Contents

D E S C E N T M I S S I O N B U I L D E R I I

Introduction

D E S C E N T M I S S I O N B U I L D E R I I

What it does

Descent Mission Builder II is a utility used to edit and create new levels for the registered version of the game Descent and Descent II by Interplay Productions.

Version

This is the registered version of the Descent Mission Builder II. This version works with the registered (not shareware) version of Descent and Descent II. It does not require any files from the game except the *.pig files from Descent and Descent II which must be present in order to view the texture graphics.

The Descent pig file must be version 1.4a or later. If you have an earlier version, you should run the upgrade patch on the game Descent. This patch is located on the distribution disk(s) of Descent Mission Builder II (see the readme.txt file for more information).

The Descent II pig files should be stored on the hard drive. You should use the "medium" (or above) option when installing the game in order to get all of the pig files onto the hard drive. The Descent Mission Builder II does not read the Descent II CD, nor is this CD required to be inserted in the CD when running the Mission Builder.

Notices

Credits People who contributed to this product.

What's New

FILES

Hog Manager
The new Hog Manager allows you to import, extract, delete, and rename files in a hog file. You no longer have to get a 3rd party hog utility just to edit hog files.

You can also save you level directly into any hog file. You have the choice of replacing the entire hog file or adding to it. It doesn't get any easier than this.

You can also open non-level files (those without the *.rdl or *.rl2 extension). The mission builder will extract the file to a user chosen location then try to launch a program "associated' with the extension. For example, *.pcx files loads a paint program. If you want to edit these files, you will need to import them back into the hog file when finished.

Mission Manager
The new Mission Manager allows you make multiple level missions. It also allows you to add secret levels and the new extended fields to the mission file. There is a place for your name, the date, and several other useful fields. This is intended to help organize the vast quantity of levels found on web pages.

Startup
When the mission builder first starts, the last rdl or rl2 file will be loaded. This only happens if you are editing the rdl/rl2 file by itself (not within a hog file). So, if you like this feature, then edit rdl or rl2 files separately.

Safety Features
Asks "Are you sure?" if changing pig when using custom textures

Detects changes made to a level and asks you if you want to save your changes when you exit.

Warns user if they have loaded a level which contains a custom texture file (*.pog) which has more than just cube textures in it. The POG file format actually permits changes to any texture. The mission builder only allows changes to cube textures. You can still edit a level which has an extended POG file. Simply press "Cancel" when asked if you want to save the custom textures.

You can no longer save to descent.hog or descent2.hog.

TEXTURES

Shaded Texture Mapping
The View menu now offers the option to render the level with shaded textures. This allows you to see the lighting adjustments prior to play testing you level.

High Resolution Mode
You can now switch between low resolution and high resolution with a single key press. Use the low resolution setting to speed up the screen updates on slower machines.

Moving Textures
Textures that move now have a thin white arrow drawn on them when viewing the texture mapped level. They also move in the texture dialog so you can check the speed.

Texture Alignment
Textures are no longer reset when you move them. This was necessary to prevent from loosing alignment changes just because a side was moved slightly. The down side is you now have to reset the textures prior to aligning them if you want them to be at the correct scale. For this reason, a Reset Marked button was added to the Align dialog.

Better Custom Texture Support
The Import bitmap feature now allows data to be imported without remapping any colors. This is useful when creating transparencies (color #254 and #255) which use to be remapped to another color. All you have to do is make sure you use the same palette as the level. You can get this palette by saving any texture to a file, then use that file as the base for your modifications.

The import feature now supports 16 color bitmap files in addition to the normal 256 color bitmaps.

Favorite Textures
The texture picker now has a button which allows you to see "only" the texture used in the level. This allows you to quickly find the texture you want without scrolling through several hundred textures. As before, you can use the left mouse button to select Texture1 and the right mouse button to select texture 2.

I also managed to get rid of those silly white pixels that some people were experiencing on the Texture Picker dialog.

Pasting Multiple Textures
Added "Paste Marked" cubes to Texture dialog. This allows you to change the texture of multiple cubes at once.

EDITING

Point Marking
Instead of just marking cubes, you can now mark one or more points. The letter 'm' will either mark a point, line, side, or cube depending on what the editing mode is set to. You can also use the mouse to draw a box around a bunch of points. Then, switch to block mode to move them all at once!

New Check Boxes in Preferences
"Show point coordinates instead of cube dimension' - This changes the information displayed on the lower tool bar.

"Show cross hairs instead of globe" - After you get the hang of moving the view around, you can hide the globe with this option.

"Hide marked cubes & points unless in block mode" - This is particularly useful when lighting an entire level at once.

Point Editing
By popular demand, you can now edit the coordinates of every point in the level. Several people have thanked me for adding this feature, but I'm still not sure exactly why. After all, if you like editing data, then you can always use a hex editor on the level file :-)

Undo
There is a single step undo command which works when you move one or more points around. It doesn't work for deleting, joining, or object editing, but it's better than nothing.

More Tool Bar Functions
You can now enlarge or reduce marked blocks. Be careful with this feature, because it doesn't move the objects.

Block Cut & Paste Auto-Join Feature and Quick Paste
When you paste in a block file, sides which had children when the block file was created will be joined to cubes in the level. This is particularly useful with pasting in a "slice" or a room.

There is also a way to quickly paste in the last block file. Just press "Shift+Ctrl+Ins".

Join / Unjoin Current Sides
Two new join/unjoin functions have been added - Join Current Side allows you to "punch out" a cube side wall between two cubes. Unjoin Current Side does just the opposite.

Mode Selection via Arrow Keys
Arrow keys no longer change you to "side" mode. If you want to change modes and sides at the same time, use the letter S instead.

OBJECTS

Vertigo Robot Support
Now you can finally create levels which contain the Vertigo Series robots.

Advanced Object Editing
The object dialog has a new button on it called Advanced. This pops up a dialog with a few interesting fields in it. Most of these fields are ignored by Descent, but some are not. This feature was added for the advanced user who wants to tweak the objects to come up with some interesting results. Little is known about these fields, but you are welcome to explore the possibilities.

Robot Editor Changes
HXM files have been reduced in size. They now only contain the modified robot characteristics instead of all the robot characteristics.

The robot editor now has a new option: firing_wait2. This is used to set the time delay for weapon 2.

The max score has been changed from 250 to 30,000.

Changed robot edit's Big Explosion to a slider instead of a button. This gives you greater on how much damage robot does to your shields after it explodes.

Objects and Secret Return
You can now edit the orientation of the secret return object from the object dialog. It will always be the last object on the list. Note that you can move this object from one cube to another but you cannot move it from the center of the cube. This is a limitation of the game, not the editor. Note: The secret return object will not appear unless the level contains a secret exit.

The new reset button allows you to set the object to its default orientation. This was added primarily to fix a bug with the secret object's orientation matrix which made it look funny when you return from a secret level.

Highlight Colors and Objects
Some colors were changed a bit. Before some of the objects were colored red (robots and reactors). But, red has now been reserved for editing. So robots were changed to purple and reactors were changed to orange. So, now when something is red, you can move it with the tool bar.

Robot Wireframes
The current object is now drawn using the polymodel data found in the DESCENT2.HAM file. This allows you to see the shape and size of the robot instead of just an arrow head. If there is no polymodel data for the object, then it is drawn as a thick arrow. This is the case for powerups, hostages, Descent 1 objects.

LIGHTS

Blinking / Exploding Lights
Reduced the level size by only saving the necessary blinking and exploding light values.

There is even a "Lights On/Off" button on the light adjustment dialog which shows you what it looks like with all the blinking lights off and all the exploding lights broken. This dialog is also modeless which means you don't have to close it when you want to do something else.

Another feature you might want to take advantage of, is the ability to turn off a single blinking light. To do this, just select the side with the blinking light on it and it will turn off!

Texture Lighting
The texture lighting has changed dramatically. The previous version assigned a value to every texture type and used these values to calculate the lighting. The new version uses the actual light values that Parallax used to light their levels. Now, only textures which end with " - light" give off light.

The lighting algorithm has also changed. You now have control of the brightness to light the level. The results are amazing.

VIEW

Reset View
Tired of positioning the level just right? Now you can align to the current side so it lies in the plane of the screen by pressing Alt+Home. Try it!

MISCELLANEOUS

Bug fixes
There were two significant bugs in the original Descent Mission Builder II that were fixed:
1) Deleting a cube no longer moves blinking lights on higher cubes.
2) Pressing the Esc key when the About screen shows up no longer disables saves.

Electronic Registration
Removed electronic registration, Fixed some spelling errors, Enhanced check function,
Reduced the size of most of the dialogs, Fixed robot attributes dialog, Added door08 and wall01 to wall clips.

Hot keys
The "Copy other cube Texture" hot key F6 has now been changed to Ctrl+F6 to help avoid accidentally pressing the button. The F5, F6, F7, F8, and F9 keys all effect the view mode.
Also, the caps key is now ignored when selecting hot keys.

Credits

FOR BRAINWARE:

Programming:
Bryan Aamot

Design:
Bryan Aamot

Art Design:
Bryan Aamot

FOR INTERPLAY:

Executive Producer:
Alan Pavlish

Produced by:
Raphael Hernandez

Line Producer:
Kihan Pak

Director of Quality Assurance:
Jeremy S. Barnes

Assistant Director of Quality Assurance:
Darren "Gothmog" Monahan

Administrative Assistant:
Dave "Natas" Simon

Quality Assurance Technical Coordinator
John Werner

Quality Assurance Technicians
Bill "So-Lo" Delk
Aaron "Saut" Meyers
Steve Cabiness

Compatibility
Dan Forsyth
Derek Gibbs
Jack Parker
Phuong "Norman" Nguyen
Aaron Olaiz
Marc "Psycho" Duran

Lead Testers
Jason "Ryo-ohki" Nordgren
Cory "Faerl" Nelson
Darrell "I am the Maestro" Jones
Chad "Boogie" Allison

Upgrade Patch Testers
Chad "Boogie" Allison, Director of Quality Assurance
Cory Nelson, Lead Tester
Scot Humphreys
John Stavros
Tymothi Loving

Testers
Rene Hakiki
Erick "Valhuru" Lujan
Bill "So-Lo" Delk
Chris "Damon" Peak
Henry "Totoro" Kahng
Greg "Moose" Baumeister
Anthony "Sinlover" Taylor
Steve "Nirvana" Baldoni
Dave "Lazwik" Hendee
Erick "Valhuru" Lujan
Derek "Cobra" Bullard
Chris "Chaos" Keenan
Doug "Wolf" Avery
Daniel Huffman
Matt "Drake" Golembiewski

Special Thanks to:

Parallax	for making such a great game.
Jessica Mulligan	for her belief in me and the product.
Mike Kulas	for your responses to my e-mails.
Matt Toschlog	for all the special updates to Descent2 to support the Mission Builder.
Nightshift	for supplying a beta site area in the Compuserve Game Publishers Beta forum.
Steffen "Cyborg" Reinhart	for doing the additional work required to maintain Beta forum.

Extra special thanks to the CompuServe and Internet beta testing team:
Banning "Spaz" Andrews
Doug "Blaster" Blair
Mac Broughton - Action Games forum sysop
Jason "Sub Zero" Castle
Bill "Ice" Hamilton
Craig Norton
Jim "Shooter" Reed
Dan "Ebola" Wentz
Kevin Lee
Nate "Grim Fish" Yost
Timothy "Sparhawk" Rhodes
Bradley Schneider
James "SamuraiCat" Webster
Mycol "Rock" Wahnee
Mike "Gooch" Toguchi
Mike "KoolBear" Sullivan
David "Dracula" Graser
John "Scorpio" Bell
Ben "Lupo" Fisher III
Patrick "Snowpepsi" Talaczynski
Colin "Castig8r" Krawcyk

John "Cyberchill" Berg III
Dustin Riggs
Daniel Santos
Christopher "Lava" Hynes

Level Structure

Descent levels are composed of cubes connected together to make the structure of a mine. When you play the game, you fly inside of the cubes. But, when you view the mine from the editor, you see the cubes from outside. This is similar to the AUTOMAP feature of the game.

Each cube has 6 sides. Each side has one or two textures. Textures are put on all sides which are not joined to other cubes

Each cube has 8 corners. These corners have (x,y,z) coordinates which are called vertices.

When a cube shares four corners of a side with another cube, they are considered to be joined. Several cubes joined together form tunnels and rooms. Each cube in a level is joined with one or more cubes.

Cubes contain objects, walls, triggers, and lights.

Pieces of a level

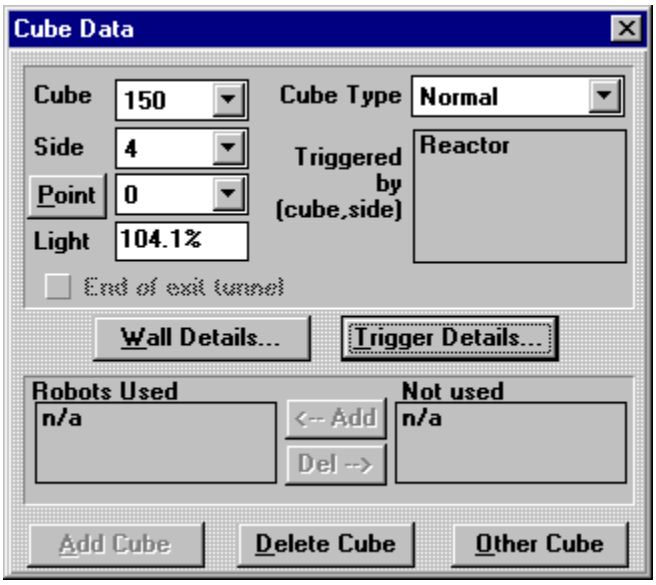
Cubes	Cube parameters.
Objects	Items, hostages, and enemies.
Walls	Doors, Illusions, Walls, and a place to put a trigger.
Triggers	Fly through these to activate doors or other tricks.
Reactor Triggers	Events that occur when the reactor blows up.
Exploding Lights	Automatic exploding lights.
Blinking Lights	Blinking lights.

