TZU RELEASES	I
ABSTRACT	1
VERSION, COPYRIGHT AND LICENSE INFORMATION	1
1 STRUCTURE OF A VEHICLE.....	1
1.1 WHAT IS A VEHICLE?	1
1.2 DESIGNING OUR VEHICLE	1
1.3 CREATING THE VEHICLE	2
1.3.1 <i>What is in the Code?</i>	2
1.3.2 <i>Creating the Base Vehicle Object</i>	3
1.3.3 <i>Creating the Vehicle Interior</i>	5
1.3.4 <i>Help! Let Me Out! Creating a Vehicle Hatch</i>	6
1.3.5 <i>Home James, And Don't Spare The Horses – The Controls</i>	8
1.3.6 <i>Sit Back And Enjoy The View – Add A Window</i>	10
1.4 MAKING EVERYTHING LOAD.....	12
2 ADVANCED VEHICLE FEATURES.....	13
2.1 ENTERING AND LEAVING VEHICLES	13
2.2 LOOKING INSIDE VEHICLES	14
2.3 PUTTING OBJECTS INSIDE A VEHICLE	15
2.4 VEHICLE CONTROLS AND DRIVING	15
2.4.1 <i>Defining Rooms as “Vehicle Accessible”</i>	15
2.4.2 <i>Checks Made When Driving</i>	16
2.4.3 <i>Specifying a Direction to Drive</i>	18
2.4.4 <i>Driving Into Other Vehicles</i>	19
2.4.5 <i>Driving On and Off The Ferry</i>	21
2.5 CONTROLLING ACCESS TO VEHICLES	23
2.6 SHORTCOMINGS WITH THE VEHICLES IMPLEMENTATION	23
3 MORE WINDOWS OPTIONS.....	25
3.1 SIMPLE WINDOWS	25
3.2 SPECIFYING THE WINDOW'S TARGET ROOM	28
3.3 WINDOWS THAT DON'T LOOK LIKE WINDOWS.....	31
3.3.1 <i>Repairing the Vandalized Roman Telescope</i>	31
3.3.2 <i>Portable Objects – A Crystal Ball</i>	32
3.4 WINDOWS THAT CLOSE AND LOCK	34
4 WORLD FILES CODE DETAILS.....	37
4.1 THE COMPLETE COACH	37
4.2 A VEHICLE FOR OTHER VEHICLES – THE RIVER FERRY	39
4.3 THE BASIC WINDOW	40
4.4 A WINDOW TO A DISCONNECTED ROOM.....	41
4.5 THE ROMAN TELESCOPE	41
4.6 THE CRYSTAL BALL	42
5 HELP FILE CHANGES.....	43
5.1 LOOK.....	43
5.2 ENTER	44
5.3 LEAVE	44
5.4 DRIVE.....	44
5.5 VEHICLES	45

1 STRUCTURE OF A VEHICLE

1.1 What is a Vehicle?

A vehicle is a complex set of objects and rooms that allows people to move around the mud in ways other than walking. When combined with dg_scripts, these can be very powerful.

A vehicle consists of a number of objects and rooms:

- i. The vehicle object. This defines the overall vehicle structure, and specifies the room that characters will go to if they enter the vehicle.
- ii. One or more vehicle rooms. These rooms comprise an isolated pocket within the zone, which can only be reached by entering the vehicle (although immortals can directly teleport to them).
- iii. A hatch object. This is the only way to leave the vehicle, and would normally be loaded into one of the vehicle rooms. A vehicle can have more than one hatch.
- iv. A control object. This is used to drive the vehicle, and again, for it to be useful needs to be loaded into one of the vehicle rooms.
- v. A window object. This allows people within the vehicle to see outside of the vehicle.

In order to use the vehicle, a new room flag (vehicle) has been defined, along with the “drive” command. Using the vehicle control object, players can issue drive commands to move the vehicle from room to room, provided that room allows vehicles by being flagged with the vehicle flag.

This document will take you through designing a vehicle, and showing where in the CircleMUD code modifications have been made to support this.

1.2 Designing our Vehicle

The first stage in this process is to work out what it is we want the vehicle to be like. In keeping with the theme of my mud, which is a traditional medieval style, I think a coach pulled by a team of horses, would be suitable. The coach will have two rooms, the interior seating area, and the rooftop driver’s position. The control (which we will call the reins) will be located in the driver’s seat. Hatches and windows will be located as required.



Of course, other (non-vehicle) objects can be included here as well, such as a coachman’s whip, bags and luggage items, and so forth. The only limit is your imagination.

Now that we have a rough idea of what form our vehicle should take, we will list the objects and rooms in a table. You will need to check your actual zones, using rlist and olist, to determine what room and object vnums are available.

Vehicle Objects			
Type	Vnum	Description	Location
Vehicle	2870	Coach	2860
Hatch	2871	Coach door	2860
Control	2872	Reins	2861
Window	2873	Coach window	2860
Window	2874	No Description	2861

Vehicle Rooms			
Name	Vnum	Purpose	Notes
Coach	2860	Interior of Coach	Entry / exit room
Driver's Seat	2861	Where the vehicle controls are	Up from entry

1.3 Creating The Vehicle

We will now go through each of the steps to create the vehicle, and make it able to be entered, manipulated and driven by player characters.

1.3.1 What is in the Code?

Before we start to define the various objects, what changes have been made to the CircleMUD code to allow these new object types to be defined? Firstly, in structs.h, we define the new object types:

```
#define ITEM_FOUNTAIN 23 /* Item is a fountain */
#define ITEM_VEHICLE 24 /* Item is a vehicle */
#define ITEM_HATCH 25 /* Item is a vehicle hatch */
#define ITEM_WINDOW 26 /* Item is a vehicle window */
#define ITEM_CONTROL 27 /* Item is a vehicle control */
#define ITEM_PORTAL 28 /* Item is a portal */
```

We also add the new room flag so vehicles can drive places:

```
#define ROOM_VEHICLE 16 /* Requires a vehicle to pass */
```

We add descriptive entries to strings in constant.c too:

```
/* ROOM_x */
const char *room_bits[] = {
    "DARK",
    "DEATH",
    ..<lines snipped>
    "OLC",
    "*", /* BFS MARK */
    "VEHICLE",
    "\n"
};
```

```

/* ITEM_x (ordinal object types) */
const char *item_types[] = {
    "UNDEFINED",
    "LIGHT",
    "SCROLL",
    "WAND",
    <lines snipped>
    "FOUNTAIN",
    "VEHICLE",
    "HATCH",
    "WINDOW",
    "CONTROL",
    "PORTAL",
    "\n"
};

```

We also have to allow for this in oedit.c as well. The routine oedit_disp_val1_menu displays the appropriate prompt message when entering object values:

```

case ITEM_VEHICLE:
    write_to_output(d,
        "Enter room vnum of vehicle interior : ");
    break;
case ITEM_HATCH:
    write_to_output(d,
        "Enter vnum of the vehicle this hatch belongs to : ");
    break;
case ITEM_WINDOW:
    write_to_output(d,
        "Enter vnum of the vehicle this window belongs to : ");
    break;
case ITEM_CONTROL:
    write_to_output(d,
        "Enter vnum of the vehicle these controls belong to : ");
    break;

```

As well, in oasis.h the number of room flags and item types is adjusted to include these new types.

Of course, other changes have been made to the code to allow you to enter, drive and leave the vehicle. This will be discussed later in this document.

1.3.2 Creating the Base Vehicle Object

The first thing to do is to create the vehicle object. This will tie everything together. Using oedit, enter the various values for the vehicle.

```

-- Item number : [2870]
1) Namelist : coach
2) S-Desc   : a weather-beaten coach
3) L-Desc   :-
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
4) A-Desc   :-
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy

```

```
part of the seat to sit on. This vehicle has certainly seen
better days.
```

```
5) Type          : VEHICLE
6) Extra flags   : NOBITS
7) Wear flags    : NOBITS
8) Weight        : 0
9) Cost          : 0
A) Cost/Day      : 0
B) Timer         : 0
C) Values        : 2860 0 -1 -1
D) Applies menu
E) Extra descriptions menu
M) Min Level     : 0
P) Perm Affects : NOBITS
S) Script        : Not Set.
Q) Quit
Enter choice :
```

You can also add some extra descriptions to the object. This is probably a good idea, because most players will try “look at coach”, and you want something more inspiring than the default “You see nothing special..” message. Let’s load the coach and see what we find.

```
500H 100M 82V > look
Trade Road
    The traffic on this road seems light at best, becoming a bit
    thicker to the east. Westwards, Trade Road continues towards
    the intersection with Clay Street. A large open-air building
    is to the north, and southwards is a large open lot; a
    fairgrounds, closed for the time being.
[ Exits: n e w ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.

500H 100M 82V > look at coach
The coach has seen better days. While it looks functional, the
paint on the sides has long faded, and several (hopefully
unimportant) parts are held on with lashings of rope. It looks
like about six people could travel inside, and several more on
the top with the driver. Two bored horses are hitched to the
front.

500H 100M 82V > look at horses
You wonder how these old nags could pull this coach. They
appear to be refugees from the knacker's yard.
```

Some things to note about defining the vehicle object:

- i. We set an “action description” (a-desc). This will be displayed to a character who enters the vehicle
- ii. The first object value field is set to the vnum of the room that will be the vehicle interior. If there are going to be more than one room within the vehicle, this will be the vnum of the room that players will go to when they enter the vehicle.

