

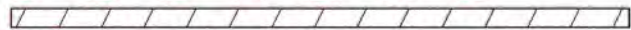
DUSKERS

STANDARD SALVAGE VESSEL
OPERATIONS MANUAL
V. A-03



(C) 2216 MISFITS ATTIC

DUSKERS



STANDARD SALVAGE VESSEL
OPERATIONS MANUAL

V. B-03

WARNING:

A very small percentage of individuals may experience epileptic seizures or blackouts when exposed to certain light patterns or flashing lights. Exposure to certain patterns or backgrounds on a monitor screen or when operating salvage drones may trigger epileptic seizures or blackouts in these individuals.

MINIMUM SYSTEM REQUIREMENTS

Bios: Salvage Vessel (Cargo Class) Version 5.1

Windows: XP / Vista / 7 / 8 / 10

Processor: SSE2 instruction set support.

Graphics: DX9 (shader model 2.0) capabilities; generally everything made since 2004 should work

DirectX: Version 9.0

Storage: 200 MB available space

Mouse, Keyboard

Mac: OSX 10.9+

Processor: SSE2 instruction set support.

Storage: 200 MB available space

Mouse, Keyboard

Linux: Ubuntu 14.04, Ubuntu 16.04

Processor: SSE2 instruction set support.

Storage: 200 MB available space

Mouse, Keyboard

CREDITS

RECORDS

Tim Keenan ...	Creator Guy (RANDOM IDEA GENERATOR, POINTER OF FINGERS)
Jeremy Phillips ...	Lead Programmer (BREAKS GAME, FIXES GAME)
Rick Sonderfan ...	Programmer (FATHER OF DRONES)
Brendan Mauro ...	Lead Artist (DRAWS PICTURES, RUNS SHIPYARD)
Jillian Ogle ...	Artist (VIEWS THE WORLD LIKE A DRONE)
Cale Bradbury ...	Effects (MADE IT (INTENTIONALLY) GLITCHY)
Ian Hicks ...	Audio (BLEEPS AND BOOPS)
Keith Moore ...	Audio (BLIPS AND BLOPS)
Robin Arnott ...	Audio (BUMPS IN THE NIGHT)
Benjamin Hill ...	Story (SPEAKER FOR THE DEAD)

INDIEFUND INVESTORS THAT MADE DUSKERS POSSIBLE...

Jonathon Blow	John Graham	Tommy Refenes
John Bizzarro	Cliff Harris	Jeff Rosen
Ron Carmel	Aron Isaksen	Kellee Santiago
Kyle Gabler	Colin Northway	Nathan Vella
Zach Gage	Sarah Northway	Matthew Wegner

ADDITIONAL HELP...

Holly Keenan ...	Pixels and Support
Travis Koller ...	Concept Art
Georgoe Cachrane ...	Mac & Linux Help
Ryan Paxton ...	Concept Animation
Brandon Surowiec ...	Help and Testing
Genevieve Duchesneau ...	Help and Testing
David York ...	Testing
Javier Ulloa ...	Testing

SPECIAL THANKS...

Double Fine
Intel
Duskers Council
Alex Austin
Ian Stocker
Family & Friends
Community & Fans

A. DRONE OPERATION OVERVIEW

4 Identification and control

B. BASIC COMMANDS

5	'A'	Open/close airlock(s)
5	'CLOSE'	Close specified door(s)
5	'D'	Open/close door(s)
6	'DEGAUSS'	Reset drone camera feed
6	'EXIT'	End mission
6	'FLAG'	Mark specified room(s)
6	'HELP'	Access help file
7	'INFO'	Report on items in a room
7	'NAVIGATE'	Auto-navigate drone(s)
7	'OPEN'	Open specified door(s)
8	'STATUS'	Report status of derelict
8	'SWAP'	Swap upgrades with nearest drone
8	'TIME'	Report mission time

C. ADVANCED COMMANDS

9	'ALIAS'	Access and edit alias file
9	'CLEAR'	Clear console
9	'COMMANDEER'	Commandeer derelict ship
9	'DESTRUCT'	Destroy derelict ship
10	'DOCK'	Re-dock boarding vessel
10	'PICKUP'	Collect dropped items