Miscellaneous

Limits	Maximums and minimums.
Rules	Rules to prevent levels from aborting the game.
Guidelines	Guide to professional looking levels.
Tricks	How to pull a fast one on players.

Cubes

Cube Dialog



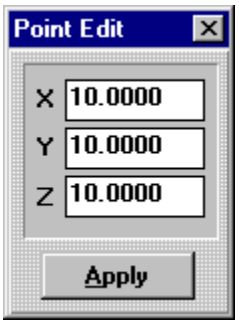
The two cubes you need to learn about first are the **Current Cube** and the **Other Cube**. The current cube's attributes are shown in the cube dialog. If you want to switch to the **Other Cube**, press the **Other Cube** button. Most editing functions use the **Current Cube** only, but there are a few commands (such as "join" and the "curve generator") which use both the **Current Cube** and the **Other Cube** .

Cube, Side, Point

These three list boxes allow you to choose the current cube, side, and point. You can also use the letters C, S, and P to select the next cube, side, and point. Use the shift key with these letters to go to the previous one.

Point Editing

The Point button brings up a point edit dialog.

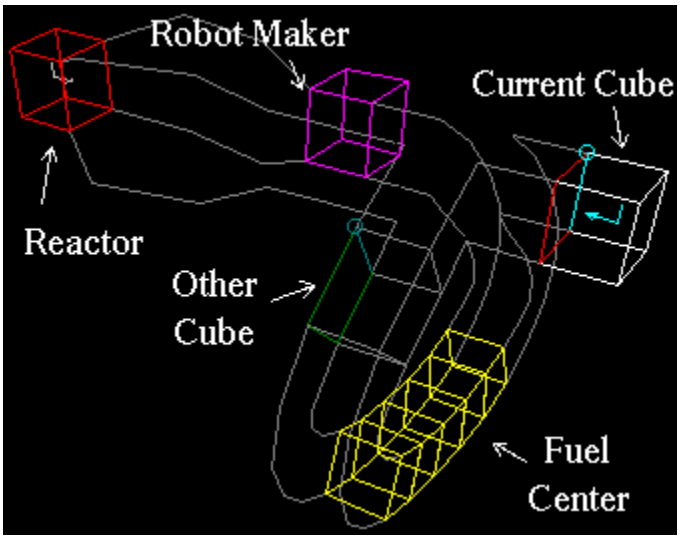


Cube Types

Cubes may be one of the following types:

Normal - Most cubes are of this type

Fuel Center - Increases your ships energy to a maximum of 100.
Robot Maker - Cubes where robots are created.
Reactor - The cube containing the control center of the mine.
Blue Goal - The cube where the blue team brings the red flag (Descent 2 only).
Red Goal - The cube where the red team brings the blue flag.(Descent 2 only)



End of exit tunnel

There should be exactly one of these per level. This box should be checked on the last cube in the exit tunnel on the side which faces out.

Cube Light

And last but not least, cubes have a "light" value. This value defines how bright certain types of objects will appear when they are in the cube. These objects include all of the "polygon rendered" objects (e.g. robots, other players, and the reactor). The brightness of side textures, power ups, hostages, and mines are not effected by this value.

Contents of a cube

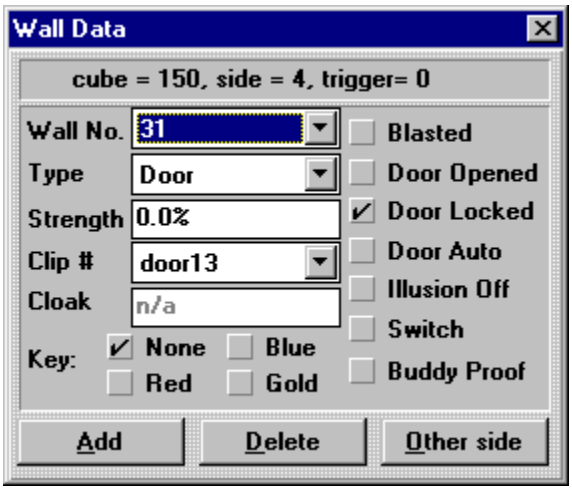
Cubes may contain one or more of the following:
Objects - (e.g. robots, powerups, mines, etc..)
Doors - (e.g. automatic, locked, exit, etc..)
Illusions - (like the energy field in fuel centers).
Triggers (e.g. sets off robot makers, opens doors, ends level, etc..)

The way the game defines doors and triggers is a little confusing to the beginner. But, it is very important that you understand the relationship between cubes, doors, and triggers.

For more information on cube walls, click here
For more information on cube triggers, click here

Walls

Walls Types



Each side of a cube that joins with another cube is allowed to have a wall. You can tell if a cube has a wall because the editor will mark the side with an 'X'. There are several types of walls:

Door	standard door, exit, secret exit, locked door.
Blastable	contains a prison door or another door animation.
Illusion	can hold any texture.
Open	used for triggers when you don't want to see anything.
Close	door which can never be opened.
Normal	wall which blocks two walls.
Overlay	control panel switch used to unlock/open doors (Descent 2 only).
Cloaked	dark tinted glass (Descent 2 only).

Note: Cloaked walls will not display textures. You can shoot thru a cloaked wall if the "empty" texture or a "see thru" texture is used. If a solid texture is assigned, it's impermeable.

Wall Strength

Blastable walls require a strength value. This value should be set to 100% for normal results. You can set this value to higher or lower to make Blastable walls tougher or easier to blast through.

Wall Clip

A wall clip (short for animation clip) is used on walls with type equal to Door. This tells the game which animation to show when the door is opened or closed. For example: door13 is the exit door (how lucky).

When selecting a clip number, Descent Mission Builder II automatically selects the texture of the coordinating wall for you. But, you can adjust the texture on the wall to hide the door until it is open for the first time. You can do this by changing the texture using the Texture dialog **after** you set the clip.

Wall Keys

If the wall is of type Door, then it can also require a key to open it (in single player mode or cooperative player mode). The choices are:

None	No key required.
Blue	Requires a blue key to open.

Red	Requires a red key to open.
Gold	Requires a gold key to open (commonly called a yellow key).

Wall Flags

Wall flags are used to specify certain attributes about walls of type Blastable, Door, and Illusion.

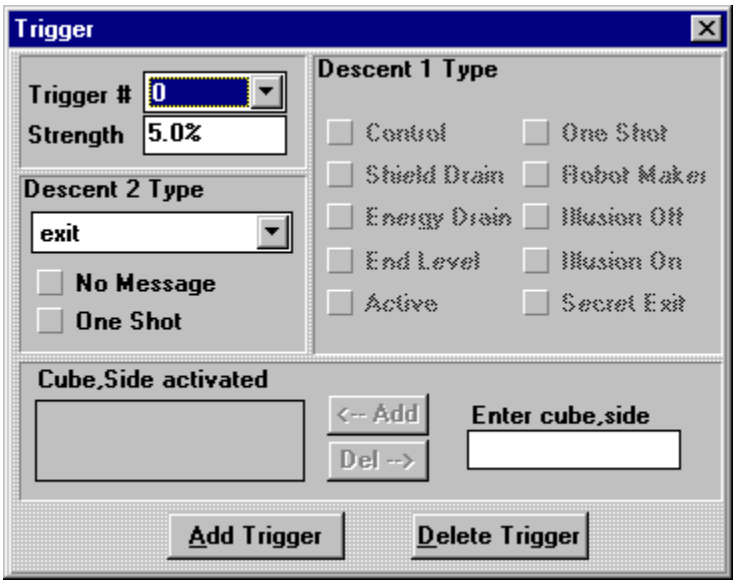
Blasted	Makes a Blastable wall start off that way.
Door Opened	This means the door starts out open and stays that way.
Door Locked	This door can not be opened by a player. It can only be opened by a trigger or when the reactor blows up.
Door Auto	This makes a door automatically close after it is opened.
Illusion Off	Sets the illusion off. Use a trigger to turn it on.
Switch	Control Panel used to unlock/open doors (Descent 2 only).
Buddy Proof	Prevents Guide-Bot from entering this cube (Descent 2 only).

Other Side

Use this button to select the cube opposing side of the current cube. This is useful for adding double sided walls or comparing walls.

Triggers

Triggers



If a side contains a wall, then it may optionally contain a trigger. You can tell if a wall has a trigger because it will have an arrow pointed in the direction you have to fly through (or shoot) the trigger in order to trip it.

NOTE: Triggers only work in one direction, so if you fly through the trigger in the opposite direction, it will not do anything.

Triggers are used to cause certain events to happen:

For Descent 1, you can have **one or more** of the following types:

End Level	normal exit.
Control	opens a door on another cube.
Robot Maker	activates a robot maker cube.
Secret Exit	secret exit.
Shield Drain	drains your shields.
Energy Drain	drains your energy.
Illusion Off	turns off an illusion type of wall on another cube.
Illusion On	turns on an illusion type of wall on another cube.
One Shot	(reserved for future use)
Active	(reserved for future use)

Every mine should contain one **End Level** trigger. Any other triggers are optional.

In Descent 1, you can combine trigger events by selecting more than one type. For instance, you could make a trigger that opens a hidden door and drains your shields at the same time. Although this would be a pretty nasty trick to play on other players. Perhaps you could put some shield boosts just out of reach to give them a fighting chance <grin>.

For Descent 2, you can pick from **only one** of the following types:

open door	opens a door on another cube.
close door	closes a door on another cube.
matcen	activates a robot maker cube.
exit	normal exit.
secret exit	secret exit.
illusion off	turns off an illusion type of wall on another cube.
illusion on	turns on an illusion type of wall on another cube.
unlock door	unlocks a door on another cube.
lock door	locks a door on another cube.
open wall	enables wall
close wall	disables wall
illusonary wall	turns a wall into an illusion
light off	turns off an light.
light on	turns on an light.

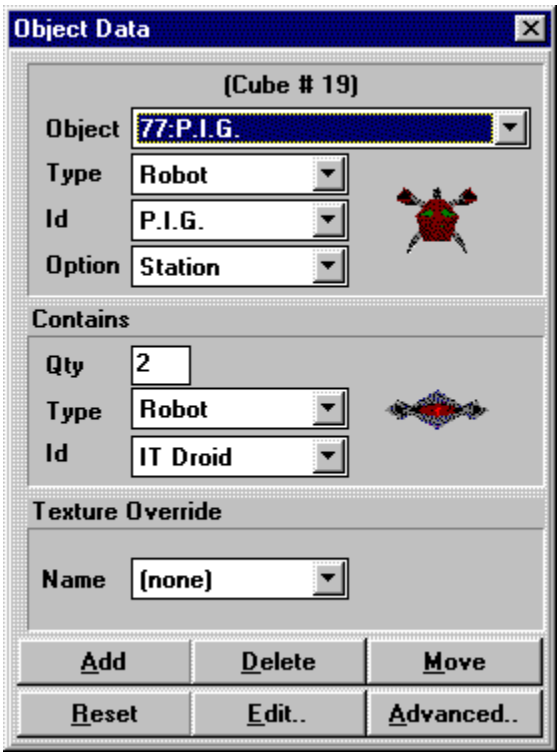
In Descent 2, you can only pick one of the trigger types from the list box. They can not be combined.

Hints

Triggers give a level a life of its own. You can create traps, puzzles, or tricks with triggers once you get the hang of it.

Objects

Object types



Objects are grouped by types:

Robot	These objects try to prevent you from achieving your goal.
Hostage	These are the guys you are trying to rescue.
Player	This is you or one of the other players in a multiplayer game.
Mine	This is a red mine which explodes when you get near it (Descent 2 only)
Powerup	Includes extra life, invulnerability, shield boost, energy boost, keys, lasers, cannons, bombs, missiles, and cloaking devices.
Reactor	The main reactor.
Coop	Cooperative player.

A cube may have as many objects as you like (up to the maximum number of 350 objects). However, it is not recommended that you place more than 5 in one cube.

Recommended number of objects

Robot	None required (100 typical).
Hostage	None required (4 typical).
Player	1 required for single player game, 8 required for multiplayer game.
Mine	Descent 2 only, None required (15 typical).
Powerup	None required (75 typical).
Reactor	1 required (you can put more but you may get strange results).
Coop	3 required for cooperative games.

Object Id

The Id (short for identification) specifies the type of object.

Robot	One of twenty four different kinds of robots.
Hostage	You should number one of the objects 0 and the rest 1 (although it does not seem to make any difference).
Player	8 players numbered 0 through 7.
Mine	Only one type one "Mine" (Descent 2 only)
Powerup	One of twenty two different kinds.
Reactor	0 for Descent 1, 1 through 6 for Descent 2.
Coop	3 CoOp players numbered 8, 9, and 10 (again, it does not make any difference how these are numbered).

