
Worldcraft **t** **User's Guide**

Version 1.0

by Mark Kusec and Rowan Crawford
18 April 2021

Table of Contents

1 Introduction.....

- 1.1 About constructive solid geometry.....
- 1.2 About the interface.....
- 1.3 Learning the ropes.....
- 1.4 Other sources of information.....
- 1.5 Credits.....

2 The Basics.....

- 2.1 Overview of levels.....
- 2.2 Editing in three dimensions.....
- 2.3 Selecting and manipulating objects.....
- 2.4 Grouping and filtering objects.....
- 2.5 Compiling the level.....
- 2.6 Manually compiling the level.....

3 Solids.....

- 3.1 Overview.....
- 3.2 Creating solids.....
- 3.3 Applying textures to solids.....
- 3.4 Carving sections from objects.....

4 Entities.....

- 4.1 Overview.....
- 4.2 Entity keys.....
- 4.3 Creating point entities.....
- 4.4 Creating solid entities.....
- 4.5 Editing an entity's properties.....

5 Tutorial.....

- 5.1 A simple room.....
- 5.2 Adding some features.....
- 5.3 Enhancing the room.....
- 5.4 Joining two rooms.....
- 5.5 VisGroups.....
- 5.6 A moving door.....
- 5.7 A teleporter.....
- 5.8 A moving platform.....
- 5.9 Frequently Asked Questions.....

6 Menus.....

- 6.1 File Menu.....
- 6.2 Edit Menu.....
- 6.3 Map Menu.....
- 6.4 View Menu.....
- 6.5 Tools Menu.....
- 6.6 Window Menu.....

7 Toolbars.....

- 7.1 MapView Bar.....
- 7.2 MapOperations Bar.....
- 7.3 Object Bar.....
- 7.4 MapTools Bar.....
- 7.5 Filter Control (VisGroups).....
- 7.6 Texture Bar.....
- 7.7 Status Bar.....

8 Windows.....

- 8.2 Paste Special.....
- 8.3 Run Map in Quake (F9).....
- 8.4 Texture Application (Shift+A).....
- 8.5 Options.....
- 8.6 Properties.....

Note: This document is best viewed in MS Word's "Page Layout" view.

1.1 Introduction

Welcome to Worldcraft, a graphical level editor for id Software's, Quake. With Worldcraft, you can make your own Quake maps quickly and easily! Worldcraft is written by DOOM Construction Kit author, Ben Morris.

1.1.1.1 About constructive solid geometry

1 Building blocks

Worldcraft uses a technique called Constructive Solid Geometry (CSG) to create maps. CSG is based on the simple idea of using building blocks - cubes, cylinders and wedges - to create more complex shapes such as a six-walled room. In Worldcraft, these building blocks are called **solids**.

If you ever did any DOOM editing, you'll remember that everything was made up of lines. In Quake level editing, there are no isolated lines - each object must be a three-dimensional object with volume. A six-walled room, then, is made up of six cubic solids - one for each of the four side walls, and two for the ceiling and floor. **This concept is simple, but vital.**

To view an example of a Quake room, load Worldcraft and Open the file ROOM.MAP. This map contains two areas: a sealed room, and a collection of blocks used to make the sealed room. The blocks are moved into touching positions - in fact, they can freely intersect - and the resulting enclosed space is what's seen by the players in Quake.

In fact, any solid can be used to **carve** a volume out of other solids. For example, you can place a solid within the room's wall and tell Worldcraft to subtract the solid from the wall - effectively punching a hole right through the wall! This feature is not limited to using cubic shapes as a carving tool: any solid - cylinders, cubes and wedges - can be used to carve other solids.

And of course, you can place all your favorite Quake monsters, items, weapons and other entities in your new maps with Worldcraft.

1.1.1.2 About the interface

The Worldcraft interface has been designed to maximize ease of use. The help is divided into 3 basic categories; menus, buttons and windows.

Many of Worldcraft's features can be accessed by various different means; for example, 'Ungroup' can be selected from either the menu, the onscreen button, by pressing the right mouse button (RMB) over the selected object or by using the Ctrl+U hotkey. This should allow everyone to find a particular workflow method suited to their style of editing.