D. DRONE UPGRADES

11	Gather	
11	Generator	
11	Interface	
12	Lure	
12	Mine	
12	Motion	
12	Probe	
13	Pry	
13	Scan	
13	Sensor	
14	Shield	
14	Sonic	
14	Speedboost	
14	Stealth	
15	Stun	
15	Teleport	
15	Tow	
16	Trap	
16	Turret	

E. CONSOLE

17	Auto complete
17	Command chaining
18	Alias file

F. SHIP UPGRADES

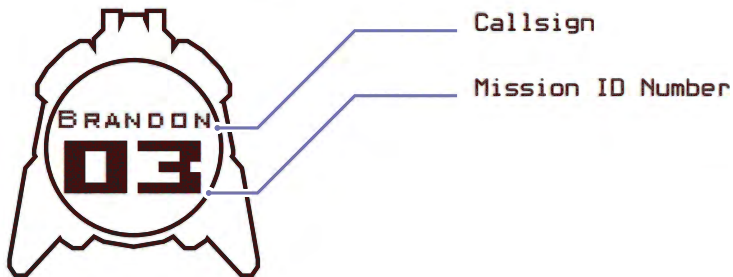
19	Cannon
19	Decontaminate
19	Overload
20	Remote power
20	Reroute
21	Transporter

G. EXTRA

22	Additional operations
----	-----------------------

A. DRONE OPERATIONS OVERVIEW

Identification and control



Switch Active Drone (via Mission ID Number)



Toggle view. Drone camera / Schematic overview



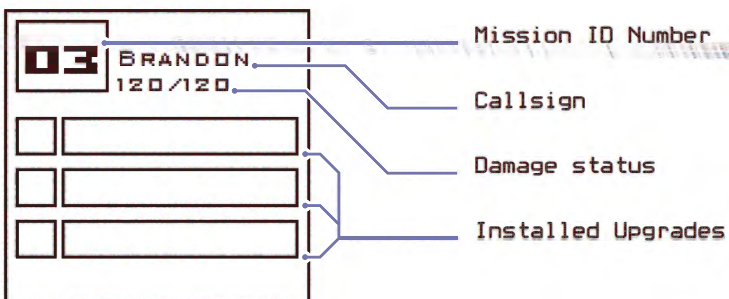
Drive Forward



Drive Backward



Rotate



B. BASIC COMMANDS

> 'A'

Toggle open/closed state of one or more specified airlocks, so long as they are powered.

```
Syntax: a<#>...
```

> 'CLOSE'

Explicitly close specified doors to specified room(s), so long as they are powered.

```
Syntax: close (d<#>... | r<#>... | all)
```

Usage:

```
close d1 d3    close door 1 and door 3 only,
                if powered
close r5        close all powered doors to room 5
close all      close all powered doors
```

> 'D'

Toggle open/closed state of one or more specified doors, so long as they are powered.

```
Syntax: d<#>...
```

Usage:

```
d2            toggles door 2 open/closed
d2 d3        toggles door 2 and 3 open/closed
```

Alternate usage:

```
open d2       open door 2 only if closed
close d2 d3   close door 2 and 3 only if open
```

> 'DEGAUSS'

Reset drone video feed, use to fix visual artifacting and corruption. Run command from drone view.

Usage: `degauss`

> 'EXIT'

Returns all drones in the docking bay to home ship.

Usage: `exit`

WARNING:

Any drones outside of docking bay will be lost.

> 'FLAG'

Toggle open/closed state of one or more specified doors, so long as they are powered.

Syntax: `flag [r<#>... | clear]`

Usage:

`flag r2 r3` toggles flag on/off for room 2 and 3
`flag clear` clears all room flags

> 'HELP'

Access salvage operations manual.

Syntax: `help [all | command]`

Usage:

`help` access help file
`help all` access help file
`help swap` display help on swap command

> 'INFO'

Report on items in a room.