Object Contents

Robot objects may contain between 0 and 99 robots or powerups. Normally you should only put a quantity of 1 to 3 objects, but it sure makes for a spectacular effect to load a robot with 99 blue shield boosting powerups.

Object Textures

Descent allows you to override the textures of any object which is drawn as a series of polygons. These types of objects are called "Poly Objects". Poly Objects include: Robots, Reactors, and Players. To change the texture of Poly Objects, selected the object then press the "Poly Object" check box on the texture dialog. Then you can select any texture from the Texture1 list including animations which make for some interesting results (try lava02-anim, or fan01-anim). Note: Texture 2 is not used for Poly Objects.

Edit Button (Descent 2 version 1.2 or above)

Pressing the 

buttons allows you to edit various aspects of a robot. The mission builder will automatically detect if any robots have been "modified" and prompt you when you save the level (into a hog file) to see if you want to save the modifications.

The mission builder will create a special file called a HXM (extended ham) file and store it in the HOG file you create when you save the level. Descent II version 1.2 (or above) uses this file to replace the attributes of the robot(s) you have modified. For earlier version of Descent II, this file is ignored.

For more information about editing robots, see the ROBOT WORKSHOP section.

Move Button

Pressing the 

button allows you to move an object from one cube to another.

Reset Button

Pressing the 

button will reset the object's rotation.

Advanced Button

Pressing the 

button will bring up the Advanced Object Data dialog.

For more information about advanced object editing, see the ADVANCED OBJECT DATA section.

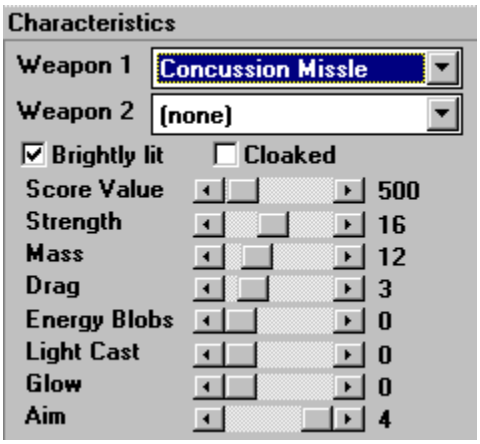
Robot Workshop (Descent II only)

Note: *Requires version 1.2 or greater of Descent II.*

Characteristics

The robot workshop is an advanced feature but easy enough for even a novice to use. Most of the values listed along side the list boxes are in "magic number" units. In other words, they don't use real metrics like "pounds" and "feet/sec". You will simply have to play around with these values until you get the desired results.

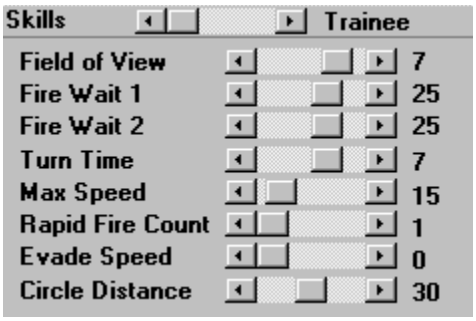
Hint: Some of these values have special meaning when they are set to 0. So, don't assume that 0 is less effect than 1.



This controls the physical attributes of a robot.

Weapon 1 type	Type of primary weapon.
Weapon 2 type	Type of secondary weapon.
Brightly lit	Whether cube light effects them (I think?). Always checked.
Cloaked	Makes robot nearly invisible
Score Value	How many points you get if when it is destroyed
Strength	How hard it is to destroy this robot (1=easiest, 0=can not destroy)
Mass	How much the robot weighs
Drag	Physical property used by the game engine.
Energy Blobs	How it converts laser fire to those green homing blobs. (0=none)
Light Cast	Unknown
Glow	Unknown
Aim	Aiming accuracy of the robot

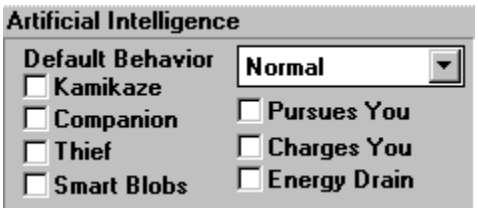
Skills



These are attributes which change depending on the players skill level. There are 5 sets of attributes, one for each skill level. If you adjust an attribute for one skill level, you should adjust it for all the other skill levels.

Players skill	0=Trainee, 1=Rookie, 2=HotShot, 3=Ace, 4=Insane.
Field of View	Angle at which it can see you
Fire Wait 1	How long it waits until it shoots weapon 1
Fire Wait 2	How long it waits until it shoots weapon 2
Turn Time	How long it takes to turn
Max Speed	The maximum speed at which it moves
Rapid Fire Count	How many projectiles it releases when firing
Evade Speed	How fast it evades your shots
Circle Distance	How far away it tries to circle around you

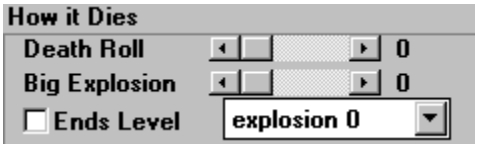
Artificial Intelligence



These check boxes effect the robot's intelligence.

Kamikaze	Rams you and blows up if it has a weak shield.
Companion	Only a Guide bot uses this.
Thief	Steals from you or not
Smart Blobs	Has smart blobs or not (in other words, I'm not sure).
Pursues You	Ya know, chases you.
Charges You	Get your credit card ready, well maybe not.
Energy Drain	Drains your energy (dah).

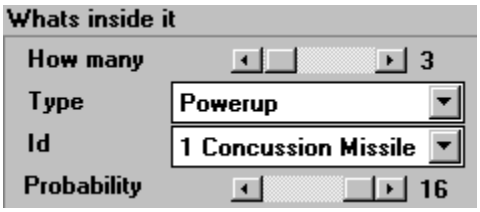
How it Dies



Death Roll	How many times it spins before exploding.
Big Explosion	How much damage object does when it explodes.

Ends Level	Also known as a Boss robot.
Explosion graphics	Animation clip shown when it explodes.

What's in it

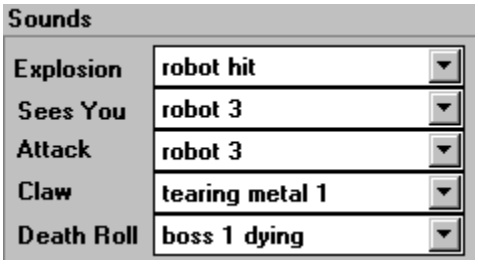


This is the default contents of a robot. Ever notice how some robots squirt out a little present for you when they are destroyed, but not always. Well, this is the setting which adjusts these parameters.

How many	quantity of items it can carry
Type	Either a powerup or a robot
Id	ID of powerup or robot
Probability	Odds that the item(s) will release.

Note: You can over ride the quantity, type, and ID on a robot by robot basis in the Object Dialog. This way, if you want one particular robot of a given type to be different than all the rest, you can adjust it.

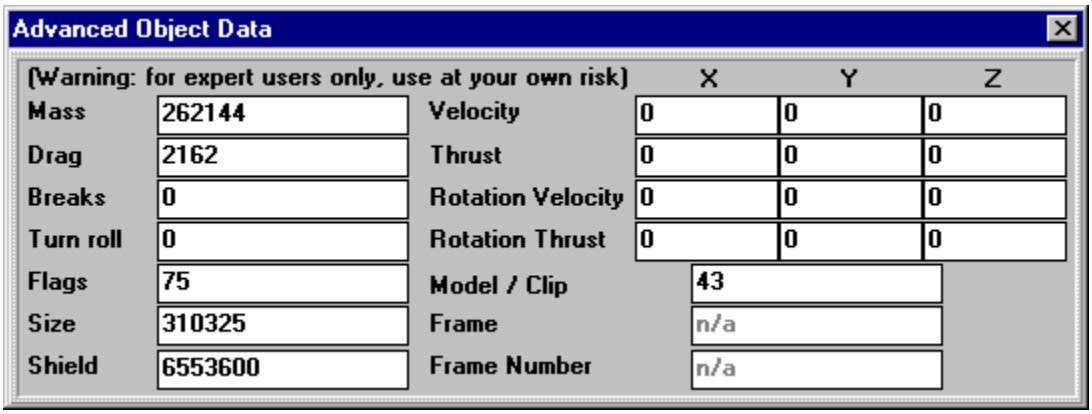
Sound



Each robot type has several sounds that it can make (depending on its characteristics). Use these list boxes to adjust the sounds.

Advanced Object Data

Warning: This is for expert users only, use at your own risk.



Data

Not a lot is know about this data except that most changes have no effect on the game. But, this data is left here for the user to play with on the outside chance that you may actually be able to affect the robot. This data only pertains to the current object and does not affect other objects.

Mass	The mass of this object
Drag	How fast this slows down
Breaks	How much brakes applied
Turn roll	How much it banks when it turns
Flags	Misc physical flags
Size	The size of this object
Shield	How difficult it is to destroy this object
Velocity	Velocity vector of this object
Thrust	Constant force applied to this object
Rotation Velocity	Rotational velocity (angles)
Rotation Thrust	Rotational acceleration
Model / Clip	Robot model number / Powerup clip number
Frame	Frame time
Frame Number	Starting frame number

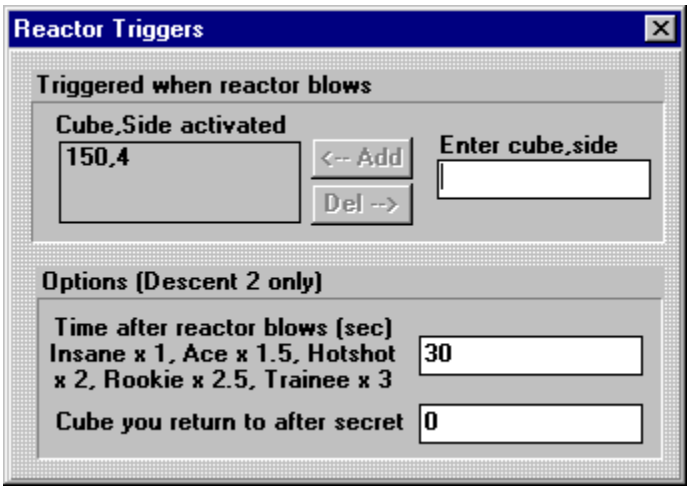
Reactor Triggers

What the heck are these?

Reactor Triggers are simply a list of doors which open when the reactor blows up. For example, the exit door opens after the reactor blows up to give you a way out.

Adding Reactor Triggers

Reactor triggers are automatically added whenever you make an exit door or when you make a reactor type cube. If you want to add additional doors which trigger when the reactor blows up, select *Reactor Triggers* from the *Tools* menu.



Enter in the cube and the side which has the door and click on the Add button. For example, if you want robots to block the mine exit only after the reactor blows up, you can create a locked door which opens only during the escape count down. Lets say the door is on cube 53 side 2, then you would type 53,2 in the **Enter cube,side** edit field and click on Add. If the number was not a valid cube or a valid side, then it will not be added to the list. You can have up to 10 cube/sides in the list.

Descent II options

Descent II allows you to change the number of seconds you have to exit the level after the reactor blows up. The actual number of seconds is a function of the difficulty level. If you entered a time of T seconds, the actual time will be as follows:

Difficulty Setting	**Time to escape**
Trainee	3.0 x T
Rookie	2.5 x T
Hotshot	2.0 x T
Ace	1.5 x T
Insane	1.0 x T

It also allows you to specify which cube you will return to after you return from a secret level. This number is automatically set to the current cube when you insert a secret exit from the Insert menu. But, you can adjust it here if you want to appear somewhere else.

Limits

Maximums and minimums

There are certain limits to what Descent™ can handle. The Descent Mission Builder II allows you to make levels all the way up to the maximum capabilities of the game.

Descent 1 Limits

Item	Max	Min	Notes
Cube size	~100	1	Cubes over this size can show up distored
Cubes	800	1	Level 1 "Lunar Outpost" uses 285 of these.
Vertices	2800	8	Since cube corners share vertices, this number does not need to be 8 x 800 = 6400. On average, cubes share over half their corners with other cubes.
Walls	175	0	This limit is hard to reach unless you make lots of illusions.
Objects	350	1	The only object required is Player 0
Robot Makers	20	0	This is a cube where robots materialize
Fuel Centers	50	0	This is a cube which recharges your ships power cells.
Textures	?	1	Too many textures cause the level to come up with static instead of textures. Depends on memory.

Descent II Limits

Item	Max	Min	Notes
Cube size	~1000	1	Cubes over this size can show up distored and can cause the game to crash when players get too close.
Cubes	900	1	Level 1 "Ahayweh" uses 228 of these.
Vertices	3600	8	Since cube corners share vertices, this number does not need to be 8 x 900 = 7200. On average, cubes share over half their corners with other cubes.
Walls	254	0	This limit is hard to reach unless you make lots of illusions.
Objects	350	1	The only object required is Player 0
Robot Makers	20	0	This is a cube where robots materialize
Fuel Centers	50	0	This is a cube which recharges your ships power cells.
Textures	?	1	Too many textures cause the level to come up with static instead of textures. Depends on memory.