Likewise, the arrangement of buttons and windows can be customized to your particular taste. Most users will find the default arrangement quite suitable and it's recommended to at least become familiar with the program before thinking of customization.

1.1.1.3 Learning the ropes

The fastest way to become familiar with Worldcraft and creating Quake levels is to follow through the **Tutorials** that are included in these help topics. Or, read through the User's Guide, starting with the section entitled **Overview of levels**. As you are working with Worldcraft, you can use the Reference section for help with specific dialog boxes, menus, tools and utilities.

1.1.1.4 Other sources of information

1 Worldcraft's web site

The most up-to-date releases and information can be found at the Worldcraft web site:

<http://www.islandnet.com/worldcraft/>

2 Quake map specs

An excellent source of Quake editing information can be found at:

<http://www.infi.net/~nichd/qmapspec.html>

Note: Microsoft's DirectX3 must be installed to run Worldcraft. DirectX3 drivers can be found at either the Worldcraft web site or any Microsoft source.

1.1.1.5 Credits

Worldcraft is a trademark of Ben Morris, and is Copyright © 1996 by Ben Morris. All rights reserved.

This documentation and the help file were written by Rowan Crawford and Mark Kusec, Copyright © 1996. All rights reserved. Produced and edited by Mark Kusec. Used with permission by Ben Morris.

Please send comments or questions regarding this documentation to Mark Kusec at mkusec@direct.ca.

Rowan Crawford can be reached at sumaleth@starfury.apana.org.au.

1.2The Basics

1.2.1.1Overview of levels

Quake level editing is based on a concept completely different to usual 3D editing and so while it is surprisingly simple, the immediate hurdle is realising that it truly is as simple as it looks. Indeed, the hardest aspect of creating a level is coming up with the design itself.

Levels are made from two types of objects: **solids** and **entities**. These two object types are used in completely different ways, with solid usage being exceedingly simple, and entity usage being somewhat more involved. Together, solids and entities make a powerful editing system, yet, after a little practice with them, they are surprisingly easy to use.

1Solids: your building blocks

A solid is a fully enclosed convex object which is used to create the level itself along with things such as moving platforms and doors. Worldcraft has four standard types of solids - block, wedge, cylinder and spike - but it really doesn't matter what shape the solid is, so long as it is fully enclosed and has no concave elements.

Often, solids will be grouped together to form a more complex object than the 'convex only' rule allows. When using the 'carve' tool on a solid, you will often find it has been sliced up into multiple solids automatically (all grouped) to avoid concave sections.

Solids can overlap each other without a problem and have no restrictions on size, position and orientation. This freedom is what allows Quake levels to have a much more interesting design than those in Doom, but keep in mind that the number of solids determines how long the level will take to QBSP and VIS, and how fast it will play in Quake.

2Entities: your level's features

An entity is anything that isn't a solid, or is a solid that does something special. Entities range from monsters, doors, trains, weapons, switches, wind plus a multitude of other functions. They are used to make the levels more interesting than simply a walk from the start to the end.

Entities generally have their own default behavior, but many can be changed by editing or creating "keys" or by altering "flags".

These two object types are used in completely different ways with solid usage being exceedingly simple, and entity usage being somewhat more involved. But as with everything, once you've done it a few times it becomes second nature.

3The necessities

A map really only needs two things to be playable in Quake; an area enclosed by solids, and a player start point. Anything after that is really just a bonus.

Lights are important in making a level look good, and as with everything else, once the concept is understood they are very powerful. There are quite a few

light_entity types, covering fluorescent, strobing, normal and even some wall torches complete with prepackaged wall attachments.