- iii. Other object fields include whether the vehicle has certain “container” characteristics, such as being able to be opened, closed, locked and unlocked. As well, you can indicate whether the vehicle is “transparent”, that is, whether can you look inside it from outside. These advanced features are discussed later in this document, but they follow the same conventions as for other container-like objects.
- iv. It is probably a good idea to clear the “take” wear flag, otherwise some person will try to take the coach. We haven’t bothered to set an item weight, rent or value, since they really don’t apply to items that can’t be taken, bought or sold.

1.3.3 Creating the Vehicle Interior

Now we need to create the rooms inside the vehicle. These will form an isolated “pocket” within the zone, with no direct connections to the outside world.

```

-- Room number : [2860]          Room zone: [28]
1) Name       : Inside An Old Coach
2) Description :
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
3) Room flags  : INDOORS NO_TRACK
4) Sector type : Inside
5) Exit north  : [ -1], B) Exit northwest : [ -1]
6) Exit east   : [ -1], C) Exit northeast : [ -1]
7) Exit south  : [ -1], D) Exit southeast : [ -1]
8) Exit west   : [ -1], E) Exit southwest : [ -1]
9) Exit up     : [ 2861], F) Exit in       : [ -1]
A) Exit down   : [ -1], G) Exit out       : [ -1]
H) Extra descriptions menu
S) Script      : Not Set.
Q) Quit
Enter choice :
-- Room number : [2861]          Room zone: [28]
1) Name       : The Driver's Seat
2) Description :
    From up here at the front of the coach, the driver would
    normally have control of the horses' reins and the coach brake.
    Strange, there doesn't seem to be a driver.
3) Room flags  : INDOORS NO_TRACK
4) Sector type : City
5) Exit north  : [ -1], B) Exit northwest : [ -1]
6) Exit east   : [ -1], C) Exit northeast : [ -1]
7) Exit south  : [ -1], D) Exit southeast : [ -1]
8) Exit west   : [ -1], E) Exit southwest : [ -1]
9) Exit up     : [ -1], F) Exit in       : [ -1]
A) Exit down   : [ 2860], G) Exit out     : [ -1]
H) Extra descriptions menu
S) Script      : Not Set.
Q) Quit
Enter choice :

```

Note how the rooms have been set as INDOORS. This allows us to make proper use of any window objects that we may place within the vehicle. Don’t confuse the “Inside” sector type (menu item 4) with the “INDOORS” room flag. The sector type is used to

determine the movement points used when entering that sector. The INDOORS room flag is used to check certain actions, such as “look outside”

Now, we have the two rooms. Let’s see what happens when we enter the coach.

```
500H 100M 82V > enter coach
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.

Inside An Old Coach
    The seats are stained, the floor is covered with dirt and
who knows what else. You are certainly traveling third class
in this coach.
[ Exits: u ]

500H 100M 82V > u
The Driver's Seat
    From up here at the front of the coach, the driver would
normally have control of the horses' reins and the coach brake.
Strange, there doesn't seem to be a driver.
[ Exits: d ]

500H 100M 82V > d
Inside An Old Coach
    The seats are stained, the floor is covered with dirt and
who knows what else. You are certainly traveling third class
in this coach.
[ Exits: u ]

500H 100M 82V >
```

When we entered the coach, you can see we were shown the “a-desc” message from the main vehicle object. We can move between the two rooms that comprise the coach’s interior. However, there is no way for us to leave the coach.

We from outside the coach, we can also look inside it.

```
500H 100M 82V > look in coach
You look inside and see:
[ 2860] Inside An Old Coach [ INDOORS NO_TRACK ]
    The seats are stained, the floor is covered with dirt and
who knows what else. You are certainly traveling third class
in this coach.
[ Exits: u ]
[2873] A small window provides a view of the outside world.
[2871] A door provides a way out of the coach.
```

There are some situations where you will be unable to look inside the vehicle, but these will be covered later. This will list the name of the room that is the vehicle’s entry point.

1.3.4 Help! Let Me Out! Creating a Vehicle Hatch

Now that we can enter our vehicle, and move around inside it, we need a way to get out. This is where we use the hatch object. Note that we will associate the hatch with the

vehicle's vnum using the first object value flag. We will also have to load the hatch in the room inside the vehicle that we want the exit to be in.

```
-- Item number : [2871]
1) Namelist : door
2) S-Desc : a door
3) L-Desc :-
A door provides a way out of the coach.
4) A-Desc :-
Phew! You are glad to be out of that stinking coach.
5) Type : HATCH
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight : 0
9) Cost : 0
A) Cost/Day : 0
B) Timer : 0
C) Values : 2870 0 -1 0
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script : Not Set.
Q) Quit
Enter choice :e
Extra desc menu
1) Keyword: door
2) Description:
Thank goodness! At least you can get out of this vile coach
when you want to. The door's lock has vanished long ago, now a
loop of twine holds the door closed.

3) Goto next description: <Not set>

0) Quit
Enter choice :
```

Again, the hatch can also take on some container attributes.

Once we load the hatch into one of the interior rooms, we can see our way out:

```
500H 100M 82V > enter coach
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.

Inside An Old Coach
The seats are stained, the floor is covered with dirt and
who knows what else. You are certainly traveling third class
in this coach.
[ Exits: u ]
A door provides a way out of the coach.

500H 100M 82V > look at door
Thank goodness! At least you can get out of this vile coach
when you want to. The door's lock has vanished long ago, now a
loop of twine holds the door closed.
```

So we now have a door to get out of the coach.

```
500H 100M 82V > leave
Phew! You are glad to be out of that stinking coach.

Trade Road
  The traffic on this road seems light at best, becoming a bit
  thicker to the east. Westwards, Trade Road continues towards
  the intersection with Clay Street. A large open-air building
  is to the north, and southwards is a large open lot; a
  fairgrounds, closed for the time being.
[ Exits: n e w ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
```

By the way, what do other people see when we enter and leave the vehicle?

```
36H 118M 74V >
Maclir enters a weather-beaten coach.

36H 118M 79V >
Maclir leaves a weather-beaten coach.

36H 118M 74V >
```

Now, if we are in the coach, what do we see?

```
36H 118M 81V >
Maclir enters a weather-beaten coach.

36H 118M 81V >
Maclir leaves a weather-beaten coach.

36H 118M 74V >
```

1.3.5 Home James, And Don't Spare The Horses – The Controls

So far, so good. We have a vehicle that we can enter, move around inside it, and leave. All the appropriate messages get sent as we interact with the vehicle. But – we aren't able to travel anywhere in it. This requires a vehicle control object. Again, the first object value flag associates the control object with the vehicle, and we need to load the control somewhere within the vehicle.

```
-- Item number : [2872]
1) Namelist : reins
2) S-Desc   : a set of leather reins
3) L-Desc   :-
A set of leather reins is harnessed to the horses.
4) A-Desc   :-
<not set>
5) Type     : CONTROL
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day  : 0
B) Timer    : 0
C) Values   : 2870 0 0 0
D) Applies menu
```

```

E) Extra descriptions menu
M) Min Level      : 0
P) Perm Affects: NOBITS
S) Script        : Not Set.
Q) Quit
Enter choice :e
Extra desc menu
1) Keyword: reins
2) Description:
These reins are harnessed to the horses, by pulling on these to
exert pressure, and no doubt with a lot of shouting, you could
direct the horses to pull the coach somewhere.

3) Goto next description: <Not set>

0) Quit
Enter choice : 0

```

So, we load the control object to a suitable room within our vehicle.

```

500H 100M 82V > look
The Driver's Seat
    From up here at the front of the coach, the driver would
normally have control of the horses' reins and the coach brake.
Strange, there doesn't seem to be a driver.
[ Exits: d ]
A set of leather reins is harnessed to the horses.

500H 100M 82V > look at reins
These reins are harnessed to the horses, by pulling on these to
exert pressure, and no doubt with a lot of shouting, you could
direct the horses to pull the coach somewhere.

500H 100M 82V >

```

Using the “drive” command, we can now move the coach.

```

500H 100M 82V > drive e
Trade Road
    Trade Road seems to widen here slightly, allowing more
traffic to travel through the area. You spot near-identical
gatehouses to the east and west marking the boundaries of Trade
Road, and this neighborhood. To the south is a large
fairgrounds area, closed for the time being. A building to the
north seems to be locked up tight, with a large, 'FOR RENT'
sign out front...
[ Exits: e w ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
A Citizen of Midgaard passes you without a glance.

500H 100M 82V > drive e
Trade Road
    Several people are walking along the road here, but they
seem to be avoiding you with eyes and movements for the most
part. The citizenry seem quiet; depressed or merely sedentary
you would guess. Trade Road leads far to the west from here,
and there is a largish gatehouse not far to the east. A north-
south running road labeled 'Staid Avenue' intersects here,

```

```
taking much of the traffic away from the main road to the
south.
[ Exits: n e s w ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
A cityguard stands here.

500H 100M 82V >
```

Notice how the room description you see is not that of the Driver's Seat, but the room and its contents, which the vehicle has now moved into. This is the key to how vehicles move around; the room(s) inside the vehicles, and their contents, remains fixed. However, it is the vehicle object itself that is moved from one external room to another when the vehicle is driven. This is covered later in Section 2.4, "Vehicle Controls and Driving" on page 15.

While you are driving around you may find a use for the "brief" command to reduce the amount of information displayed.

1.3.6 Sit Back And Enjoy The View – Add A Window

What you have probably noticed by now is that you cannot easily see where the vehicle is at any time. The normal "look" command will always show you the vehicle room you are in. If you drive the vehicle to another room, as part of the driving action the driver sees the new room. But none of the other passengers in the vehicle are aware it has even moved.