TIP: Run 'info' on a fuel access point for report on derelict ships fuel capacity.

```
Syntax: info [drone]
```

Usage:

```
info          report on items in room
info 3       report on items in room occupied
             by drone 3
```

> 'NAVIGATE'

Auto-navigate drone(s) to a room or another drones location.

```
Syntax: navigate (drone)... [r<#>...]
```

Usage:

```
navigate 1 2 r3  navigate drones 1 and 2 to room 3
navigate 1 3     navigate drone 1 to drone 3)
navigate 4       navigate drone 4 to current drone
```

> 'OPEN'

Explicitly open specified doors or all doors to specified room(s), so long as they are powered.

```
Syntax: open (d<#>... | r<#>... | all)
```

Usage:

```
open d1 d3     open door 1 and door 3 only,
               if powered
open r5        open all powered doors to room 5
open all       open all powered doors
```

> 'STATUS'

Displays ship information determined from ship log and initial scans.

```
Usage: status
```

> 'SWAP'

Swap upgrades with nearest drone.

```
Syntax: swap [source_item target_item [target_drone]]
```

Usage:

```
swap (access swap interface)
```

```
swap 1 2 3 (Non GUI)
```

> 'TIME'

Reports elapsed mission time.

```
Usage: time
```

C. ADVANCED COMMANDS

> 'ALIAS'
Open alias file editor.

Aliases are powerful custom commands that can be used to abbreviate a command or series of commands.

i Refer SECTION E: CONSOLE for further reference.

> 'CLEAR'
Clear console window.

Usage: `clear`

> 'COMMANDEER'
Take command of derelict. Drone/equipment transfers to currently docked derelict.

- All rooms must be verified.
- Vessel must be free of infestations.
- Any radiation must be contained.
- No airlocks exposed to space.

Usage: `commandeer`

> 'DESTRUCT'
Set self-destruct timer on drone in seconds.
To cancel: Type 'destruct' again (quickly!).

Syntax: `destruct [t<1-60 seconds>]`

Usage:

`destruct` (default = 3 seconds, if unspecified)

`destruct t5` (self-destruct timer set to 5 seconds)

> 'DOCK'

Re-dock boarding vessel to specified airlock.

Syntax: `dock [a<#>]`

Usage:

`dock a2` (re-dock boarding vessel to airlock 2)

> 'PICKUP'

Pick-up dropped items within range.

Usage: `pickup`

'GENERATOR'
Use drone to power nearby power units, providing power to ship rooms.
Usage: `generator`

'INTERFACE'
Access nearby powered terminal and access interface commands.
Usage: `interface [list]`

'INTERFACE'
Access nearby powered terminal and access interface commands.
Usage: `interface [list]`
interface commands:
- `scan` (attempts to scan all powered comms)
- `toggle` (toggle powered turret on/off)
- `status` (shows turret ship layout)
interface list:
- `list` (list interface commands)
- `access` (access nearby powered terminal)

D. DRONE UPGRADES

- > **'GATHER'** (requires 25000 credits)
Collect fuel and scrap from occupied room.

Syntax: `gather [all | list] | [<drone>]`

Usage:

```
gather           (gathers nearby scrap and fuel)
gather all      (gathers all scrap and fuel in room)
gather 3        (commands drone 3 to gather all scrap
                  and fuel in room)
gather list     (lists how much scrap has been gathered
                  by this drone)
```

- > **'GENERATOR'**
Use drone to power nearby power inlet, providing power to ship rooms.

Usage: `generator`

- > **'INTERFACE'**
Access nearby powered terminal and access interface commands.

Syntax: `interface [list]`

Usage:

```
interface        (access nearby powered terminal)
interface list  (list interface commands)
```

Interface commands:

```
shipscan        (attempts to scan all powered rooms)
defence         (toggle powered turrets on/off)
survey         (shows derelict ship layout)
```

> 'LURE'

Drop a lure to attract biological infestations from nearby rooms.

```
Syntax: lure [drone] | [drop | pickup]
```

Usage:

```
lure                (drop or pickup lure)
lure drop           (drop lure)
lure pickup         (pickup un-used lure)
lure 2              (drone 2 drops equipped lure)
```

> 'MINE'

Drop a proximity mine.

```
Usage: mine
```

> 'MOTION'

Activates motion detector on stationary drone which scans adjacent rooms for movement.

```
Usage: motion
```