Rules

Rules define those things required by the game to prevent the game from aborting. Below is a list of a few rules you should try to follow before testing your level:

1) Object 0 must be player 0 because the game uses this object as the starting point for a single player game.
2) Always run **File/Check mine** before you test your level. This feature checks most of the data within a level to make sure that the editor created it correctly. It also checks to see that the user has not created any illegal geometry or is missing essential items for a complete level.
3) A mine should have exactly one of each of the 8 players (id = 0 to 7). Otherwise, the game will abort when a multiplayer game is run.
4) A mine should have exactly one of each of the 3 cooperative players. Otherwise, the game will abort when a cooperative multiplayer game is run.
5) Never put a door on a side of a cube which has no joining cube. Otherwise, the game will abort when the door closes.
6) Each level should have a reactor or an end boss robot so you can finish the level.
7) Each level should have an exit which opens when the reactor blows up (or the end boss robot is destroyed).
8) Keep the maximum dimension of the cube under 100. Larger cubes sizes have been shown to work, but your results may be vary. It is recommended that cubes do not exceed 100 (5 standard cubes lengths).
9) Do not use Lava or Water as the object texture override. This will cause the game to crash when the player bumps into the robot.

Breaking the rules

Breaking the rules

When the game does not like something, it executes a function called "assert". Assert sends a message to the DOS window with some information about what it did not like. This is indented for level designers who want to find out why their level is not working.

Below is an example of a typical assertion (see last line where is says 'Error:...')

```
C:\games\DESCENT>descent
DOS/4GW Professional Protected Mode Run-time  Version 1.97
Copyright (c) Rational Systems, Inc. 1990-1994

DESCENT   v1.5   Jan 5, 1996
Copyright (C) 1994, 1995 Parallax Software Corporation
DESCENT is a trademark of Interplay Productions, Inc.

Type 'DESCENT -help' for a list of command-line options.

Available memory
----------------
Conventional:     482 KB
Extended:      15,860 KB
Virtual:       91,744 KB


Error: Invalid station type 95 in fuelcen.c
```

This is not the same thing as a "40 column crash". A 40 column crash is caught by the game's operating system and it does not contain any information that is useful for a level designer.

Guidelines

Guidelines define standards that level designers should adhere to in order to create professional looking levels:

1) Avoid making tunnels which overlap one another. Although the game can handle this, it is confusing when viewed from the "AUTOMAP" option in the game. Such structures are not possible in the real world and should be avoided in levels to give a greater sense of realism.
2) Keep the sides of a wall as flat as possible. That is to say, keep all corners of a particular side co-planer. If you were to lay the side of a cube on a flat surface, all four corners should touch the surface. Some variance is taken into account by the game's graphics engine. If the engine determines that a point is out of whack, it will split the side into two triangles and render them as such.
3) Keep objects within the boundaries of their cube. All objects have a cube number. If you move the object outside of the cube, the game will try to correct the problem by making the object head toward or start in the center of the cube. There is no real advantage of moving an object outside its cube.
4) Do not put a key behind a door which requires that key (unless of course you have provided another path to reach the key).
5) Avoid hiding keys in robots unless it is obvious that you must destroy this robot.
6) Prevent reactors from floating off the ground. They look funny from the bottom.
7) Make sure you cannot see the *end of the exit tunnel* from the exit door. The game does not draw any texture for this side so the player will see whatever was left over from the previous screen. You can help avoid this problem bending the tunnel and by gradually decreasing the texture lighting in the exit tunnel to zero at its end.
8) Use the "Average corner lighting" feature of the light adjustment dialog to keep the texture corner lighting smooth.
9) Keep lights on the ceilings to give the player some idea which end is up.
10) Avoid making rooms with a large number of cubes joined together. This slows down the game for those of us who do not have ultra fast computers (yet).
11) Align textures from one cube to the next (especially those with patterns).
12) Adjust the cube light so robots will appear to light up when they fly under textures which have lights on them.
13) Avoid making walls which have no thickness because they look unreal. The editor is designed to prevent you from accidentally making this type of wall but you can still do it if you try hard enough.
14) Avoid placing a secondary texture on a side which completely covers the primary texture. This simply slows down the game.
15) Make sure the primary texture is not transparent. This will cause the player to see a wall which is not completely redrawn.
16) Make sure doors are put on both of the facing sides of joining cubes. Otherwise, the player will only see the door from one side.
17) Keep the animation clips on adjacent doors the same. Otherwise, the player will see one type of door when the door is closed, and another type of door after the door begins to open.
18) Put doors on standard size cubes. Tiled doors look funny when they open and close. *HINT: Use the **Stretch to Fit** button in the **Texture Align** dialog.*
19) Discolor or shade secret walls a bit to give the player a chance to find it. What may seem obvious for the level designer, may be obscure for a player.

Tricks

Tricks are not recommended to create professional looking levels, but, on the other hand, they sure are fun! Below is a list of a few tricks that you can play around with:

1) Hide robots behind locked doors which open when the player crosses a trigger.
2) Try putting an animated texture on a robot like a fan. This can make for some pretty bizarre effects.
3) Try making an energy draining trigger with a negative value. When you fly through this trigger your energy will actually jump up! (for Descent 1 only)
4) Make an illusion wall and put an animated texture on it or some other partly transparent texture.
5) Put multiple powerups or a weapons of the same type on top of one another. In the game, they will appear as a single object but when the player flies over them, some will be left behind.
6) Change the texture on a door after you set the clip number. This will make the door appear different until it is first opened.
7) Try making an illusion of water so you can fly through it. This gives illusion of underwater flying. Or, try using lava instead of water.
8) Put robots behind illusions that look like a normal wall. Then set their behavior to "Still". This gives the impression that the walls are shooting at the player.
9) Try your luck at editing the Advanced Object Data. Most of these entries don't do anything, but some combinations of robot type and parameter have some odd effects.

Opening and Saving

Opening

Selecting File/Opens will bring up a dialog box so you can open an existing level. The editor keeps track of the last level you have worked on. You can quickly open this level by selecting it from the bottom of the File menu.

When you are finished working on a level and you want to save the changes, select the Save command

You can create new levels by using the New command.

To quickly open a level, click the Open button on the toolbar.

 Open button

Saving

Selecting File/Save As will bring up a dialog box so you can save the current level. When the level is saved, a mission file (*.msn) will be created automatically. The mission file allows the game to see the new level when selecting a mission. The name of the mission file is the same as the name of the RDL or HOG file except the extension is *.msn. By default, the mission file created is set to type = normal which means this is a single player level.

Selecting File/Save will bring up a dialog box if this is the first time you have saved this level after creating a new level or after opening an old level. Or, if this is not the first time, it will be saved immediately. Once again, a mission file will be automatically created.

To quickly bring up the "Save As" dialog box, click the Save button on the toolbar.

 Save button

Dialog Box Options

File Name

Select or type the name of the level you want to open or save. This box lists files with the filename extension selected in the List Files Of Type box. To see a list of files with a particular extension, type an asterisk (*), a period, and the three-character extension, and then press ENTER. To see files with more than one filename extension, type a semicolon (;), a second asterisk (*), a period, and another three-character extension. For example, if you want to see all files with the .RDL extension and all files with the .HOG extension, type *.rdl;*.hog

List Files Of Type

Select the type of file you want to see in the File Name list.

Descent Level (*.RDL)

Lists all Descent 1 files in the current directory that were saved with the .RDL extension.

Descent Level (*.RL2)

Lists all Descent 2 files in the current directory that were saved with the .RL2 extension.

Hog File (*.HOG)

Lists all Descent 1 or Descent 2 files in the current directory that were saved with the .HOG extension. A list box will appear when trying to open a hog file which contains the sub-files within the hog.

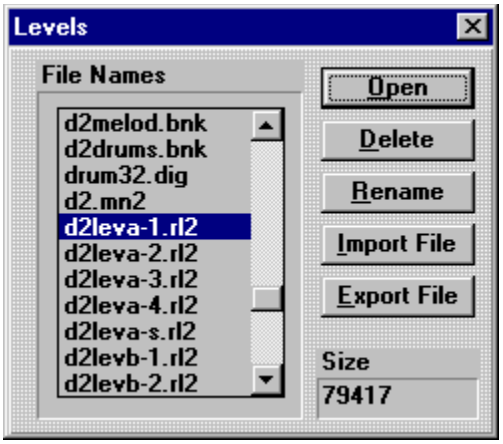
Drives

Select the drive that contains the file you want to open.

Directories

Select the directory that contains the file you want to open.

Hog Manager



File Names

This is a list of all of the sub-files in the HOG file.

Open

This with either load an RDL or RL2 file into the mission builder or run an external application assocated with the extension.

When an RDL or RL2 file is loaded, the mission builder looks for two other files with the same base name. One is a POG file which contains custom textures for the level. The other is a HXM file which contains custom robot behavior.

If you open a non RDL or RL2 file, the mission builder will prompt you to see if you want to extract the file from the HOG for editing with an external program. After extracting the file, the mission builder will call a windows function which will try to load the program associated with the file's extension. If successful, the external program will be launched and automatically open the file. Any changes to the file will have to be "imported" back into the HOG file if you want to keep your changes. This feature extends the mission builder capabilities for the future by allowing 3rd party developers to make programs for the specific file types.

Warning: If you use another program to edit POG or HXM files, then you may lose information if you choose to save your change to custom textures or robot attributes in the mission builder. If you don't want to change these files, just press the Cancel button when the mission builder prompts you to see if you want to save the custom textures or robots.

Delete

The delete button will remove a file from the HOG. This is an irreversable process, so be careful.

Rename

You may rename any file in the hog file. The mission builder will automatically truncate the file name to 12 characters but will not do any checking to make sure you typed in a legal name.

Import File

This allows you to add files to the hog file. Files will be appended to the end of the HOG file due to the time it takes to move around the other files.

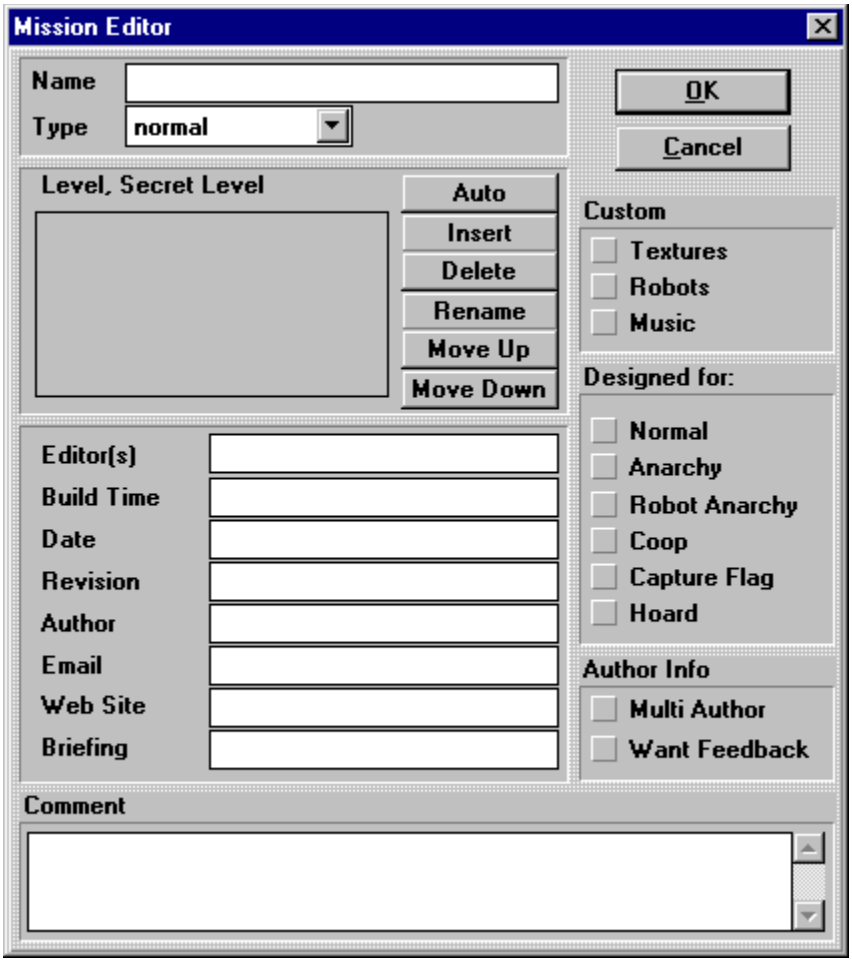
Export File

This allows you to extract a file from the HOG.

Size

This contins the number of bytes in the currently selected file.

Mission Manager



Name and Type

Name This is the name of the mission as it appears on the menu when you start a new game.

Type Set this to either Normal (single player game) or Anarchy (multiplayer game)

Levels and Secret Levels

Auto Reads the current HOG file and fill in the list box with the RDL/RL2 file names that were found

Insert Allows you to insert a level name to the list

Delete Deletes the selected level name from the list

Rename Renames the selected level name

Move Up Moves the selected level up the list by one entry

Move Down Moves the selected level down the list by entry

Information

Editor(s) Editing tools: DMB2, DTX, Devil, etc..

Build Time This is how long it took you to create this level (hours, days, weeks, whatever).

Date This is a good place to put the release date of your level.

Revision Start out with revision 1.0. When you make a minor change, change this to 1.1, 1.2, etc.. When you make a major change, go to 2.0.

Author Put your name here and anyone else who help you create the level.

Email	Your e-mail address so people can contact you.
Web Site	Where to get this level.
Briefing	If this contains a TXB file, type in the name of it here. You will need to use a 3rd party utility to create briefing files, but at least you have a place to declare that you are using one.