This problem is partially solved by the use of the window object. Again, in the first object value field, we associate the window to the vehicle object, and once created, we have to load the window into the vehicle room that you want the passengers to be able to see out of.

```
500H 100M 82V > oedit 2873
-- Item number : [2873]
1) Namelist : window
2) S-Desc   : a window
3) L-Desc   :-
A small window provides a view of the outside world.
4) A-Desc   :-
<not set>
5) Type     : WINDOW
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day : 0
B) Timer    : 0
C) Values   : 2870 0 -1 -1
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script    : Not Set.
Q) Quit
Enter choice :
```

Now if we load the window object into our coach interior, and try to look outside.

```
500H 100M 82V > look
Inside An Old Coach
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
[ Exits: u ]
A small window provides a view of the outside world.
A door provides a way out of the coach.

500H 100M 82V > look out window
Trade Road
    The traffic on this road seems light at best, becoming a bit
    thicker to the east. Westwards, Trade Road continues towards
    the intersection with Clay Street. A large open-air building
    is to the north, and southwards is a large open lot; a
    fairgrounds, closed for the time being.
[ Exits: n e w ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.

500H 100M 82V >
```

So we now have our window, and we can look outside. However, what about places on the vehicle where you should automatically be able to see outside? If we don't give the window object a long description, then it will not get listed as part of the room contents, even though it is still in the room, and can still be used.

```
500H 100M 82V > oedit 2874
-- Item number : [2874]
1) Namelist : window
2) S-Desc   : window
3) L-Desc   :-
Undefined
4) A-Desc   :-
<not set>
5) Type     : WINDOW
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day : 0
B) Timer    : 0
C) Values   : 2870 0 -1 -1
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script    : Not Set.
Q) Quit
Enter choice :
```

So now when we load the object in the driver's seat room, even though no object shows up on the room listing, we can still interact with it. These so-called "invisible objects" are useful where we want an object that players can interact with, but doesn't show in the room listing. For safety's sake, these should have their take flag cleared.

```

500H 100M 82V > look
The Driver's Seat
    From up here at the front of the coach, the driver would
normally have control of the horses' reins and the coach brake.
Strange, there doesn't seem to be a driver.
[ Exits: d ]
A set of leather reins is harnessed to the horses.

500H 100M 82V > look out
Trade Road
    The traffic on this road seems light at best, becoming a bit
thicker to the east. Westwards, Trade Road continues towards
the intersection with Clay Street. A large open-air building
is to the north, and southwards is a large open lot; a
fairgrounds, closed for the time being.
[ Exits: n e w ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.

500H 100M 82V >

```

So, there is our basic vehicle. We can enter and leave it, look out of it, and drive it around.

1.4 Making Everything Load

You can either use zedit or manually edit the zone file to make sure the objects are loaded.

```

* Coach
R 0 2815 2870 0          Coach
O 0 2870 1 2815 0
R 0 2860 2871 0          Hatch
O 0 2871 1 2860 0
R 0 2861 2872 0          Control
O 0 2872 1 2861 0
R 0 2860 2873 0          Window
O 0 2873 1 2860 0
R 0 2861 2874 0          Window
O 0 2874 1 2861 0

```

2 Advanced Vehicle Features

2.1 Entering and Leaving Vehicles

The vehicle object, which can be considered to also be the way in to the vehicle rooms, and the vehicle hatch (which is the way out of the vehicle rooms) can also be assigned container-like attributes and behavior. In particular, these can be made openable and lockable, and have a key assigned to them.

These attributes are set in the second and third object value positions:

```
Enter choice : c
Enter room vnum of vehicle interior : 2860
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: NOBITS
Enter flag, 0 to quit : 1
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: CLOSEABLE
Enter flag, 0 to quit : 3
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: CLOSEABLE CLOSED
Enter flag, 0 to quit : 0
Vnum of key to unlock vehicle (-1 for no key) : -1
What is the vehicle's appearance (-1 for transparent) : -1
```

So we have to open the coach before we can enter it.

```
500H 100M 82V > enter coach
But it's closed!

500H 100M 82V > open coach
Okay.

500H 100M 82V > enter coach
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.

[ 2860] Inside An Old Coach [ INDOORS NO_TRACK ]
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
[ Exits: u ]
[2873] A small window provides a view of the outside world.
[2871] A door provides a way out of the coach.
```

Likewise, if the door were locked, we would get the standard message about locked doors or containers. So, assuming we had modified the coach to make it lockable, and we had the correct key, this is what we would see:

```
500H 100M 82V > open coach
It seems to be locked.

500H 100M 82V > unlock coach
*Click*

500H 100M 82V > open coach
Okay.
```

Exactly the same process applies to hatches.

2.2 Looking Inside Vehicles

A vehicle object can be defined as “transparent”, that is, if the vehicle object is not closed, from outside you can see inside the vehicle.

```
500H 100M 82V > look in coach
You look inside and see:
[ 2860] Inside An Old Coach [ INDOORS NO_TRACK ]
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
[ Exits: u ]
[2873] A small window provides a view of the outside world.
[2871] A door provides a way out of the coach.
```

However, if the transparency attribute (object value four) is set to 0, then you cannot see inside it unless you actually enter the vehicle.

```
500H 100M 82V > oedit 2870
-- Item number : [2870]
1) Namelist : coach
2) S-Desc   : a weather-beaten coach
3) L-Desc   :-
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
4) A-Desc   :-
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.
5) Type     : VEHICLE
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day : 0
B) Timer    : 0
C) Values   : 2860 0 -1 0
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script    : Not Set.
```

```
Q) Quit
Enter choice : q
500H 100M 82V > look in coach
You cannot see inside that.
```

As well, even if the vehicle is “transparent”, but it is room the vehicle object leads to is dark, and the player cannot see in the dark, they get the “It is pitch black...” message, just as if they were in a dark room.

2.3 Putting Objects Inside a Vehicle

Well, if a vehicle can have some container-like properties, can we place things inside it, and then take them out?

```
500H 100M 82V > put glass in coach
You put a glass sphere in a weather-beaten coach.

500H 100M 82V > look in coach
You look inside and see:
[ 2860] Inside An Old Coach [ INDOORS NO_TRACK ]
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
[ Exits: u ]
[3133] This glass sphere seems to contain a cloudy vapour. ..It
has a soft glowing aura!
[2873] A small window provides a view of the outside world.
[2871] A door provides a way out of the coach.

500H 100M 82V > take glass from coach
You will need to enter it first.
```

So, we can put objects in a vehicle from outside the vehicle, assuming the vehicle object – the door, so to speak, is open. However, we cannot take objects from within the vehicle unless we are within the vehicle.

2.4 Vehicle Controls and Driving

The “drive” command, which has the synonym “pilot”, is used to move the vehicle around. If the character simply enters a directional command, then the assumption is they are trying to move around within the vehicle.

2.4.1 Defining Rooms as “Vehicle Accessible”

Not all locations are suitable for a vehicle. In fact, a vehicle can only be driven into rooms that you specifically allow it to. The new room flag “VEHICLE” has been introduced to specify that a vehicle (any vehicle) can enter this location. This has been added as one of the room flags options in the Oasis command `redit`.

Remember that flagging a room with VEHICLE means any vehicle can enter that room. If you have several vehicles, you may need to have buffer rooms not flagged as vehicle accessible, so you don’t end up with someone driving a coach through the water, or sailing a submarine up Main Street.

If a player tries to drive a vehicle into a room that has not been flagged as accessible to vehicles, they get a message saying that cannot go that way.

```
500H 100M 82V > look
[ 2861] The Driver's Seat [ INDOORS NO_TRACK ]
    From up here at the front of the coach, the driver would
normally have
control of the horses' reins and the coach brake.  Strange,
there doesn't seem
to be a driver.
[ Exits: d ]
[2872] A set of leather reins is harnessed to the horses.

500H 100M 82V > look out
You look outside and see:
[ 2815] Trade Road [ VEHICLE ]
    The traffic on this road seems light at best, becoming
a bit thicker to the east.  Westwards, Trade Road continues
towards the intersection with Clay Street.  A large open-air
building is to the north, and southwards is a large open lot;
a fairgrounds, closed for the time being.
[ Exits: n e w in ]
[2870] An old, weather-beaten coach, pulled by two horses,
waits for some passengers.

500H 100M 82V > drive w
[ 2814] Trade Road [ VEHICLE ]
    There seems to be more people and activity to the east,
where there is a large building.  Westward, you see the
intersection of Trade Road and Clay Street, which runs north-
south.
[ Exits: e w ]
[2870] An old, weather-beaten coach, pulled by two horses,
waits for some passengers.

500H 100M 82V > drive n
Alas, you cannot go that way...
```

2.4.2 Checks Made When Driving

The routine `do_drive` (in `vehicles.c`) controls most of the logic involved when a player enters a “drive” command. The first few checks are basic sanity checks to see if the character is able to drive the vehicle:

- What is the character’s position?
- Are they blind?
- Can they see in this room?