> 'PROBE'

Deploy a roving probe that moves from room to room scanning for items. If adjacent to a working probes, it can be retrieved with the 'pickup' command.

```
Syntax: probe [drone] | [drop | pickup]
```

Usage:

```
probe                (drop or pickup probe)
probe drop           (drop probe)
probe pickup         (pickup active probe)
probe 1              (drone 1 drops equipped probe)
```


> 'PRY'

Can be used to force open unpowered doors. Cannot be used to close doors.

```
Syntax: pry [drone] [d<#>]
```

Usage:

```
pry d2          (command drone equipped with pry  
                to open d2)  
pry 3 d4        (drone 3 to pry open d4)
```

> 'SCAN'

Scans the current room for undetected items.

```
Syntax: scan [drone]
```

Usage:

```
scan           (scan room with current drone)  
scan 2         (scan room with specified drone)
```

> 'SENSOR'

Drop a motion sensor which continually scans the room for threats.

```
Syntax: sensor [drone]
```

Usage:

```
sensor         (drop sensor at current drones location)  
sensor 2       (command specified drone to drop sensor)
```

WARNING:

once dropped, a sensor cannot be retrieved.

> 'SHIELD'

Shield upgrade is used to negate damage to drone. Using the 'shield' command will toggle the shield on and off. Recharges when inactive.

Usage: `shield`

> 'SONIC'

Activate a sonic pulse that repels organic threats. Sonic will deactivate if drone moves. Recharges when inactive.

Syntax: `sonic [drone] [on | off]`

Usage:

`sonic` (toggle sonic on and off)
`sonic off` (explicitly deactivate sonic upgrade)
`sonic 3 on` (explicitly activate sonic upgrade on drone 3)

SPEED BOOST UPGRADE

Drones can be upgraded with a speed boost modification for an increase in movement speed.

> 'STEALTH'

Toggles stealth field on and off. Prevents threats from detecting drone while active. Recharges when inactive.

Usage: `stealth`

WARNING:

taking damage or activating upgrades may deactivate stealth field.

> 'STUN'

Drop treat activated proximity stun device.

```
Syntax: stun [drone] [drop | pickup]
```

Usage:

```
stun                (deploy stun device)
stun 3              (deploy stun device from drone 3)
stun drop           (drop stun device)
stun pickup         (pickup stun device)
```

> 'TELEPORT'

Teleport drone or item to specified room. Destination room must have been previously scanned or visited.

```
Syntax: teleport [sensor | trap] (room)
```

Usage:

```
teleport r3         (teleport drone to room 3)
teleport trap r3    (teleport trap to room 3)
```

> 'TOW'

Tow equipped drones can move ship upgrades and disabled drones.

```
Syntax: tow [drone] [item_name]
```

Usage:

```
tow                (engage or disengage tow)
tow 2              (engage or disengage tow from drone 2)
```

> 'TRAP'

Drop remotely operated explosive. Detonated traps will destroy everything within a room.

```
Syntax: trap [drop | pickup] | [boom]
```

Usage:

```
trap                (deploy trap)
trap drop           (deploy trap)
trap pickup         (pickup deployed trap)
trap boom           (detonate all deployed traps)
```

> 'TURRET'

Deployed drone turret provides offensive fire within the drones frontal cone of vision.

```
Syntax: turret [drone] [on | off]
```

Usage:

```
turret             (toggle turret on and off)
turret on          (explicitly activate turret upgrade)
turret 3 on        (explicitly activate turret upgrade on
drone 3)
```

E. CONSOLE

AUTO-COMplete

The console will attempt to auto-complete the command being typed.

```
> navigate
```

Pressing [TAB] or [SPACE] will complete the command.

```
> navigate █
```

Pressing [ENTER] will complete and run the command.

```
> motion
  r3: results inconclusive
Motion Sensors Activated
> █
```

COMMAND CHAINING

Semicolon [;] can be used to execute more than one command on a single command line.

```
Example: open d5; navigate 1 r4; navigate 3 r7
```

ALIAS FILE

The alias file can be used to create and store both simple user-defined abbreviated commands and complex command aliases.