Comment

Put whatever you want here. Try to limit the line lengths 80 characters long.

Custom

Textures	This box will automatcally be checked if you are using a POG file.
Robots	This box will automatically be checked if you are using a HXM file.
Music	Check this box to indicate that you are using custom music.

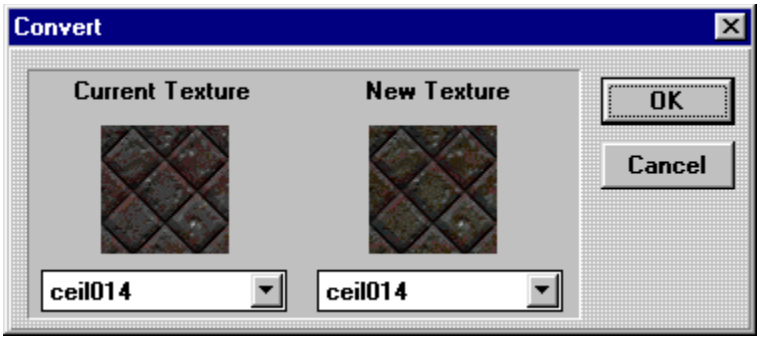
Designed for

Normal	Single player game play
Anarchy	Multiple player death match mode
Robot Anarchy	Multiple player with robots mode
Coop	Cooperative multiple player mode
Capture Flag	Capture the Flag multiplayer mode
Hoard	Multiplayer Hoard mode

Author Info

Multi Author	Check this if there were multiple authors for this mission.
Want Feedback	Check this if you want people to e-mail you with feedback.

Converting Levels



Converting from Descent 1 to Descent 2

To convert a Descent level (*.RDL) to a Descent II level (*.RL2):

1) Load the Descent 1 level you want to convert
2) Select "Convert from D1 to D2" from the File menu.
3) (optional) Edit the list of replacement textures
4) Press OK

The texture on the left is the texture from the Descent level you want to convert. The texture on the right is the Descent II texture that will be substituted.

Customizing Textures

As you scroll through the left textures, the texture on the right will show you the closest matching Descent II texture. You can change the Descent 2 texture by picking a different texture from the list on the right side.

Robots

The converter does not change the robots to Descent II robots. This is because Descent II supports all the Descent robots. If you want to change the robots, you need to select the robot you want to change (with the right mouse button) then set the object ID to one of the Descent II robots.

The same rule applies to power ups and all other types of objects.

Checking the new level

After you have completed a conversion, it might be necessary to fix a few things. Run the "check" feature (under the File menu), and fix any "Errors" that may exist.

Display

View Options

The Mission Builder supports 6 ways to view cubes:

Hide	Hides all cube lines and points.
Partial	Shows only the cube lines which are not shared with other joined cubes.
All	Shows all cube lines.
Nearby cube lines	Shows the current cube, its nearby joined cubes and all mine vertices.
Texture Mapped	Shows the mine with textures painted on the cubes.
Partial Texture Mapped	Shows the mine with textures painted on the nearby cubes.

Hint: You can quickly change between these modes by using the F7, F8 and F9 keys.

*Note1 : In **Nearby cube** mode, the current cube will always be centered on the screen. Therefore, panning functions will be disabled in this mode.*

*Note2: **Texture Mapped** mode requires proper PIG file selection in the File/Preferences dialog.*

Shading and High Resolution

There are two options which dictate how to the texture mapping the level.

Shading	Enables texture shading based on texture corner lighting.
High Resolution	Four times as many pixels are drawn in this mode.

Zooming, Panning, and Rotating

USING THE MOUSE

The fastest way to move around is with the mouse. Hold down the CTRL key and move the mouse to move the level around on the screen. When you hold down both the CTRL key and the SHIFT key, then you can rotate the mine. Pressing the right mouse button while the CTRL key is held down causes the level to ZOOM in. Likewise, the right mouse button zooms out.

USING THE KEYS AND AUTO REPEAT BUTTONS

There are several ways to move the mine around on the screen. You can use one of the buttons on the tool bar, use the menus, or use the hot keys (as listed on the menu).

 Pans left

 Pans up

 Pans right

 Pans down

 Zooms in

 Zooms out

 Rotates vertically clockwise

 Rotates vertically counter-clockwise

 Rotates horizontally clockwise

 Rotates horizontally counter clockwise

 Rotates clockwise

 Rotates counter clockwise

Centering

You can center the mine on the current cube, the current object, or center the entire mine.

To quickly center something, set the focus of the screen on the main window, then press one of the following hot keys:

Center on Cube	Home
Center on Object	Shift+Home
Center Mine	Ctrl+Home
Align side rotation	Alt+Home

Turning on/off Walls, Specials, Lights, and Objects

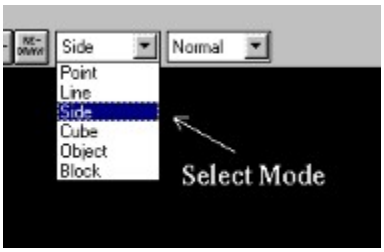
If you only want to show certain objects or suppress the "X" drawn for walls, you can check or uncheck any of the various options on the bottom of the View menu.

*Note: **Specials** refer to coloring of fuel centers, robot makers, reactor cubes, and flag goals.*

Selections

Select Mode

The **select mode** sets what will be effected when moving, rotating, or sizing. To change the **select mode**, choose one of the items on the list located on the menu bar (see picture below).



Modes are automatically switched when you press the following keys:

'p' or 'P'	sets to point mode
'l' or 'L'	sets to line mode
's' or 'S'	sets to side mode
'c' or 'C'	sets to cube mode
'o' or 'O'	sets to object mode

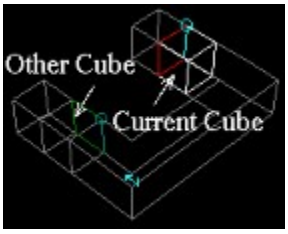
Note: left and right arrow keys change the current side but do not change the select mode.

Selecting Cubes

The level editor highlights two cubes to help with editing. Most of the time, only the current cube is used. But some functions (like using the curve generator), requires the user to specify two cubes at once. This is where the other cube comes in.

The **Current Cube** is colored white, green, and red. The green part represents the currently selected side. The red part represents the currently selected point, line, side, or cube depending on the select mode.

In the following picture, The select mode is set to "Side", the current cube is colored white, the current side is colored red, the current line is colored cyan, and the current point is circled in cyan.



The **Other Cube** is also shown in the picture above. Notice that its current side and point are also highlighted. This is important when joining cubes, using the curve generator, and setting robot or door triggers from the Insert menu.

You can select the current cube in one of the following ways:

1) by clicking on it with the left mouse button.
2) by pressing 'c' or 'C' keys to increment/decrement the cube number

3) using the arrow up or arrow down keys to move to the adjacent cube.

To swap the selection between the primary and secondary cube, simply press the space bar or select **Other Cube** from the Cube Dialog.

Selecting Objects

The current object will be colored while the remaining objects will be gray. The colors are as follows:

Cyan	Players and Coop. players
Yellow	Powerups (includes keys, weapons, etc...)
Blue	Hostages
Red	Reactor
Dark Red	Robots

You can change the current object selection by clicking the right mouse button on the desired object. Or, you can press the 'o' or O' keys to increment/decrement the object number.

*Note: If you select the **other cube** by pressing space bar, you will get an alternate current object. In other words, there are actually two current objects: one associated with the current cube and another associated with the **other cube**.*

Insert and Delete

You can insert or delete items using the dialog boxes, menus, or keyboard.

Insertion

There are four basic categories of things you can insert from the Insert menu:

Item	Hot key	Description
Cubes	Insert	normal, reactor cube, robot maker, or fuel center
Objects	Alt+Insert	copy of current object, player, robot, mine, or powerup
Doors	(none)	auto door, exit, or secret exit
Triggers	(none)	door activator, robot maker activator, shield/energy drainer.

Each of these items perform a series of commands that you would otherwise have to do by hand. You will probably make use of these functions extensively.

Cubes

When a cube is inserted, it takes the textures and light values from the current cube. The shape of the cube depends on the cube insertion mode (described below).

When a **reactor** cube is inserted, the cube type is set to **reactor**, the textures are set to "rock006", a reactor object is added, and it is linked to any exit door that may exist.

When a **robot maker** is inserted, the cube type is set to **Robot maker**, the textures are set to **misc16-anim**, and single **Class 1 drone** is added to the robot maker list.

When a **fuel center** cube is inserted, the cube type is set to **Fuel center**, the textures are set to **misc059**, and two illusion walls are added with textures of **misc17-anim**.

Objects

When an object is inserted, it takes a copy of the last object and places it in the center of the current cube. If there is already an object in this cube, then it bumps the position of the new object up a bit.

NOTE: Object 0 must be player 0.

If a **player** object is added, the object type is set to player, and the **id** is set to the lowest unused player **id**. If all players exist, then an error message will be displayed.

If a **robot** object is added, the object type is set to **Robot**, and the **id** is set to **Class 1 drone**.

If a **weapon** object is added, the object type is set to **Powerup**, and the **id** is set to **Laser**.

If a **powerup** is added, the object type is set to **Powerup**, and the **id** is set to **Energy boost**.

Doors

When an auto door is inserted, a wall is added to each side of joining cubes. On each door, the wall type is set to **Door**, the wall flag **Auto** is set, and the clip number is set to **door01**.

When an exit or secret exit door is added, a wall is added to each side of joining cubes. On each exit, the wall type is set to **Door,** the wall flag **Locked** is set, and the clip number is set to **door13**. In addition, a trigger is added to the current side's wall with the **End Level** or **Secret Exit** flag set.

Triggers

Inserting a trigger will first add a wall of type **Open** to the current cube side, then add a trigger with the appropriate flags set.

Inserting a **door opening** or **robot triggering** trigger is a little trickier. This requires that the secondary cube be placed on the cube (and side) which contains the item being triggered. Then a wall will be created of type **Open** on the current cube side. Finally a trigger will be added to the wall with the appropriate flags set.

Deleting

There are four also four things you can delete from the Delete menu:

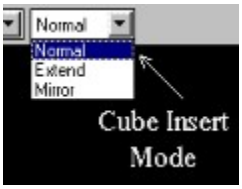
Item	Hot key	Description
Cubes	Delete	current cube
Objects	Alt+Delete	current object
Door/Walls	none	current side's wall.
Triggers	none	current side's trigger

If a wall is deleted and it contains a trigger, the trigger will also be deleted. When you delete a wall, it will not automatically delete the joining cube's wall. You have to select this wall first, then select the delete command again.

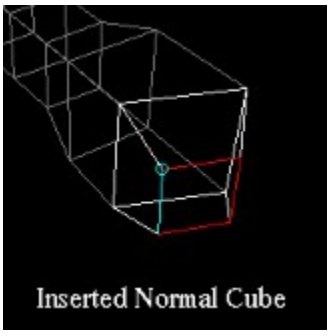
Cube Insertion Modes

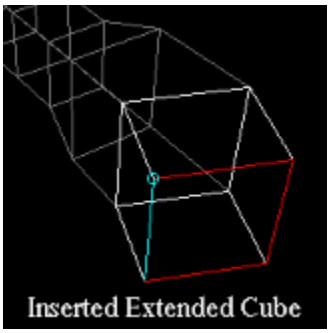
There are three options for inserting cubes

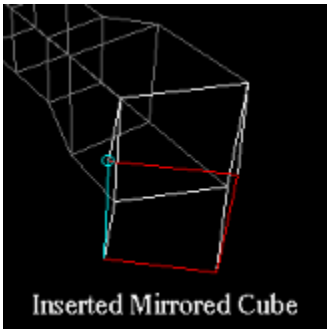
1) Normal Makes the new side 20 x 20 (standard cube side)
2) Extended Makes the new side the same size as the current side.
3) Mirror Makes a mirror of the current cube



Below is a sample of how these modes effect the shape of an inserted cube:







Each mode has its advantages. It just depends on what you are trying to do.

Changing the Shape of Things

The Edit Tool Bar

The edit tool bar matches the numbers on a standard number pad. If you enable the Num Lock key then you can use the number pad to perform the same functions as when you click on the Edit Tool Bar buttons.

To bring up the editing tool bar:

press this button on the menu tool bar
or select "Edit tool bar..." from the Tools menu
or press F5

These buttons do different things depending on the Select Mode:

Moves point, line, side, cube, object, or block in the direction of the currently selected side.

Same as above but moves away from side.

Rotates side, cube, object, or block in relative to the current side.

Same as above but rotates in opposite direction.

Enlarges lines, sides, or cubes (does not enlarge blocks).

Same as above but shrinks.

Bends a side to the left or moves item in the direction of the current line.

Bends a side upwards or moves item in the opposite direction of the current line..

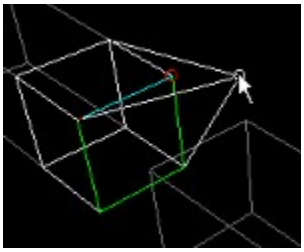
Bends a side to the right or moves item in the direction of the adjacient line.

Bends a side down or moves item in the opposite direction of the adjacient line.

Joining and Unjoining

Using the Mouse

If you need to move a point to another points location, then this feature will do the job quickly and easily. Hold the left mouse button on the current point, drag and drop mode will be enabled. As you hold down the button, three white lines will be drawn connecting to the point being moved. When you are on another point, a larger white circle will be drawn.