Then we see if we can use the controls. This means:

- Are there any controls here?
- Can they use the controls?
- Can we find the vehicle that belongs to those controls?

```

ACMD(do_drive)
{
    int    dir;
    struct obj_data *vehicle, *controls;

    if (GET_POS(ch) < POS_SLEEPING) {
        send_to_char(ch, "You can't see anything but stars!\r\n");
    } else if (AFF_FLAGGED(ch, AFF_BLIND)) {
        send_to_char(ch,
            "You can't see a damned thing, you're blind!\r\n");
    } else if (IS_DARK(IN_ROOM(ch)) && !CAN_SEE_IN_DARK(ch)) {
        send_to_char(ch, "It is pitch black...\r\n");
    } else if (!(controls = find_control(ch) )) {
        send_to_char(ch,
            "You have no idea how to drive anything here.\r\n");
    } else if (invalid_align(ch, controls) ||
        invalid_class(ch, controls) ||
        invalid_race(ch, controls)) {
        act("You are zapped by $p and instantly
            step away from it.",
            FALSE, ch, controls, 0, TO_CHAR);
        act("$n is zapped by $p and instantly steps away from it.",
            FALSE, ch, controls, 0, TO_ROOM);
    } else if (!(vehicle =
        find_vehicle_by_vnum(GET_OBJ_VAL(controls, 0))) ) {
        send_to_char(ch, "You can't find anything to drive.\r\n");
    } else {
        char    arg[MAX_INPUT_LENGTH], arg2[MAX_INPUT_LENGTH];

        argument = any_one_arg(argument, arg);
        one_argument(argument, arg2);
        if (!*arg) {
            send_to_char(ch, "Drive, yes, but where?\r\n");
        } else if (is_abbrev(arg, "into") ||
            is_abbrev(arg, "inside") ||
            is_abbrev(arg, "onto") ) {
            /* Driving into another vehicle */
            drive_into_vehicle(ch, vehicle, arg2);
        } else if (is_abbrev(arg, "outside")) {
            drive_outof_vehicle(ch, vehicle);
        } else if ((dir = search_block(arg, dirs, FALSE)) >= 0 ) {
            /* Drive in a direction... */
            drive_in_direction(ch, vehicle, dir);
        } else {
            send_to_char(ch, "That's not a valid direction.\r\n");
        }
    }
}

```

Two helper functions locate the vehicle controls and the actual vehicle. The first action is to locate a vehicle control object, which is what the function “find_control” does, first looking in the room the character is in for a control type object. Failing that, the character’s inventory is searched, and finally the equipment. Then, we try to find the vehicle that the control is meant to control. Function “find_vehicle_by-vnum” does this (although in a pretty strange way).

Once we are satisfied that the player can actually drive the vehicle, we look at the command arguments. Three different functions are called – “drive_into_vehicle”, “drive_out_of_vehicle” or “drive_in_direction” are called depending on what they requested. Of course, if some invalid entry was made, like “drive to the bar” or “drive fast”, the error message “That’s not a valid direction” is given.

It is the role of the three “perform” functions to validate the type of driving that was specified on the command.

2.4.3 Specifying a Direction to Drive

If the player enters the command “drive <direction>”, the routine

```

void drive_in_direction(struct char_data *ch,
                      struct obj_data *vehicle, int dir)
{
    char    buf[MAX_INPUT_LENGTH];

    if (!EXIT(vehicle, dir) ||
        EXIT(vehicle, dir)->to_room == NOWHERE) {
        /* But there is no exit that way */
        send_to_char(ch,
                    "Alas, you cannot go that way...\r\n");
    } else if (IS_SET(EXIT(vehicle, dir)->exit_info,
                    EX_CLOSED)) {
        /* But the door is closed */
        if (EXIT(vehicle, dir)->keyword)
            send_to_char(ch, "The %s seems to be closed.\r\n",
                        fname(EXIT(vehicle, dir)->keyword));
        else
            send_to_char(ch, "It seems to be closed.\r\n");
    } else if (!IS_SET_AR(ROOM_FLAGS(EXIT(vehicle,dir)-
>to_room),ROOM_VEHICLE)) {
        /* But the vehicle can't go that way*/
        send_to_char(ch,
                    "The vehicle can't manage that terrain.\r\n");
    } else {
        /* But nothing!  Let's go that way! */
        int was_in, is_in;
        sprintf(buf, "%s leaves %s.\n\r",
                vehicle->short_description, dirs[dir]);
        send_to_room(vehicle->in_room, buf);
        was_in = vehicle->in_room;
        obj_from_room(vehicle);
        obj_to_room(vehicle,
                    world[was_in].dir_option[dir]->to_room);
        is_in = vehicle->in_room;
        if (ch->desc != NULL)
            look_at_room(is_in, ch, 0);
        sprintf(buf, "%s enters from the %s.\r\n",
                vehicle->short_description,
                dirs[rev_dir[dir]]);
        send_to_room(is_in, buf);
    }
}

```

Again, there are the sort of sanity checks you would expect – there is no exit from the room the vehicle is in it that direction, or a door that way is closed, or the destination room is not flagged as accepting vehicles.

Assuming all will be okay with driving in that direction, we remove the vehicle from the starting room, and place it in the destination room. Appropriate messages are sent to all characters in the source and destination room, using “send_to_room”, because we can’t use “act()” since the character is not necessarily in the vehicle’s room. As well, we show the character the new room that the vehicle moves to.

2.4.4 Driving Into Other Vehicles

We don’t mean t-boning some other vehicle; rather, this is the ability for a vehicle to enter another vehicle. First, we need a suitable vehicle that we can drive out vehicle on and off. The objects and rooms involved in this are listed in section 4.2 A Vehicle for Other Vehicles – the River Ferry on page 39. We also set the “vehicle” flag on the river rooms, so our ferry can be driven along the river.

If the player enters the command “drive into <vehicle>”, the routine “drive_into_vehicle” is called to do the driving. Again, some sanity checks first:

- They have to actually specify something to drive into.
- The vehicle they have specified has to be in the room the vehicle they are driving is in
- That object has to be a vehicle.
- The room inside the vehicle has to have its vehicle flag set.
- You can’t drive a vehicle into itself.

Assuming all that works, we then go through a similar process in moving the vehicle from the starting room to the room within the destination vehicle as we would with the drive in a direction process, and send messages using “send_to_room” to the rooms involved.

```
void drive_into_vehicle(struct char_data *ch,
                      struct obj_data *vehicle, char *arg)
{
    struct obj_data *vehicle_in_out;
    int    was_in, is_in, is_going_to;
    char   buf[MAX_INPUT_LENGTH];

    if (!*arg) {
        send_to_char(ch, "Drive into what?\r\n");
    } else if (!(vehicle_in_out =
                get_obj_in_list_vis(ch, arg, NULL,
                world[vehicle->in_room].contents)) ) {
        send_to_char(ch, "Nothing here by that name!\r\n");
    } else if (GET_OBJ_TYPE(vehicle_in_out) != ITEM_VEHICLE) {
        send_to_char(ch, "That's not a vehicle.\r\n");
    } else if (vehicle == vehicle_in_out) {
        send_to_char(ch,
            "My, we are in a clever mood today, aren't we.\r\n");
    } else {
```

```

is_going_to = real_room(GET_OBJ_VAL(vehicle_in_out, 0));
if (!IS_SET_AR(ROOM_FLAGS(is_going_to), ROOM_VEHICLE)) {
    send_to_char(ch,
        "That vehicle can't carry other vehicles.");
} else {
    sprintf(buf, "%s enters %s.\n\r",
        vehicle->short_description,
        vehicle_in_out->short_description);
    send_to_room(vehicle->in_room, buf);
    was_in = vehicle->in_room;
    obj_from_room(vehicle);
    obj_to_room(vehicle, is_going_to);
    is_in = vehicle->in_room;
    if (ch->desc != NULL)
        look_at_room(is_in, ch, 0);
    sprintf(buf, "%s enters.\r\n",
        vehicle->short_description);
    send_to_room(is_in, buf);
}
}
}

```

If we wish to drive our vehicle out of another, “drive_out_of_vehicle” does the work for us. The various sanity checks are firstly, is there a hatch in our vehicle’s room, and secondly, can we find the vehicle this hatch belongs to?

Assuming all is right, we again move the vehicle object from its current room to the room that the outer vehicle is within, and send messages to both rooms.

```

void drive_outof_vehicle(struct char_data *ch,
                        struct obj_data *vehicle)
{
    struct obj_data *hatch, *vehicle_in_out;
    char buf[MAX_INPUT_LENGTH];

    if ( !(hatch = get_obj_in_list_type(ITEM_HATCH,
        world[vehicle->in_room].contents)) ) {
        send_to_char(ch, "Nowhere to drive out of.\r\n");
    } else if (!(vehicle_in_out =
        find_vehicle_by_vnum(GET_OBJ_VAL(hatch, 0)))) {
        send_to_char(ch, "You can't drive out anywhere!\r\n");
    } else {
        sprintf(buf, "%s exits %s.\r\n",
            vehicle->short_description,
            vehicle_in_out->short_description);
        send_to_room(vehicle->in_room, buf);
        obj_from_room(vehicle);
        obj_to_room(vehicle, vehicle_in_out->in_room);
        if (ch->desc != NULL)
            look_at_room(vehicle->in_room, ch, 0);
        sprintf(buf, "%s drives out of %s.\r\n",
            vehicle->short_description,
            vehicle_in_out->short_description);
        send_to_room(vehicle->in_room, buf);
    }
}

```

2.4.5 Driving On and Off The Ferry

We can now look at see how all this works together. First, we drive the coach to the ferry terminal, where a ferry waits to take us across the river.

```
500H 100M 82V > drive s
The Ferry Terminal
    This is where the ferry to the other side of the river
    departs from. The road runs into the water so that wagons can
    drive on and off the ferry.
[ Exits: n s ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
This ferry carries people and wagons across the river.

500H 100M 82V > drive into ferry
The Ferry
You are one the ferry, travelling across the river.
[ Exits: None! ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
A wheel lets someone control the ferry.
A ramp lets you walk or drive on and off the ferry.
```