> 'ALIAS'

Open alias file editor.

Your alias file is preconfigured with the following 2 aliases that demonstrate the alias command structure.

```
===== ALIAS EDITOR =====
begin=open a1;navigate 1 2 a1; generator 2; status
end=navigate 1 2 3 4 r1
|
[ESC] CANCEL           [CTRL + S] SAVE           [ALT + U] VALIDATE
```

> 'begin'

The begin command will open airlock 1, move drones 1 and 2 into the room beyond airlock 1, command drone 2 to power the power inlet of present room then list status of the derelict ship.

> 'end'

Execute the 'end' command to direct all active drones to room 1 (r1).

ADD ALIAS FROM COMMAND LINE

New aliases can be added to the alias file directly from the command prompt:

```
> alias end=navigate 1 2 3 4 r1
```

F. SHIP UPGRADES

> 'CANNON'

When installed, the ships cannon can target a room.

```
Syntax: cannon (r<#>)
```

Usage:

```
cannon r4 (target and fire ship cannon at room 4)
```

> 'DECONTAMINATE'

Clears radiation from a room.

```
Syntax: decontaminate (r<#>)
```

Usage:

```
decontaminate r4 (removes radiation from room 4)
```

> 'OVERLOAD'

Affects found electronics in the target room and damages surroundings.

```
Syntax: overload (r<#>)
```

Usage:

```
overload r4 (overload electronic equipment in room 4)
```

WARNING:

```
high chance of equipment failure.
```

> 'REMOTE POWER'

Powers a known power inlet remotely (w/o a drone).

```
Syntax: remote | power [r<#>]
```

Usage:

```
remote r4      (powers a power inlet in room 4)
power r2       (powers a power inlet in room 2)
```

> 'REROUTE'

Allows derelict ship inlet power to be rerouted to adjacent rooms. Subsequently powered rooms must be connected to the first room.

```
Syntax: reroute [r<#>] [r<#>...] | [status]
```

Usage:

```
reroute r2 r4 r6 r7 (reroute power from inlet in
                    room 2 to rooms 4, 6 and 7)
reroute r2 status   (displays number of rooms that
                    can be powered by the inlet in
                    room 2)
reroute r2 -r6      (powers down room 6 if
                    connected to room 2)
reroute r2 +r6      (powers up room 6 if
                    connected to room 2)
```

WARNING:

a power inlet cannot exceed its capacity.

> 'TRANSPORTER'

- Transports drone(s) to and from Outposts/Derelicts. Destination room must have an active signal (indicated on schematic view). Room to room transport will occur through docking bay.

Syntax: `transport [(drone... | all) [r<#> | drone]`

Usage:

```
transport 1 r2      (transport drone 1 to room 2)
transport all r2    (transport all drones to room 2)
transport 2 1       (transport drone 1 to drone 2's
                    room - must have transport
                    receiver)
```

G. EXTRA

COMMAND CHAINING

- [;]** Semicolon [;] can be used to execute more than one command on a single command line.
> open d5; navigate 1 r4; navigate 3 r7

ADJUST CONSOLE DISPLAY

- CTRL + +** Increase the console text size.
- CTRL + -** Decrease the console text size.
- FB** Toggle size of the console window.

CLEARING COMMANDS

- CTRL + C** Clear the input line of a partially typed command from the command line.
- CTRL + U** Clear everything before the cursor position on the command line.
- CTRL +** Delete the previous word from the command line.
BACKSPACE

CONSOLE NAVIGATION

- CTRL + A** Jump cursor to the beginning of the command line.
- CTRL + E** Jump cursor to the end of the command line.
- SHIFT + ↑** Page up in console window.
(Alternative key: **SHIFT + PGUP**)
- SHIFT + ↓** Page down in console window.
(Alternative key: **SHIFT + PGDN**)

COMMAND HISTORY

- CTRL** + **↑** Move backwards through the console command history.
- CTRL** + **↓** Move forwards through the console command history.
(Can be used to clear the input line of a partially entered command.)

SCHEMATIC VIEW OPERATIONS

- HOME** Centers the schematic view.
- I** Toggle schematic view icons on/off.