Move the pointer over an existing point. When you are on a point, a white circle will be drawn. When you have reached the point you want to move to, release the mouse button.

After moving a point, the editor will figure out whether the current cube is connected to other cubes. If a match was found, then the cubes will be attached. If not, then the cube will return to it's previous shape.

The editor automatically makes and deletes vertices as needed when you drag and drop.

Note: You can not release the point to thin air since the editor has no idea how far into the screen you want the point to be. Keep in mind, this is a 3-d editor.

Using the menus or keyboard

You can also join or unjoin sides, lines, or points using the menu or by pressing the hot key:

Joining:

Item	**Hot key**	**Description**
Current Side	Ctrl+C	Joins current side only
Sides	Ctrl+S	Joins sides by adding a new cube or by moving the current cube.
Lines	Ctrl+L	Joins lines
Points	Ctrl+P	Joins points

Unjoining:

Item	**Hot key**	**Description**
Current Side	Ctrl+Shift+C	Joins current side only
Sides	Ctrl+Shift +S	Unjoins sides
Lines	Ctrl+Shift +L	Unjoins lines
Points	Ctrl+Shift +P	Unjoins points

The editor automatically makes and delete vertices as well as updating the connection of cubes.

Hint: The Mission Builder uses the current point of each side being joined to determine how to connect the points. You should select the current point *before* joining.

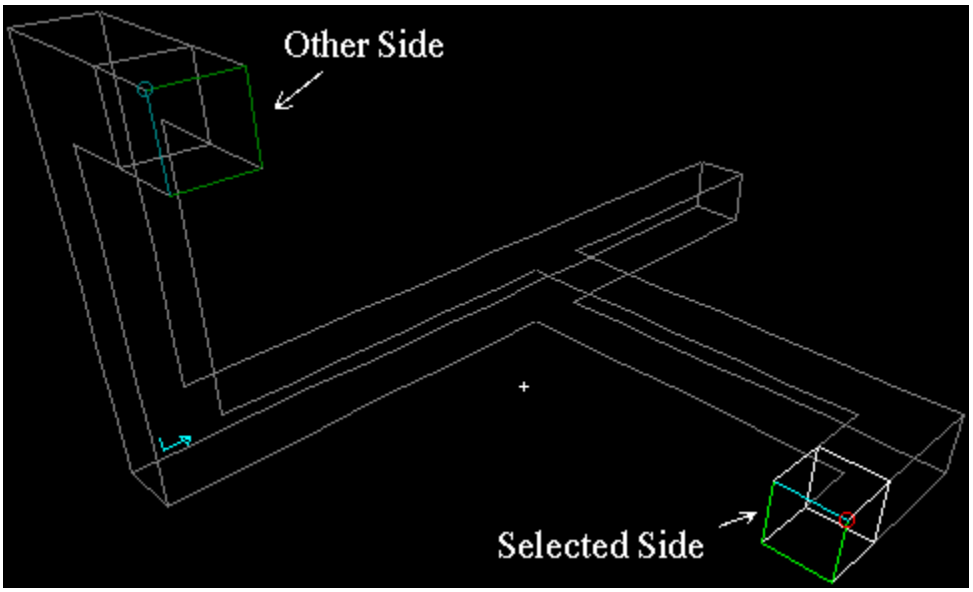
Note: When you unjoin sides, lines, or points, nothing appears to happen. But, the editor creates new

points with the same coordinates as the existing points. After you join, try moving the effected point and you will see the results more clearly.

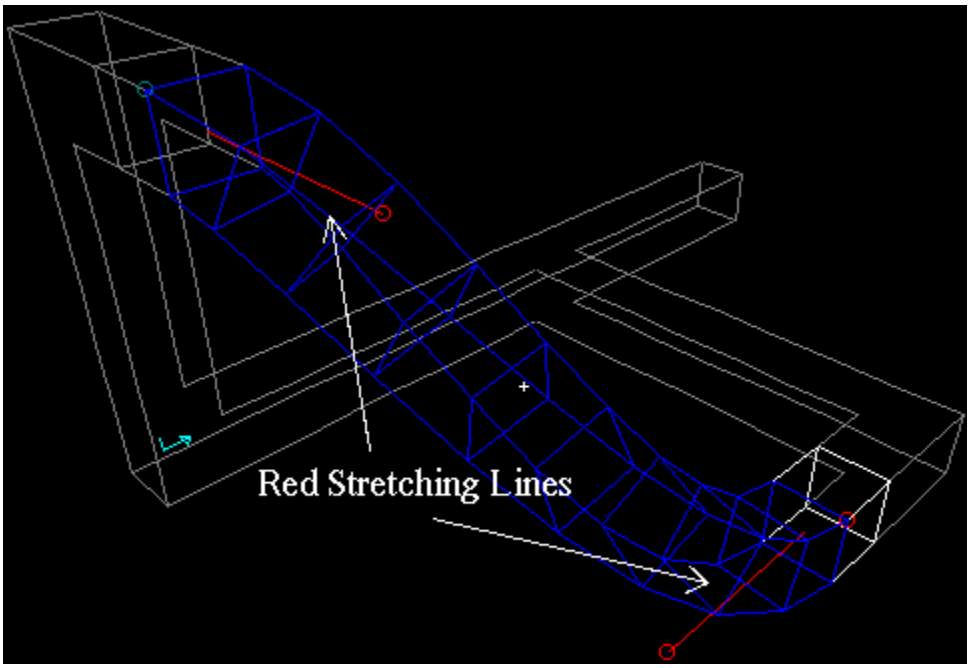
Curve Generator

Overview

The Curve Generator feature of the editor allows you to make tunnels of varying shape and size. First you must choose two cube sides which represent the end points of the tunnel.



Then select Curve Generator from the Tools menu or press the letter G.



The blue cubes represent the shape of the tunnel being created. The red stretching lines control the direction and angle of the tunnel.

To change the shape of the tunnel, you can either change the tunnel end points or you can adjust the

length and direction of the stretching line.

When you are finished creating the shape you want, press the "G" key again. Then you can decide whether you want to keep the tunnel or not.

Stretching

Press the "[" or the "]" bracket keys to adjust the length of the red stretching lines.

Bending

To bend the tunnel, you must bend side of the cube which attaches to the end of the tunnel. This is most easily done by using the arrow keys on the Number Pad (numbers 2,4,6,8). Make sure the Num Lock is active and that you have selected the side which faces the tunnel.

Twisting

You can also twist the tunnel by pressing the "P" or "L" keys. This tells the curve generator how to connect the four end points of the two cubes.

Size

The size of the tunnel is based on the sizes of the two cube sides at the end of the tunnel. The bigger the sides, the bigger the width of the tunnel.

Hints

Before you make a tunnel, it is best to create new cubes for each tunnel end. This way you can adjust these new cubes to change the shape of the tunnel without effecting the shape of the mine.

Once you have created a tunnel, use the check feature to make sure the new cubes have legal shapes. The curve generator can create some pretty weird shapes if you are not careful.

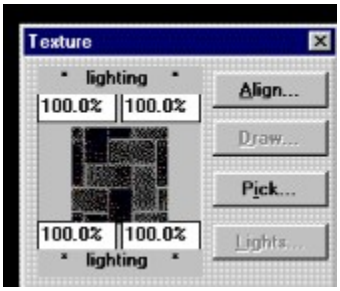
While you have enabled the curve generator, you can not delete or add any cubes.

Light Adjustment

Corner Light

Corner light determines how bright the cube side textures will appear. Each cube has 6 sides and each of those sides has 4 corners. This yields 24 light values per cube (phuh...). Luckily, only those sides which have no joining cube or those sides which have a wall, require a light value.

To adjust these values, type in the percentage from (0 to 200%) that you want in the Texture Dialog box.

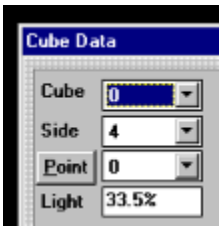


Note: values greater than 100% will not appear any brighter than those at 100%.

Cube Light

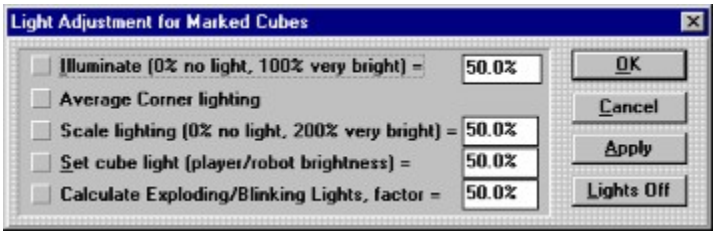
Cube lighting effects how bright polygon objects (robots, the reactor, and players) will appear when they are inside of the cube.

To adjust these values, type in the percentage from (0 to 100%) that you want in the Cube Dialog box.



Adjusting multiple cubes at once

There are four lighting calculations you can perform on multiple cubes at once. But, before you can execute any of these calculations, you must mark the blocks you want to adjust.



To mark a block, you can either drag a box around it with the mouse or you use the options under the Edit menu.

Illuminate

This adjusts the texture lighting for

Note: The values assigned to each light are the same as those used by Parallax.

Average Corner lighting

This will average the light setting of all marked cube sides which share a common point. For example, if side1 has a corner light of 50% and side2 has a corner light of 100%, then they will both be set to 75%. This can actually cause bright lights set to 100% to become dimmer. It is for this reason that it is recommended that you scale the lighting after you perform an average.

Scale lighting

This function literally multiplies the side light values by a percentage. For example, if the side light is 80% and you set the scale to 50%, then the new side light value will be 40%.

Note: if the side light is 0%, then it does not matter what the scale is set to because zero times anything is still zero.

Set cube light

This function sets (absolutely) the cube light of all the marked cubes. For example, if you set this value to 75%, then all of the cube light values will be set to 75% regardless of their previous setting;

Calculate Exploding / Blinking Lights

This function sets the delta lights which are used by exploding, blinking and trigger controlled lights. When you save a level, you have the option to update this data. If you decide to do it by hand, then you can mark all the blocks that you want to have delta light. Sometimes you may want to exclude areas that are affected by these lights. In this case, simply unmark the cubes where you don't what the light to be reduced.

Lights On/Off Button

This allows you to turn on or off all the delta lights. When the lights are off, the entire level should be dark. This turns off even non-breakable lights so beware.

For more information on lights, see the following:

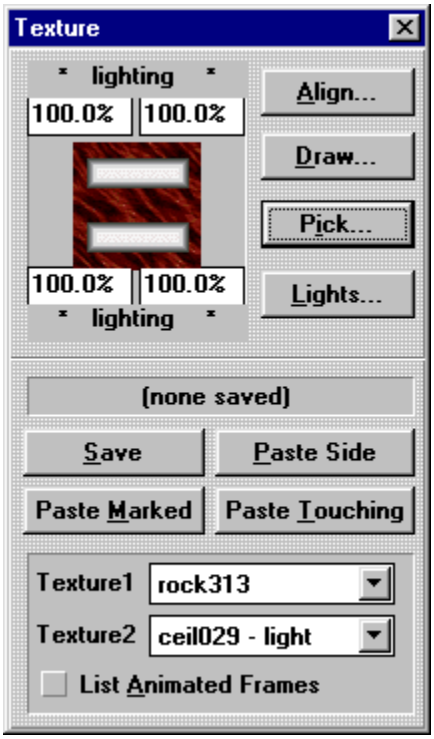
Exploding Lights	Automatic exploding lights
Blinking Lights	Blinking light adjustment

Textures

Cube textures

To change the textures on a side of a cube, you need to bring up the **Texture** dialog.

press this button to bring up the Texture dialog
or select "Texture edit" from the Tools menu.



If you see the following message:

you will need to select the *Preferences* item under the *File* menu. Then set the **Pig File Directory** to the directory where descent.pig file can be found.

If you see the following message:

it means that you need to upgrade your version of Descent or that you have a shareware version of Descent.

If you see the following message:

simply press the 's' key until a texture appears.

Now that you have a texture, use the **Texture1** and **Texture2** combo boxes to select the desired textures.

Note: Avoid using a texture2 which completely overlaps the texture1.

Cut and Pasting Textures

Use the Save button to copy the current side's texture and light values. Then select a different side and press the Paste Side button.

If you want to paste a bunch of sides at once, then select a side along a series of cubes which use the same textures. The press the Paste Touching button. This will replace all touching textures along the pathway which have the same textures.

If you want to replace all the textures of each marked cube, press the Paste Marked button.

Hint: If you want to only change texture 1, then select (none) for texture 2. This is useful for the PASTE ALL function when you want to change the background texture and leave all the lights alone.

Aligning Textures

Pressing the Align.. button brings up the **Align Texture** dialog. This allows you to adjust the orientation of the texture on the cube or wall.

For more information, see: Texture Alignment

Texture Picker

Pressing the Pick.. button brings up the **Texture Picker** dialog. This allows you to select textures with a click of the mouse button..

For more information, see: Texture Picker

Editing Textures (Descent II v1.2 and above)

Pressing the Draw.. button brings up the **Edit Texture** dialog. This allows you to modify the texture bitmap with custom graphics of your own.