Now let's get out of the coach and drive the ferry.

```
500H 100M 82V > down
Inside An Old Coach
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
[ Exits: u ]
A small window provides a view of the outside world.
A door provides a way out of the coach.

500H 100M 82V > leave
Phew! You are glad to be out of that stinking coach.

The Ferry
You are one the ferry, travelling across the river.
[ Exits: None! ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
A wheel lets someone control the ferry.
A ramp lets you walk or drive on and off the ferry.

500H 100M 82V > look out
You look outside and see:
The Ferry Terminal
    This is where the ferry to the other side of the river
    departs from. The road runs into the water so that wagons can
    drive on and off the ferry.
[ Exits: n s ]
This ferry carries people and wagons across the river.

500H 100M 82V > drive south
On The River
```

```
The river current picks up speed as it flows past the
southern wall of Midgaard. Upstream to the east, you see the
Midgaard docks, and beyond them, the bridge that connects South
Midgaard to the city proper. To the west, a dark forest
stretches from the water's edge off to the far distance.
[ Exits: n e w ]
This ferry carries people and wagons across the river.

500H 100M 82V >
```

And when we get back to the shore, we can drive off the ferry.

```
500H 100M 82V > drive n
The Ferry Terminal
    This is where the ferry to the other side of the river
departs from. The road runs into the water so that wagons can
drive on and off the ferry.
[ Exits: n s ]
This ferry carries people and wagons across the river.

500H 100M 82V > enter coach
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.

Inside An Old Coach
    The seats are stained, the floor is covered with dirt and
who knows what else. You are certainly traveling third class
in this coach.
[ Exits: u ]
A small window provides a view of the outside world.
A door provides a way out of the coach.

500H 100M 82V > u
The Driver's Seat
    From up here at the front of the coach, the driver would
normally have control of the horses' reins and the coach brake.
Strange, there doesn't seem to be a driver.
[ Exits: d ]
A set of leather reins is harnessed to the horses.

500H 100M 82V > drive out
The Ferry Terminal
    This is where the ferry to the other side of the river
departs from. The road runs into the water so that wagons can
drive on and off the ferry.
[ Exits: n s ]
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.
This ferry carries people and wagons across the river.

500H 100M 82
```

2.5 Controlling Access to Vehicles

Vehicles and controls can be restricted to prevent certain classes, races or alignments from using these objects. This is done by specifying the same object extra flags that apply to weapons and armor – ANTI_GOOD, ANTI-MAGE, ANTI-ELF for example. The same “act” messages are sent to the player and to the room if they try to either enter a vehicle or drive with a control that they are not permitted to use.

```
You are zapped by $p and instantly step away from it.  
$n is zapped by $p and instantly steps away from it.
```

This works in exactly the same manner as the class, alignment and race checks in the “equip_char” function in handler.c, with the exception that the character doesn’t drop the item when zapped by it, since they are generally not holding either the vehicle or the controls.

2.6 Shortcomings With The Vehicles Implementation

There are a number of potential problems builders need to be aware of with the current implementation of vehicles.

1. There is no distinction between land, sea or air vehicles. In the ferry example above, we could have driven our coach into the ferry at the ferry wharf, then driven the ferry up the river (assuming rooms were vehicle enabled). Now, part of the way up the river, we could have driven our coach out of the ferry, and the coach would be there in the river.
2. The use of “send_to_room” for the various messages, calling “sprintf” and the use of buffers is not the ideal way to generate and send messages.
3. If allowed by rooms being vehicle enabled, a vehicle could move from one zone to another. However, the characters within the vehicle still occupy rooms within their original zone – in fact, these rooms never “move”. It is only the vehicle object that moves from room to room. This could cause problems with zones resetting, and either creating duplicates of vehicle objects, or not resetting because player characters are still thought to be within the zone.
4. Vehicle objects really need to be unique, since the vehicle vnum is tied to a single room number, and other vehicle objects (hatches, windows and controls) are also tied by the same vnum, and will locate the first instantiation of that object.

To illustrate the main disadvantage – no distinction between land and sea, here is an example. We start out in the ferry, which also contains our coach, up the river a bit.

```
500H 100M 82V > drive e  
On The River  
    North of here you see the miserable buildings of Dark Alley.  
The river flows from east to west. The riverbanks are too  
steep to climb.  
[ Exits: e w ]  
This ferry carries people and wagons across the river.  
  
500H 100M 82V > enter coach
```

You open the door and climb into the coach. You look at the grimy state of the interior, and try to find the least filthy part of the seat to sit on. This vehicle has certainly seen better days.

Inside An Old Coach

The seats are stained, the floor is covered with dirt and who knows what else. You are certainly traveling third class in this coach.

[Exits: u]

A small window provides a view of the outside world.

A door provides a way out of the coach.

500H 100M 82V > u

The Driver's Seat

From up here at the front of the coach, the driver would normally have control of the horses' reins and the coach brake. Strange, there doesn't seem to be a driver.

[Exits: d]

A set of leather reins is harnessed to the horses.

500H 100M 82V > drive out

On The River

North of here you see the miserable buildings of Dark Alley. The river flows from east to west. The riverbanks are too steep to climb.

[Exits: e w]

An old, weather-beaten coach, pulled by two horses, waits for some passengers.

This ferry carries people and wagons across the river.

500H 100M 82V > drive w

On The River

North of here you see the dump. The river flows from east to west. The riverbanks are too steep to climb.

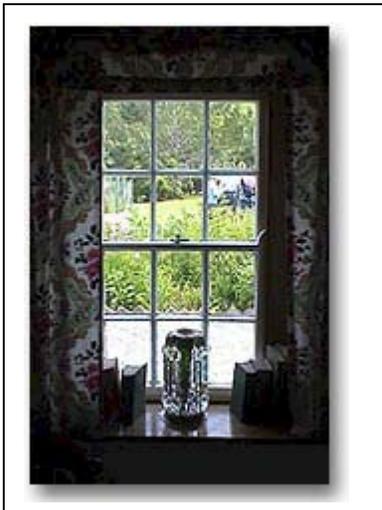
[Exits: e w]

An old, weather-beaten coach, pulled by two horses, waits for some passengers.

500H 100M 82V >

Driving a coach and horses on (in?) the river doesn't quite seem right.

3 More Windows Options



As well as provide means for players to see outside of vehicles, a window object can be added directly to a room. This can allow players to look outside the room, or it can provide a way to see a room that is not directly connected to where they are.

We make use of two terms when dealing with windows. The first is the “viewport” – the actual object that the player is looking through. Obviously, this needs to be defined as having an object type of “ITEM_WINDOW”. The second term is the window “target” – the room vnum that can be seen through the window. In the earlier situation where we defined a window in a vehicle, the target was the room that the vehicle was in at the time the player looked out the window.

Note that we can use a variety of prepositions when looking out a window – look out <window>, look through (or thru) <window>, or even look in (or inside or into) <window>. A person can even enter “look outside” or “look out”.

3.1 Simple Windows

We want to create a way for someone to type “look outside” and be given the description of the room that is outside. We create a window almost the same as before:

```
500H 100M 82V > oedit 2875
-- Item number : [2875]
1) Namelist : window
2) S-Desc   : an invisible window
3) L-Desc   :-
Undefined
4) A-Desc   :-
<not set>
5) Type     : WINDOW
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day : 0
B) Timer    : 0
C) Values   : -1 0 -1 -1
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script    : Not Set.
Q) Quit
Enter choice :
```

Here we have the “invisible window”, that is, one that doesn’t appear on the room listing. Note the change though with the object values:

```

Enter choice : c
Enter vnum of the vehicle this window belongs to, or -1 to
specify the viewport room : -1
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: NOBITS
Enter flag, 0 to quit : 0
Vnum of key to unlock window (-1 for no key) : -1
What is the viewport room vnum (-1 for default location) : -1

```

Here, rather than the vnum of a vehicle, we specify “-1”, to indicate that the window is not associated through a vehicle object, but loaded directly into the room. And we specify the default “viewport” (or destination) vnum.

So, when we type “look outside” or “look out window”, what will we see?

```

500H 100M 82V > look
[ 2854] Trading Post [ INDOORS ]
    Bolts of cloth and barrels of spices stand in monstrous
heaps, arranged in a chaotic haphazard of valuable trade goods.
As you watch, a small man enters from the back room, grabs a
box of something unidentifiable, and leaves just as quickly. A
schedule on the wall keeps track of the caravans and ships
passing through the city -- very important information for a
merchant.
[ Exits: s ]
[3060] A cityguard stands here.
[2816] A Merchant Trader is doing inventory in his head.

500H 100M 82V > look outside
You look outside and see:
[ 2815] Trade Road [ VEHICLE ]
    The traffic on this road seems light at best, becoming a bit
thicker to the east. Westwards, Trade Road continues towards
the intersection with Clay Street. A large open-air building
is to the north, and southwards is a large open lot; a
fairgrounds, closed for the time being.
[ Exits: n e w ]
[2870] An old, weather-beaten coach, pulled by two horses,
waits for some passengers.

500H 100M 82V >

```

The room we see is exactly the same as the room we would go to if we types “leave”. Again, if we knew (or suspected) that there may be a window here, we could have types “look out window”, or even “look out”. If we examine the code in the “do_look” routine, we can see what is happening.

```

if (!*arg) /* "look" alone, without an argument at all */
    look_at_room(IN_ROOM(ch), ch, 1);
else if (is_abbrev(arg, "in"))
    look_in_obj(ch, arg2);
else if (((is_abbrev(arg, "outside")) ||
          (is_abbrev(arg, "through")) ||
          (is_abbrev(arg, "thru")))) && (*arg2))
    look_out_window(ch, arg2);

```

```

else if (is_abbrev(arg, "outside") && !EXIT(ch, OUTDIR))
    look_out_window(ch, arg2);
/* did the char type 'look <direction>?' */
else if ((look_type = search_block(arg, dirs, FALSE)) >= 0)
    look_in_direction(ch, look_type);
else if (is_abbrev(arg, "at"))
    look_at_target(ch, arg2);
else
    look_at_target(ch, arg);

```

We test for the “out” cases before we test the “look <direction>”, since in many cases where additional exits have been defined, “out” is a valid direction. We only want the “out” direction” to be processed if there is no window to look out.