For more information, see: Texture Edit

Blinking Lights (Descent II v1.1 and above)

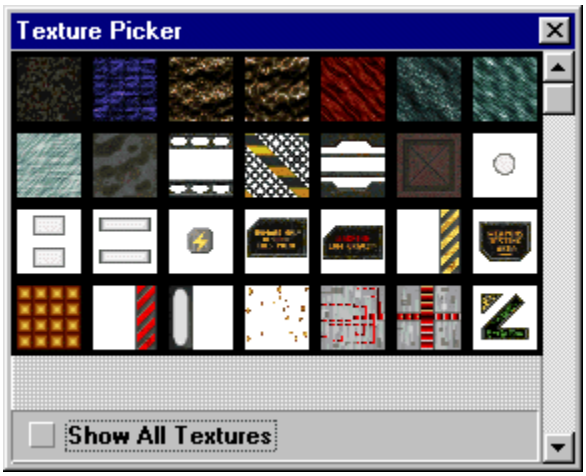
Pressing the Lights.. button brings up the **Blinking Lights** dialog. This allows you to add, delete, or modify an light which blinks inside the game.

For more information, see: Blinking Lights

Texture Picker

Texture Picker

The texture picker dialog allows you to select cube or object textures simply by clicking on the texture. This makes texture selection faster than scrolling through a list box.



Use the slide bar to scroll through the texture selection.

Texture 1, Texture 2, and Object Texture Selection

The left and right mouse button along with the shift key will do the following:

Left Mouse	Sets Texture 1 of the current side
Right Mouse	Sets Texture 2 of the current side
Shift Left Mouse	Sets the current object's texture override.

HINT: If you press the right mouse button on Texture 1 bitmap, then the texture 2 will be set to "none".

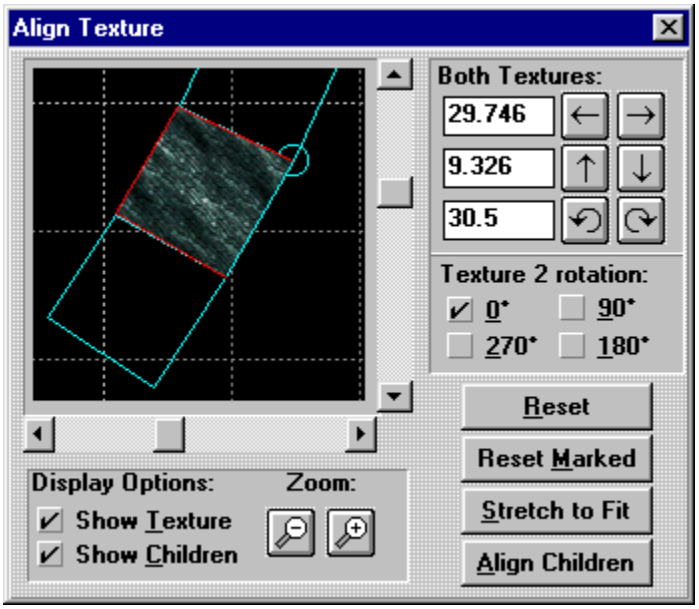
Show All Tetures

The **Show all textures** check box allows you to show all the textures or only the currently used textures.

Aligning Textures

Press the 

button on the **Texture** dialog to open up the **Align Texture** dialog.



Use the 



buttons and the


buttons to move and rotate the texture rectangle around until you have the look you want.

Use the  and

buttons to get a closer look at things. This will cause the arrow buttons and spin buttons to become more sensitive which allows fine tuning of the texture position.

Texture 2 has an addition attribute which tells the game which corner to start drawing the texture from. This is used when you want to rotate a sign post so it looks correct on the side. Zero degrees usually takes care of it unless you start spinning and twisting cubes around.

The 

button recalculates the texture position from the coordinates of the cube's side. It also puts point 0 in the center, point 1 down along the vertical axis and the rest of the points where ever they land.

The 

button is the same as the reset button except it works on all sides of all the currently marked cubes.

The 

button forces the texture to fit on the cube side.

The 

button will make children cubes (which have the same texture) line up to the current

side. This is best if used when the **Show Children** check box is checked.

And I am not going to tell you what the **Show Texture** button does <grin>...

Exploding Lights

Exploding Lights

Descent II has several textures that look like lights which will explode when shot. When a level is saved, the Mission Builder prompts you to calculates how much the adjacent cubes light will be reduced when the light is shot. This reduced amount is called a delta light value.

There are some limitations on how delta light values are calculated:

1) Only cubes up to six cubes away will be affected.
2) Delta light is inversely proportional to the distance from the light.
3) Light will not pass through doors, but will pass through illusions
4) Only sides 110 degrees from the lights direction will have delta values. In other words, the side has to be pretty much in front of light source. We allow delta light to go slightly behind the light source because the cubes in the area naturally reflect some of the light. This makes it look more natural.
5) Cubes which are direct children to the cube with the light on it, are given delta light values on all sides even if they exceed the 110 degree limit described above.

HINT: Try not to put lights too close together, because when you blow one light, the other light will be dimmed even if it hasn't been blown up yet. Also, avoid using thin cubes around the light source, because this can fool the delta light calculations and create some odd results (unless of course you are trying to make weird effects).

Note: You can also calculate the delta light by hand using the **Light Adjustment for Marked Cubes** dialog.

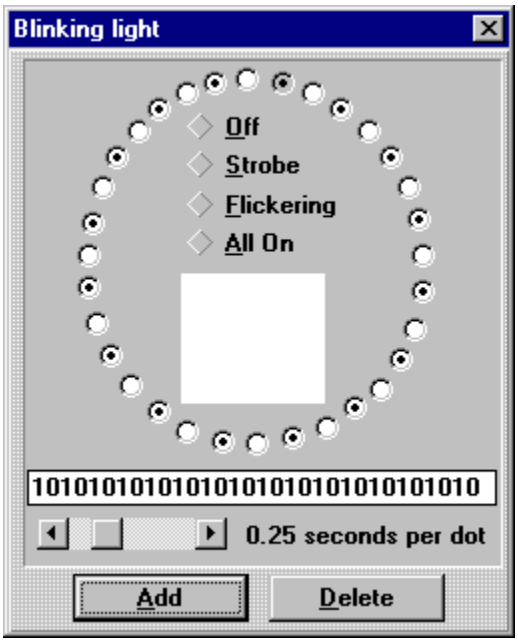
For more information on lights, see:

Blinking Lights	Blinking light adjustment
Texture Lights	Texture light adjustment

Blinking Lights

Blinking Lights

Descent II (version 1.1 and above) has a feature which allows you to make lights blink. This adds realism to a level and is highly recommended for Descent 2 levels.



Blinking lights may be added to any light type texture. If you set an exploding light to blink, then, it will flicker until it is shot out.

Descent II uses 32 bits and a timer value to set the blinking lights. The Mission Builder represents these 32 bits by a circle of 32 dots. A marked dot means that the light will be on for the number of seconds shown under the slide bar. Then the next dot will be either on or off for the same amount of time.

To make this a little easier, the Mission Builder has a couple presets for you to choose from:

Off	Turns all dots off
Strobe	Makes the light turn on and off on regular intervals
Flickering	Makes the light flicker
All On	Turns all dots on.

These radio buttons allow you to set the dots quickly without having to set each one individually. The **Off** and **All On** presets are used to get you close to the desired results, but if you leave all dots on (or off) the light will not blink.

To help give you an idea of what the effect will look like in the game, there is a square in the center blinks at the rate the light will blink in the game. The grayed dot shows the current bit effecting the light.

The slide bar adjusts the number of seconds the light will be on (or off) for each dot. The total cycle time (the time it takes to complete one revolution) is 32 times the time shown below the slider. For example, if the time says "0.25 seconds per dot" then the cycle time will be 32 * 0.25 = 8 seconds. So if you want the light to come on for 1/4 of a second every 8 seconds, then you only need to set one dot on.

The add / delete buttons are used to add a light or remove it. Descent 2 has a limited number of textures which can blink. Textures that can blink have the word "'light" at the end of their name.

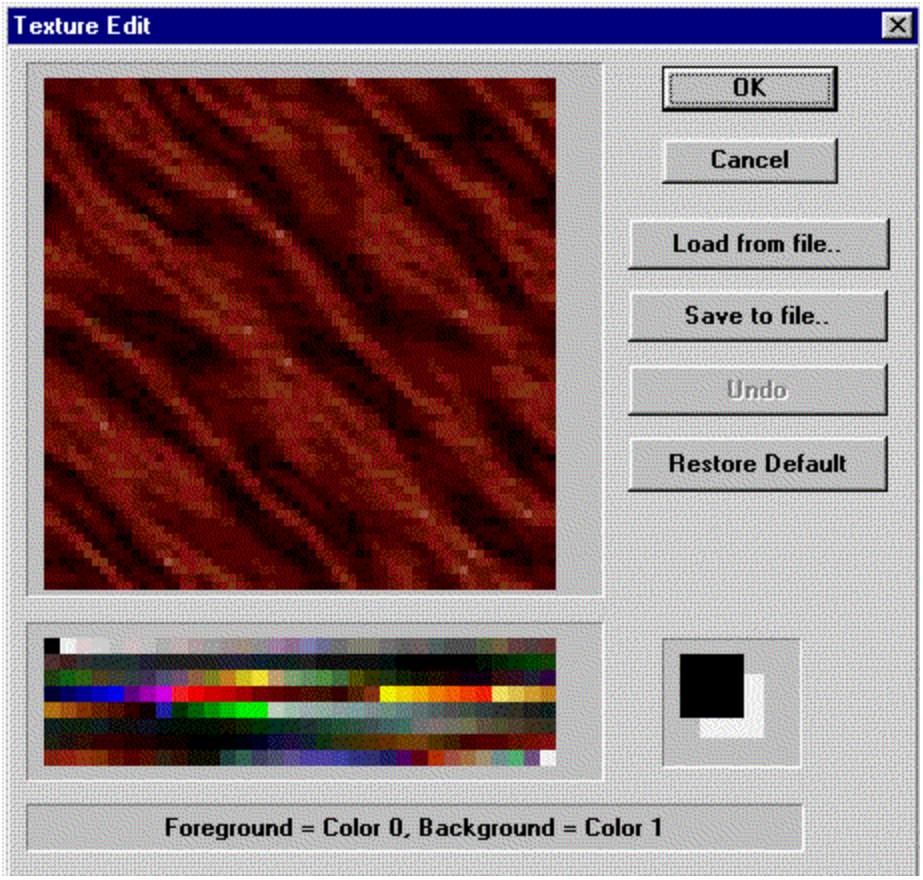
For more information on lights, see:

Exploding Lights Automatic exploding lights

Texture Editing

Texture Edit

Descent II (version 1.2 and above) has a feature which allows you to replace the textures in a level with your own. The mission builder supplies a simple editing tool which allows you to draw into the texture or import/export bitmap files (*.bmp).



Drawing

When you move your mouse cursor over the texture, it changes to a **Pencil** which allows you to draw into the texture. If you press the left mouse button, the foreground color will be drawn. If you use the right mouse button, then the background color will be used.

The **Pencil** colors are shown in the lower right corner of the **Texture Edit** dialog. The color in front is the foreground color. The one in the back is the background color.

You can change the **Pencil** colors by clicking the left or right mouse button on the palette.

*Hint: If you hold the **Ctrl** key down, you can change the **Pencil** color by clicking on the texture itself.*

Restore Default and Undo

The **Restore Default** button will replace the texture with the original bitmap from the game.

The **Undo** button will step back in case you make a mistake.

Loading and saving BMP files

You can save the current texture to a *.bmp file by pressing the **Save** button. This will create a 256 color bitmap file for you to edit with a more powerful paint program.

You can load 256 color *.bmp files by pressing the **Load** button. When the bitmap file is read, the colors will be adjusted to the closest color found in the palette. It is for this reason, that you should try to use the same palette in the bitmap file as is used in Descent. You can do this by saving a texture, modifying it with a paint program, then loading it back.

Palettes

If your paint program can save and load palettes, then you might want to save a texture and extract the palette. You can then prepare the bitmap file for Descent by applying this palette to it. The reason you might want to do this, is because some of the better paint programs (like Photoshop or Paint Shop Pro) allow you to "dither" colors as you apply a new palette. This makes bitmaps look closer to the original than if you were to read it directly into the Mission Builder.

HINT: Certain colors don't import well into the Descent palette. Flesh tones, for example, tend to come out looking yellow and/or orange after you apply the palette. It is best to stick with the color scheme used in the game if you want to get realistic looking results.

Editing

Marking Cubes into blocks

What is a block?

A block is a set of marked cubes. Blocks show up as cyan cubes or red cubes if the **mode select** is set to **block mode**.

How do I mark thee, let me count the ways...

There are several ways to mark or unmark cubes:

Function	Hot key	Description
Mark/Unmark	M	Marks cube if they are unmarked, or unmarks them if the are.
Mark All	Alt+M	Marks all cubes.
Unmark All	Ctrl+M	Unmarks all cubes.

Or you can use the mouse to create a box around the cubes you want to mark. Just press and hold the left mouse button and drag a box around the cubes you want to mark. Then let go.

If you hold down the shift key, then the box will unmark the cubes.

There is one other way you can mark (or unmark) a cube. Simply hold the shift key down and press the left mouse button on the cube you want to mark.

What you can do with marked blocks

The most powerful feature of marked blocks is the ability to create a library of rooms for you level designs. Once you have spent the time creating a complicated room, you can save it as a block file for later use. For more information on how to save block files, see the Cut and Paste section.

Another powerful feature of marked blocks is the ability to move or rotate multiple cubes at once. Simply select "Block" mode on the tool bar and use the numeric keypad to adjust the block.

Hint: Remember to keep the Num Lock key on when using the numeric keypad.

Cut and Paste

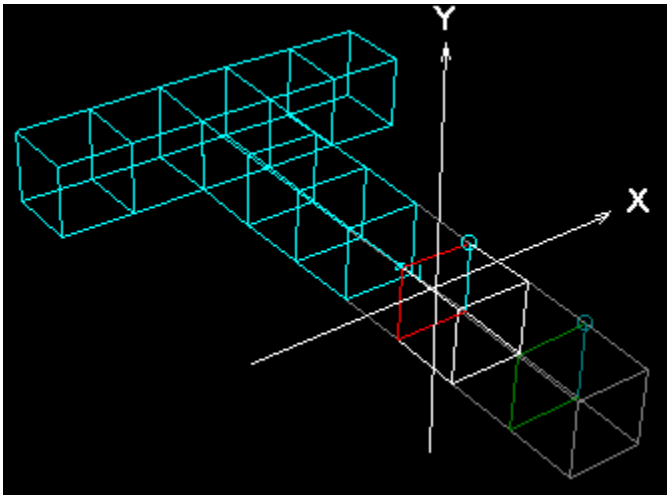
Block files

The Mission Builder allows you to cut and paste a block of marked cubes to a file (*.blk). These files make up a library of blocks for your current level design and future level designs.

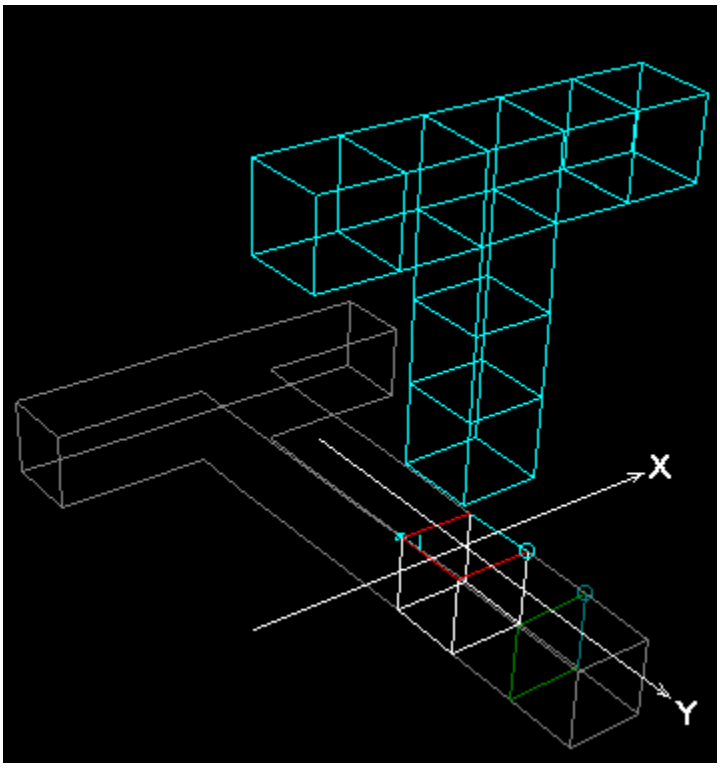
Relative angle and position

When a block file is saved, the Mission Builder uses the current cube as the reference zero. This way you can insert the block back into the level in any angle and position you want.

The reference zero position is defined as the center of the current side. The reference zero angle is the direction the side is facing.



When you paste the block back into a level, you need to select the side and the point where you want the block to be pasted.



Once you have the block pasted, you can then move or rotate it into position.

The last step is to connect the block to the other cubes. You can either use the join sides function or use insert a cube in the direction of the block. I like to use the insert cube function because you don't have to select the sides to join which makes inserting multiple blocks one step faster. The example above illustrates this method.

Hint: Make sure you put a normal size cube at the end of the block before you cut or copy it to a file. This way you can build a bunch of standard block files which can be inserted onto a standard cube.

Creating your first level

Its as easy as 1 - 2 - 3

1) Starting with a new level, press the *Insert* key several times. This will create a hallway for the player to fly down.
2) Select *Robot Object* under the *Insert* menu. This will add a **Class 1 Drone** to the currently selected cube.
3) Choose *Save As* under the *File* menu and save the level as "test1.hog" in the Descent directory (or the Descent II missions directory).

Now start Descent and load the mission called "test1". You should see a straight hallway with red walls and a robot at the end.

Beginner Steps

Stick with square cubes at first and create tunnels in various directions.
Add a group of robots then change the robot id's using the Object Dialog  .

Add several auto closing doors.

Add a fuel center or two.

Add a reactor.

Intermediate Steps

Add a robot maker and create a trigger which sets it off.

Make a few curved tunnels using the Edit Tool Bar (press F5 to bring it up).

Change textures of a cube and copy the texture to other cubes using the F6 key.

Set the lighting of the mine by marking all cubes then use the Light Adjustment dialog from the Tools menu to automatically adjust the light.

Advanced Steps

Add an exit tunnel and an exit door at the beginning of the tunnel. Don't forget to mark the end of the tunnel as "End of exit tunnel" on the Cube Dialog  .

Make a curved tunnel using the Curve Generator.

Align the textures using the Align Textures Dialog (you can get to this dialog from the Texture Dialog by pressing the "Alignment.." button). Select a side to start the alignment from, click on the Align button, then press 'c' to select the next cube. Repeat this step until the end of the tunnel has been reached. Then do the same thing for the other 3 sides of the cube.

Glossary

Click on the word to get a definition:

Alignment
Block
RDL/RL2 File
Child Cube
Coop Player
Cube
Current Cube
Cut & Paste
HOG/HXM File
Hostage
Join
Light
Mission
Object
Other Cube
PIG/POG File
Player
Power Up
Reactor
Robot
Side
Texture
Tools
Trigger
Wall
Weapon

Alignment

Alignment is a way of making the graphics on the walls appear continuous. To align the wall textures, select the Alignment button from the Texture dialog.

Blocks

A block is a group of marked cubes. Blocks can be moved, deleted, or rotated. Blocks can also be cut or copied to a file for later use. *Tip: To mark a cube, hold down the shift key as you press the left mouse button or press the letter "m" on the keyboard or drag a box around the cubes.*

Child Cube

A cube which connects to a particular cube is defined as the child of the cube. Since a cube has 6 sides, it can have up to 6 children.

Coop Players

Each mine has a total of three cooperative players. *Tip: To make an object into a cooperative player, select Object Data under the Tools menu and set the Type to CoOp. Then set the Id to 8, 9, and 10 for the three coop players.*

Cubes

Cubes are joined together to form the inside of the mine. Each cube has 6 sides and 8 points which define the corners of the cube. When cube shares four of the points of another cube, they are considered joined.

Current Cube and Other Cube

The Mission Builder highlights two cubes for editing. The primary cube is called **the "current cube"**. The **current cube** is used for most editing functions. When a second cube is required, the Mission Builder uses a secondary cube called the **other cube**. The **other cube** is used in functions such as joining sides, adding triggers, and the curve generator.

Cut & Paste

Cut and Paste is a way of copying sections of a mine into your new level. First, you cut or copy a section of the mine (called a block) to a block file (*.blk). Later, you can call up one of the block files and stick them into your mine. Cut and Copy only store the shape, light and textures of cubes (not walls, triggers, or types).

HOG/HXM File

Hog files contain missions, briefings, and levels. The DESCENT.HOG also contains a lot of other files used for the game.

HXM files contain the custom robot data created when you modify the robot data in the Robot Workshop dialog (use the edit button on the Object Dialog). These files are stored in the HOG file for the level. You must have one of these files per RL2 file if you want custom robots for that level.

Hostage

Hostages are the dudes you are trying to rescue. You should always include a few of these per level.
Your main hostage should have an Id of zero. The rest should have an Id of one.

Join

Joining is used to connect cube sides, lines or points together. When sides are joined, they share sides (four points). Sides are joined automatically when you "add" a cube and the new points match another cube side's points. You can also Un-join sides, lines, or points.

Object

Each object has type, an id, and a texture. Objects types include: Robots, Hostages, Players, Power Ups, Weapon, Reactor, or Coop Players. The id chooses the sub-type (ex. "Laser" is a sub-type of a Power Up).

Light

Cube Light - Each cube has a light value. This specifies the brightness of polygon drawn objects (robots, reactor, players) that appear in the cube. You can edit a cube's light from the Cube Data dialog.

Texture Light - Each texture has four light values (one for each corner). You can edit the corner light values from the Texture Dialog. You can also automatically calculate these values by selecting "Light Adjustment" from the Tools menu.

Blinking Light - New feature for Descent II version 1.1 and above. Allows you to blink exploding lights.

Mission

A mission contains one or more levels (*.rdl/*.rl2 files). The Mission Builder automatically creates a mission file (*.msn for Descent / *.mn2 for Descent II) for you when you save as *.hog, *.rdl, or *.rl2. You can also use the built in Mission Manager to edit the current mission file.

PIG/POG File

Pig files contain the texture graphics for descent. The Mission Builder uses this file to display the textures in the "Texture Edit" dialog.

A Pog file is a cross between a Pig and a Hog (and boy, was the farmer upset about this one). This file contains the custom textures you can edit for a Descent 2 level. You must have one POG file for each RL2 file if you want custom textures for that level.

Player

Each mine should have eight players with id = 0 to 7 plus three CoOp players with id = 8, 9, and 10.
***NOTE: Object 0 must be player 0*.**

RDL/RL2 Files

RDL (Descent 1) and RL2 (Descent II) files contain the data for a level (also called a mine). RDL stands for Released Descent Level, RL2 stands for Released descent 2 Level. It is a compressed binary file which contains vertices, segments (cubes), walls (doors), triggers, and more.

Reactor

A reactor is the power generator of the whole mine. When the reactor is severely damaged, it will initiate a swquence ending in the desctuction of the entire mine. At this point, the emergency exit door for the level will open, and you have a limited amount of time to find the exit and leave the mine.

Each mine should have only one reactor (you can put more but only the first one will shoot at you). The reactor should be put in a standard cube size (20.0 x 20.0 x 20.0) with the type to "Reactor". When you blow up the reactor, you can set "Reactor Triggers" to open the exit door or other doors which have robots behind them.

Powerup

A powerup is a type of object which is either a weapon, energizer, or key. You may selectively view sub-types of objects by selecting Objects from the View menu. *Hint: this is a good way to find the keys on a level.*

Robot

Robots are a type of object. Each robot has an optional override texture. Each robot also has a behavior setting which changes its personality in the game. To override the object's texture, set Texture1 in the Texture dialog to the desired texture.

Sides

Each cube has six sides. Each side either has a "child" cube, or a texture. If it has a child cube, it can also contain a wall (illusion, door, exit, etc..). If it has a wall then it can also have a trigger which activate other cubes.

Texture

Walls and cube sides have a one or two textures. The first texture is usually the background image such as a rock surface. The second (optional) texture would be a door animation, a ceiling light, a wall animation. Textures with the "- anim" are animation's. Choose these when you want Descent to animate the texture. If no texture is chosen for a cube side which has no child, then Descent will show a gray wall. *Hint: To set the texture of a door, you need to set the "clip" type in the Wall Edit dialog*.

Tools

Descent Mission Builder II contains a number of dialog "tools" to help you edit your mine. These include:

Edit tool bar	moves/rotates/sizes points/lines/sides/cubes/blocks/objects.
Texture edit	sets cube side graphics (also wall starting graphics) and alignment.
Point edit	sets coordinates of current point.
Cube edit	sets cube type, light, robots, walls, and triggers.
Object edit	sets object type, id, size, shield, and sub-objects within the object.
Wall edit	sets wall type, door animation clips, keys, and other wall attributes.
Trigger edit	sets trigger events which control walls.
Light Adjustment	automatically sets side light for all marked cubes.
Align Texture	set the texture orientation on a cube
Reactor Trigger	sets cube/sides which get triggered when reactor is blown up.
Curve Generator	makes a curved tunnel between the *current cube* and the *other cube*.

Triggers

If a cube has a wall then it can also contain a trigger. Triggers point to other cube sides to active them. Triggers can point to up to 10 cube/sides.

For Descent, a trigger can be used to: open a door, mark an exit or secret exit, activate a robot maker, drain your shield or energy, and/or turn on/off an illusion. You can mix and match if you like.

For Descent II, a trigger can be used to: open/close a door, activate a robot maker, mark an exit or secret exit, turn on/off an illusion, lock/unlock a door, open/close a wall, or turn lights on/off. You can NOT mix trigger types for Descent II.

Walls

Cube sides which share their side with another cube can have a wall. A wall can be either: a door, an illusion, or a place holder for a trigger. Several door types are listed in the "Walls" drop down list in the "Cube Data" dialog.

Weapons

A weapon is a type of powerup object (laser, cannon, missile, or bomb).

Technical Support

Interplay Customer Service
Mail: Interplay Customer Service
16815 Von Karman
Irvine, CA 92606

Telephone: (714) 553-6678
Fax: (714) 252-2820
Attn: Customer Service

Internet E-Mail: support@interplay.com
World Wide Web: http://www.interplay.com
BBS: Telnet to bbs.interplay.com
BBS Modem Phone Number: (714) 252-2822
America Online: E-mail IPTECH
Compuserve: GO GAMBPUB or E-mail 76702,1342
FTP Site: ftp.interplay.com