If a value for arg2 is provided – that is, a window object – then look outside, look through, or look thru (or any legitimate abbreviation) will call “look_out_window” with that as the object to look out of. Otherwise, if no arg2 value is provided, but the player types “look outside” (or “look out”, more likely), and there is no direction defined in the “out” direction, then “look_out_window” is also called, this time passing a null object string.

So, how do we determine what to look out of, and where they should look to? Let’s examine the “look_out_window” code, in particular the area where we determine what the viewport should be.

```

/* First, lets find something to look out of or through. */
if (*arg) {
    /* Find this object and see if it is a window */
    if (!(bits = generic_find(arg,
        FIND_OBJ_ROOM | FIND_OBJ_INV | FIND_OBJ_EQUIP,
        ch, &dummy, &viewport))) {
        send_to_char(ch, "You don't see that here.\r\n");
        return;
    } else if (GET_OBJ_TYPE(viewport) != ITEM_WINDOW) {
        send_to_char(ch, "You can't look out that!\r\n");
        return;
    }
} else if (OUTSIDE(ch)) {
    /* yeah, sure stupid */
    send_to_char(ch, "But you are already outside.\r\n");
    return;
} else {
    /* Look for any old window in the room */
    for (i = world[IN_ROOM(ch)].contents; i;
        i = i->next_content)
        if ((GET_OBJ_TYPE(i) == ITEM_WINDOW) &&
            isname("window", i->name)) {
            viewport = i;
            continue;
        }
}

```

The first “else if” about halfway down the above list is where we consider the situation where no object was provided on the “look outside” command. Clearly, if we are already outside, then the command makes no sense.

Otherwise, we need to find a suitable object to look outside through. For this to work, the window's namelist has to include the word "window". The reason for this is in the situation where there may be another window object in the room, we want to ensure that the "look outside" command will execute on an object that we have defined as a window that shows the outside, rather than just any object of type "ITEM_WINDOW" that may be in that room. This will become clearer later on.

If we couldn't find a suitable object to look outside through, the "viewport" object remains set to null, and this is checked later in the function.

3.2 Specifying the Window's Target Room

Not all windows will look "outside" – that is, where outside is the room that you would get to if you typed in the command "leave". You may have a room with a window looking over the street below, for example, but not have that as an exit. In this situation we specify the target room directly. The window could still be "invisible", in that it is not listed as an object directly, or you may choose to make it immediately apparent to all that it is there, waiting to be looked through.

Remember, if you choose to make the window "invisible", you should make it obvious from the room description that there is a way to see outside, and include the word "window" in the object's namelist.

Here is a room and the associated window:

```
500H 100M 82V > look
[ 3188] The Hayloft [ INDOORS ]
    This loft runs the length of the building above the stables,
    and contains bales of hay, assorted bags of feed, and a
    collection of old leatherworking tools. At the eastern end a
    large opening allows hay to be tossed up from the street
    outside.
[ Exits: d ]
[3126] ( 2) A young boy is resting here, hoping his master
won't find him.

500H 100M 82V >
oedit 3132
-- Item number : [3132]
1) Namelist : window opening
2) S-Desc   : an invisible window
3) L-Desc   :-
Undefined
4) A-Desc   :-
<not set>
5) Type     : WINDOW
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day : 0
B) Timer    : 0
C) Values   : -1 0 -1 3149
D) Applies menu
E) Extra descriptions menu
```

```
M) Min Level      : 0
P) Perm Affects: NOBITS
S) Script         : Not Set.
Q) Quit
Enter choice :
```

You can see how the room description refers to the opening at one end. As well, the window object has both “window” and “opening” as entries in its namelist, and the empty long description makes sure it is not included displayed list of contents.

Notice how we have specified the target room:

```
Enter choice : c
Enter vnum of the vehicle this window belongs to, or -1 to
specify the viewport room : -1
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: NOBITS
Enter flag, 0 to quit : 0
Vnum of key to unlock window (-1 for no key) : -1
What is the viewport room vnum (-1 for default location) : 3149
```

So, when we “look outside”, we see the street below

```
500H 100M 82V > look outside
You look outside and see:
[ 3149] Emerald Avenue [ VEHICLE ]
    You are on Emerald Avenue, which continues north. The
Concourse is south of here, and there is a large building to
the east, from which you can hear (and smell) horses. There is
a sign over the building entrance.
[ Exits: n e s ]
[3121] A large trough is filled with clean water.
[3014] A citizen of Midgaard is here.
```

We get the same result if we say “look through opening”:

```
500H 100M 82V > look through opening
You look outside and see:
[ 3149] Emerald Avenue [ VEHICLE ]
    You are on Emerald Avenue, which continues north. The
Concourse is south of here, and there is a large building to
the east, from which you can hear (and smell) horses. There is
a sign over the building entrance.
[ Exits: n e s ]
[3121] A large trough is filled with clean water.
```

If we wanted to change the default message that appears before the target room description, then we would need to provide an action description for the window object.

Let’s turn back to the `look_out_window` function, to see how we determine what we should show to the player. The earlier listing showed how we validated the supplied name of the window object if it was provided, first to make sure the object is in the room (or the player’s inventory or equipment), and secondly to make sure it is the right object type. Now, looking at the second half of that function, we can see how the viewport is used to determine the room and show that to the player.

```

if (!viewport) {
    /* Nothing suitable to look through */
    send_to_char(ch,
        "You don't seem to be able to see outside.\r\n");
} else if (OBJVAL_FLAGGED(viewport, CONT_CLOSEABLE) &&
    OBJVAL_FLAGGED(viewport, CONT_CLOSED)) {
    /* The window is closed */
    send_to_char(ch, "It is closed.\r\n");
} else {
    if (GET_OBJ_VAL(viewport, 0) < 0) {
        /* We are looking out of the room */
        if (GET_OBJ_VAL(viewport, 3) < 0) {
            /* Look for the default "outside" room */
            for (door = 0; door < NUM_OF_DIRS; door++)
                if (EXIT(ch, door))
                    if (EXIT(ch, door)->to_room != NOWHERE)
                        if (!ROOM_FLAGGED(EXIT(ch, door)->to_room,
                            ROOM_INDOORS)) {
                            target_room = EXIT(ch, door)->to_room;
                            continue;
                        }
                } else {
                    target_room = real_room(GET_OBJ_VAL(viewport, 3));
                }
        } else {
            /* We are looking out of a vehicle */
            if ( (vehicle =
                find_vehicle_by_vnum(GET_OBJ_VAL(viewport, 0))) );
                target_room = vehicle->in_room;
        }
        if (target_room == NOWHERE) {
            send_to_char(ch,
                "You don't seem to be able to see outside.\r\n");
        } else {
            if (viewport->action_description)
                act(viewport->action_description, TRUE,
                    ch, viewport, 0, TO_CHAR);
            else
                send_to_char(ch, "You look outside and see:\r\n");
            look_at_room(target_room, ch, 0);
        }
    }
}

```

We first make some sanity checks, such as not being able to find a suitable viewport, or the viewport being closed. Once we know we have a viewport that we can look out of, we have to see what type of viewport it is – a window belonging to a vehicle, or a window that is just on a room. The first object flag value determines this – if it is negative, then the window is associated with a room.

If the window is associated with a room, the fourth object value specifies the target room number. If this is -1, we search for a door leading outside, and if found, use that door's destination as the target room. Otherwise, object value four is the target room's vnum.

If the window is associated with a vehicle, we find that vehicle object, and the room that the vehicle is currently in becomes our target room.

If we cannot find a real room number that corresponds to the target room vnum, we give an error message, otherwise we display a message that we are looking outside (or the window's action description), and then call `look_at_room` on the target room.

3.3 Windows That Don't Look Like Windows

Now that we have a mechanism for creating objects that a player can look through (or out of) to display some other unrelated room, we can create a wide variety of objects that don't appear to be windows, but still use the same behavior.

3.3.1 Repairing the Vandalized Roman Telescope



The stock CircleMUD zone of Rome has an observation platform, which appears to contain a telescope. However, this is only an “extra description” in the room, and the telescope has been vandalized, and players cannot interact with the telescope. However, we now have the ability to repair it, by making it a window. First, we edit the room definition to remove the now redundant telescope e-desc. The mention of the telescope could also be removed from the room description, as it will show up in the list of objects in that room.

Secondly, we create a new window object, and target it to the top of the mountains:

```
500H 100M 82V > oedit 12040
-- Item number : [12040]
1) Namelist : telescope
2) S-Desc   : a brass telescope
3) L-Desc   :-
A fine brass telescope is mounted on the railing.
4) A-Desc   :-
You adjust the focussing knob and the image becomes clear.
5) Type     : WINDOW
6) Extra flags : NOBITS
7) Wear flags : NOBITS
8) Weight   : 0
9) Cost     : 0
A) Cost/Day  : 0
B) Timer     : 0
C) Values    : -1 0 -1 12047
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script    : Not Set.
Q) Quit
Enter choice :
```

We can also add a suitable extra description to the telescope, so we can also look at it.

```
500H 100M 82V > look
[12040] The Observation Lookout [ NOBITS ]
    You are on a platform overlooking a massive aqueduct. There
is a telescope mounted to the railing that surrounds the area
and you can hear the gurgle of water as it is channeled into
the city. Off in the distance, you can see the Mountain of the
Gods.
There is a sign mounted to the railing and a ramp leads west.
[ Exits: w ]
[12040] A fine brass telescope is mounted on the railing.
[12019] A soldier on leave is walking around looking for
entertainment.

500H 100M 82V > look at telescope
This finely crafted brass telescope should provide a great view
of the mountain tops.
```

This is looking more like it. Now lets use this telescope

```
500H 100M 82V > look through telescope
You adjust the focussing knob and the image becomes clear.
[12047] The Summit Of The Mountain Of The Gods [ NOBITS ]
    You are at the summit of a very high mountain and are
standing above the clouds. The sun is shining brilliantly here
and the clouds below look like balls of cotton. The path that
you have been walking on is damp and the scent of lilac is in
the air.
[ Exits: d ]
[12030] Jupiter, King of the Gods, is watching over his realm
here.
```

Perfect. One fully functional telescope and we don't even have to put a quarter in the slot. The only minor drawback is we cannot move it to look at other locations.

3.3.2 Portable Objects – A Crystal Ball



Our final example is an object that no medium or soothsayer should be without – a crystal ball. This one is designed to show a specific far-off location if someone looks into it.

Of course, one could be a little deceptive in how the crystal ball first appears. We really should have players determine for themselves what this object is, rather than broadcast “Hey – I am a crystal ball. Look into me to see something neat!” It is always good to make players think about things (and also read all the words that you create).

This means making the object description (the l-desc) describe the object's appearance only, and use extra descriptions to give greater clues to its purpose. Of course, the namelist should include the words “crystal” and “ball” so that when players work out the object's purpose, they can refer to it as a crystal ball.

```

500H 100M 82V > oedit 3133
-- Item number : [3133]
1) Namelist : opaque glass sphere crystal ball
2) S-Desc   : a glass sphere
3) L-Desc   :-
This glass sphere seems to contain a cloudy vapour.
4) A-Desc   :-
You gaze into the white swirl. After a few seconds, the mists
clear and shapes, colours and objects start to take shape
within the sphere.
5) Type     : WINDOW
6) Extra flags : GLOW MAGIC
7) Wear flags : TAKE
8) Weight   : 3
9) Cost     : 0
A) Cost/Day : 0
B) Timer    : 0
C) Values   : -1 0 -1 2677
D) Applies menu
E) Extra descriptions menu
M) Min Level : 0
P) Perm Affects: NOBITS
S) Script    : Not Set.
Q) Quit
Enter choice : e
Extra desc menu
1) Keyword: opaque glass sphere crystal ball
2) Description:
This glass sphere is about six inches in diameter, and seems to
be filled with a swirling opaque vapour. You can sense some
strange power drawing your eyes towards the sphere, and you
feel compelled to gaze within it.

3) Goto next description: <Not set>

0) Quit
Enter choice :

```

Notice that we have allowed the object to be taken, plus added the extra flags of “glow” and “magic”.

```

500H 100M 82V > look
[ 1204] The Immortal Board Room [ INDOORS ]
    The main hang out of the Gods, the Immortal Board Room is
the place to be. Gods exchange messages here most every day.
The mortal board room is to the east and the meeting room for
the gods is to the south. To the north is the Gods' Inn and to
the west is a post office for Gods. There is a large staircase
leading down to the main temple in the city of Midgaard. In
the northeast corner you spot a small staircase leading
upwards.
[ Exits: n e s w u d ]
[3133] This glass sphere seems to contain a cloudy vapour. ..It
has a soft glowing aura!
[3098] A large bulletin board is mounted on a wall here. It
glows with a faint aura.

```

```

500H 100M 82V > look at sphere
This glass sphere is about six inches in diameter, and seems to
be filled with a swirling opaque vapour. You can sense some
strange power drawing your eyes towards the sphere, and you
feel compelled to gaze within it.
..It has a soft glowing aura!

500H 100M 82V > look in sphere
You gaze into the white swirl. After a few seconds, the mists
clear and shapes, colours and objects start to take shape
within the sphere.

[ 2677] The Audience Chamber [ INDOORS NO_TRACK ]
    You are in a large chamber. Small bolts of energy leap from
wall to wall, crackling with power. The floor is a swirl of
indeterminable colours and patterns ever changing. Bright
bursts of colour seem to explode into the air and fade again at
random times. All this splendor plays about and draws
attention to a large throne in the center of the room, it seems
to be cut from a single enormous emerald.
[ Exits: n e s w ]
[2564] The Grand Mistress of Magic is sitting here.
[2563] A large calico cat is lounging about here.

500H 100M 82V >

```

Notice that here we typed in “look in ball” – not “look through ball”. A minor addition has been made to “look_in_obj”, in the section where the object being looked into is checked to see what type it is:

```

} else if (GET_OBJ_TYPE(obj) == ITEM_WINDOW) {
    look_out_window(ch, arg);

```

Of course, the same results would occur if the player entered “look out ball” or “look through ball”, but they are not the intuitive things one would do with a crystal ball.

Again, the crystal ball will only ever show the one destination. This looks like a useful object – maybe we should take this with us:

```

500H 100M 82V > take sphere
You get a glass sphere.

500H 100M 82V > inv
You are carrying:
a glass sphere ..It has a soft glowing aura!

```

3.4 Windows That Close and Lock

In all these examples, we haven’t made use of the second and third object flag values that specify the window as being able to be closed and / or locked, and the vnum of any key associated with the window. However, this works in exactly the same manner as any other closeable object. All you need to do is specify the closeable and lockable attributes as the second and third object values, just like you would do with a standard container. Lets go back to our coach window and make it closeable.

```

Enter choice : c
Enter vnum of the vehicle this window belongs to, or -1 to
specify the viewport room : 2870
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: NOBITS
Enter flag, 0 to quit : 1
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: CLOSEABLE
Enter flag, 0 to quit : 3
1) CLOSEABLE
2) PICKPROOF
3) CLOSED
4) LOCKED
Container flags: CLOSEABLE CLOSED
Enter flag, 0 to quit : 0
Vnum of key to unlock window (-1 for no key) : -1
What is the portal's appearance (-1 for transparent) : -1

```

We now have a window that can be closed.

```

500H 100M 82V > enter coach
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.

[ 2860] Inside An Old Coach [ INDOORS NO_TRACK ]
    The seats are stained, the floor is covered with dirt and
    who knows what else. You are certainly traveling third class
    in this coach.
[ Exits: u ]
[2873] A small window provides a view of the outside world.
[2871] A door provides a way out of the coach.

500H 100M 82V > look out window
It is closed.

500H 100M 82V > open window
Okay.

500H 100M 82V > look out window
You look outside and see:
[ 2815] Trade Road [ VEHICLE ]
    The traffic on this road seems light at best, becoming a bit
    thicker to the east. Westwards, Trade Road continues towards
    the intersection with Clay Street. A large open-air building
    is to the north, and southwards is a large open lot; a
    fairgrounds, closed for the time being.
[ Exits: n e w ]
[2870] An old, weather-beaten coach, pulled by two horses,
    waits for some passengers.

```

However, despite windows having some container-like attributes and behavior, they still cannot be used to place other items inside them, or take things from them.

```
500H 100M 82V > inv
You are carrying:
a glass sphere ..It has a soft glowing aura!

500H 100M 82V > put sphere in window
A window is not a container.
```

4 World Files Code Details

4.1 The Complete Coach

Remember that a complete vehicle is a complex composite of several objects and rooms, all of which need to interact for the vehicle to work correctly.

First, we have the vehicle object itself, and a hatch. This allows players to enter and leave the vehicle.

```
#2870
coach~
a weather-beaten coach~
An old, weather-beaten coach, pulled by two horses, waits for
some passengers.~
You open the door and climb into the coach. You look at the
grimy state of the interior, and try to find the least filthy
part of the seat to sit on. This vehicle has certainly seen
better days.
~
24 0 0 0 0 0 0 0 0 0 0 0 0
2860 0 -1 -1
0 0 0 0
E
coach~
The coach has seen better days. While it looks functional, the
paint on the sides has long faded, and several (hopefully
unimportant) parts are held on with lashings of rope. It looks
like about six people could travel inside, and several more on
the top with the driver. Two bored horses are hitched to the
front.
~
E
horse horses~
You wonder how these old nags could pull this coach. They
appear to be refugees from the knacker's yard.
~
#2871
door~
a door~
A door provides a way out of the coach.~
Phew! You are glad to be out of that stinking coach.
~
25 0 0 0 0 0 0 0 0 0 0 0 0
2870 5 -1 0
0 0 0 0
E
door~
Thank goodness! At least you can get out of this vile coach
when you want to. The door's lock has vanished long ago, now a
loop of twine holds the door closed.
~
```

Next, we add a control object, so the vehicle can be driven from room to room. Don't forget that rooms will need the VEHICLE room flag added to them.

```

#2872
reins~
a set of leather reins~
A set of leather reins is harnessed to the horses.~
~
27 0 0 0 0 0 0 0 0 0 0 0 0 0
2870 0 0 0
0 0 0 0
E
reins~
These reins are harnessed to the horses, by pulling on these to
exert pressure, and no doubt with a lot of shouting, you could
direct the horses to pull the coach somewhere.
~

```

Finally, we add two windows, one an obvious window to the coach interior, and the other an “invisible” window loaded to the driver’s area on the top of the coach.

```

2873
window~
a window~
A small window provides a view of the outside world.~
~
26 0 0 0 0 0 0 0 0 0 0 0 0 0
2870 5 -1 0
0 0 0 0
#2874
window~
window~
Undefined~
~
26 0 0 0 0 0 0 0 0 0 0 0 0
2870 0 -1 0
0 0 0 0

```

Once we have the objects that make up the vehicle, we need some rooms

```

#2860
Inside An Old Coach~
    The seats are stained, the floor is covered with dirt and
    who knows what else.  You are certainly traveling third class
    in this coach.
~
28 dg 0 0 0 0
D4
You can climb up to the driver's seat.
~
~
0 0 2861
S
#2861
The Driver's Seat~
    From up here at the front of the coach, the driver would
    normally have control of the horses' reins and the coach brake.
    Strange, there doesn't seem to be a driver.
~
28 dg 0 0 0 1
D5

```

```
You can climb back down to the coach compartment.
~
~
O O 2860
S
```

And lastly, we have to make sure all objects load in the appropriate rooms.

```
* Coach and Windows
R O 2815 2870 0      Coach
O O 2870 1 2815 0
R O 2860 2871 0      Hatch
O O 2871 1 2860 0
R O 2861 2872 0      Control
O O 2872 1 2861 0
R O 2860 2873 0      Window
O O 2873 1 2860 0
R O 2861 2874 0      Window
O O 2874 1 2861 0
R O 2854 2875 0      Window
O O 2875 1 2854 0
```

4.2 A Vehicle for Other Vehicles – the River Ferry

The ferry objects have not been fully fleshed out; there are just enough details here to test the actions in driving one vehicle into another.

```
#2876
ferry~
the river ferry~
This ferry carries people and wagons across the river.~
~
24 0 0 0 0 0 0 0 0 0 0 0 0 0
2856 0 -1 -1
0 0 0 0
E
ferry~
The ferry can carry several people, or one or two wagons,
across the river.
~
#2877
loading ramp~
a loading ramp~
A ramp lets you walk or drive on and off the ferry.~
~
25 0 0 0 0 0 0 0 0 0 0 0 0
2876 0 -1 0
0 0 0 0
#2878
wheel~
a wheel~
A wheel lets someone control the ferry.~
~
27 0 0 0 0 0 0 0 0 0 0 0 0
2876 0 0 0
0 0 0 0
#2879
window~
```

```
an invisible window~
undefined~
~
26 0 0 0 0 0 0 0 0 0 0 0 0 0
2876 0 -1 0
0 0 0 0
```

The rooms as well are basic.

```
2855
The Ferry Terminal~
  This is where the ferry to the other side of the river
departs from. The road runs into the water so that wagons can
drive on and off the ferry.
~
28 q 0 0 0 1
D0
You see Klelk Boulevard and the run down dives that line the
street.
~
~
0 0 2830
D2
You see the river.
~
~
0 0 3199
S
#2856
The Ferry~
You are one the ferry, travelling across the river.
~
28 degq 0 0 0 1
S
```

The zone file entries:

```
* Ferry
R 0 2855 2876 0          Ferry
O 0 2876 1 2855 0
R 0 2856 2877 0          Hatch (Ramp)
O 0 2877 1 2856 0
R 0 2856 2878 0          Wheel
O 0 2878 1 2856 0
R 0 2856 2879 0          Window
O 0 2879 1 2856 0
```

4.3 The Basic Window

```
#2875
window~
an invisible window~
Undefined~
~
26 0 0 0 0 0 0 0 0 0 0 0 0 0
-1 0 -1 -1
0 0 0 0
```

4.4 A Window to a Disconnected Room

```
#3132
window opening~
an invisible window~
Undefined~
~
26 0 0 0 0 0 0 0 0 0 0 0 0
-1 0 -1 3149
0 0 0 0
```

Notice how we include the word “window” in the namelist, so “look outside” will find this as an object that can be used to look outside.

4.5 The Roman Telescope

First, the changes to the Observation Platform:

```
#12040
The Observation Lookout~
    You are on a platform overlooking a massive aqueduct. There
is a telescope mounted to the railing that surrounds the area
and you can hear the gurgle of water as it is channeled into
the city. Off in the distance, you can see the Mountain of the
Gods.
There is a sign mounted to the railing and a ramp leads west.
~
120 0 0 0 0 1
D3
You can see a busy street in that direction.
~
~
0 -1 12033
S
```

The telescope itself:

```
#12040
telescope~
a brass telescope~
A fine brass telescope is mounted on the railing.~
You adjust the focussing knob and the image becomes clear.
~
26 0 0 0 0 0 0 0 0 0 0 0 0
-1 0 -1 12047
0 0 0 0
E
telescope brass~
This finely crafted brass telescope should provide a great view
of the mountain tops.
~
```

4.6 The Crystal Ball

```
#3133
opaque glass sphere crystal ball~
a glass sphere~
This glass sphere seems to contain a cloudy vapour.~
You gaze into the white swirl. After a few seconds, the mists
clear and shapes, colours and objects start to take shape
within the sphere.
~
26 ag 0 0 0 a 0 0 0 0 0 0 0
-1 0 -1 2677
3 0 0 0
E
opaque glass sphere crystal ball~
This glass sphere is about six inches in diameter, and seems to
be filled with a swirling opaque vapour. You can sense some
strange power drawing your eyes towards the sphere, and you
feel compelled to gaze within it.
~
```

5 Help File Changes

Because this has introduced some new commands, and extended the syntax of existing commands, the help system should be modified.

5.1 Look

Adding vehicles and windows provides extra options on the look command. In addition, this patch has made a number of underlying changes to the way the look command (and others) are parsed, to make the language options more natural.

```
500H 100M 82V > help look
LOOK

Usage: look
    look inside | into [the] <item>
    look [at [the]] <item> | <mobile> | <player>
    look outside
    look out[side] | through | thru [the] <item>
    look [towards [the]] <direction>

For studying your surroundings.

Examples:

    > look
Shows you where you are, and what is also in that location.
    > look at the angel
Describes the object you are looking at. Both "at" and "the"
are optional.
    > look in the bag
Will tell you the contents of the bag, if it is open. You can
use in, into or inside; they all have the same meaning.
    > look south
May give some information as to what is south.
    > look outside
May show you the room outside of where you are, if there is a
way to see that way.
    > look out the <item>
If the item is a window-like object, this may describe what you
can see through that item.

Note that if you LOOK AT CORPSE you will not see its inventory.
To see what's inside a container (i.e. a corpse) use LOOK IN
<object>.

See also: EXAMINE, GET, READ, TAKE
```

5.2 Enter

Enter now supports entering objects, as well as entering a room.

```
500H 100M 82V > help enter
ENTER

Usage: enter [place]

If you type 'enter' while you are outdoors, you'll enter any
nearby shelter. You can also follow 'enter' with an argument;
for example, 'enter window'.

500H 100M 82V >
```

5.3 Leave

Likewise, leave also exits a vehicle. However, the command syntax has not changed.

```
500H 100M 82V > help leave
LEAVE

Usage: leave

If you feel claustrophobic, typing 'leave' will make you use
the nearest obvious exit to the outside.

500H 100M 82V >
```

5.4 Drive

The drive command is a brand new command.

```
500H 100M 82V > help drive
DRIVE PILOT

Usage: drive <direction>
       drive into | inside | onto <vehicle>
       drive outside

Use this to control a vehicle you are in. Note that you may
need to be in a particular location in the vehicle, where there
are a set of controls that can actually control the vehicle. A
vehicle may not be able to travel in every possible direction
from its location. Pilot is a synonym for
drive.

See also: VEHICLES
```

5.5 Vehicles

It is probably worth explaining a little about vehicles and what they can do with them.

```
500H 100M 82V > help vehicles
VEHICLES
```

In your travels around Eternity's Song, you may come across some objects that can be entered and driven around. These are "vehicles"; and could vary from something as obvious as a coach and horses, to a nuclear submarine.

"enter <vehicle>" will take you inside, while "leave" will take you back out - assuming there is an exit. Note that vehicle exits don't appear as normal room exits. There could be any number of rooms within the vehicle, and somewhere there should be some item that you can use to drive or control the vehicle. You can "drive <direction>" (or "pilot <direction>" if that seems more appropriate) and, provided the terrain in that direction is suitable for the vehicle, the vehicle - and all inside it - will move to that location.

You may also find another vehicle that you can drive your vehicle inside. The command "drive inside <vehicle>" (or into, or onto) will, if allowed, drive you into that vehicle. You can enter the command "drive outside" to leave that vehicle. To drive the new vehicle that you now find yourself inside, you will most likely have to leave your original vehicle (so you are just inside the outer vehicle, not inside the inner vehicle) and locate the controls for this outer vehicle.

To see what is outside the vehicle, use "look outside" or "look out <window>". However, you may not be able to see outside from all locations in the vehicle.

See also: ENTER, LEAVE, DRIVE, PILOT

```
500H 100M 82V >
```