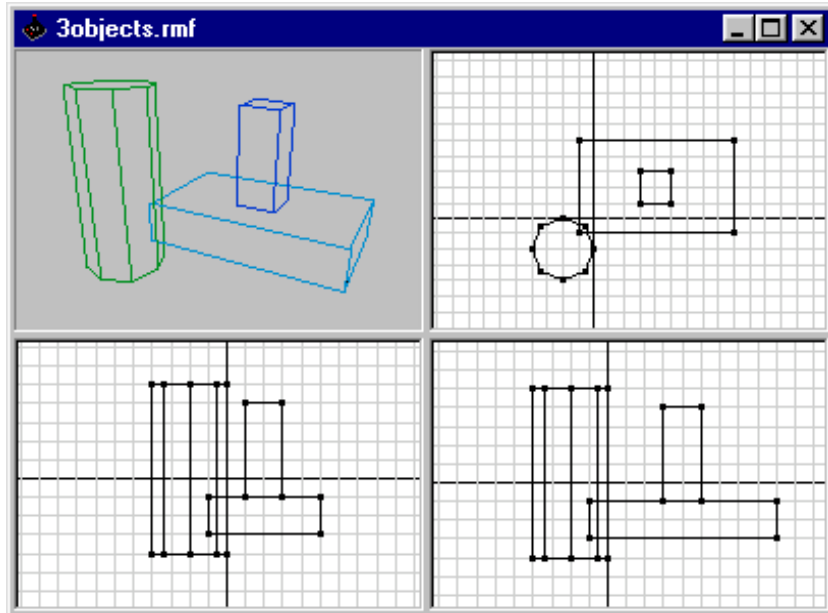
To get anything approaching real daylight, you'll be using a LOT of lights. This will slow down LIGHT calculating (see **Compiling**) but that is nothing to fear. Most Quake levels tend to stick with fairly dark lighting anyway.

If you're making the level for deathmatch or co-op play, be sure to place plenty of deathmatch and co-op start points in appropriate places. One point to watch, for deathmatch in particular, is to avoid placing start points in areas where there will be a lot of action - telefragging is generally considered annoying.

1.2.1.2 Editing in three dimensions

1 Viewing your level

Worldcraft uses a fairly standard way to **display your level**: three 2-dimensional views and one 3-dimensional camera view. You'll do most of your editing in the 2D views, and the camera view is useful for getting a good view of your map before you test it in Quake.



These are Worldcraft's 4 map views. Clockwise from top left: 3D view, top view (XY), side view (XZ), front view (YZ).

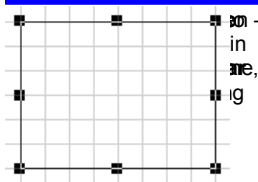
The 2D views each offer a different perspective on your map: a Top (bird's eye) view, a Side view, and a Front view. Since each view offers a different representation of your map, you can shape objects in all three dimensions.

2 Navigating in the 2D views

Each 2D view has two scrollbars that can be used to move the area you're looking at. You can also use the arrow keys to scroll the Active view.

To change the Zoom factor for the Active view, use the gray + and - keys on the right side of your keyboard, or use the Magnification tool (the magnifying glass icon on the left-hand side of the screen) - click the left button to zoom in, and the right button to zoom out.

The 2D views will also display a grid - you can change the size of the grid with the [and] keys (European users can use the A and S keys.) By default, objects



A box with nine handles.
Handles allow you to adjust
the shape of the box in one
or two dimensions at a time.

that you move and make are snapped to the grid - a common feature among CAD programs that makes alignment much easier. You can turn off this feature with the P key, or by using **Map|Snap to Grid**. You can also turn the grid off with **Map|Show Grid**.

3Navigating in the 3D view

To navigate in the 3D view, select the **camera tool**.

- To move the camera without changing its angle, hold the right mouse button (**RMB**) in the 3D view while moving the mouse.
- To rotate the view and to look up and down, hold the left mouse button (**LMB**) in the 3D view while moving the mouse.
- To move the camera forward and backward, hold **Shift+RMB** and move the mouse up and down, or use the **D** and **C** keys.

1.2.1.3Selecting and manipulating objects

Manipulating objects, whether they be solids or entities, is the whole purpose of the four view windows and is designed to be intuitive in the Worldcraft interface.

Object manipulating only occurs on selected objects/groups. Selected object(s)/groups have a bounding box drawn around them (red line) in the 2D windows and are rendered as red in the 3D view. By clicking the LMB in the center of a selected object, the manipulation type changes with the bounding box style changing accordingly. The order of type is; move/resize, rotate, shear. Note that point entities can not be resized, rotated or sheared.

1Selecting solids and entities

To select an object:

1. Choose the **selection tool**.
2. Click on one of the object's lines in the 2D views. Or, if you have a clear view of the object in the 3D view, click on one of its faces.
3. To select multiple objects, hold down the Ctrl key while selecting each object.

The selected object will be highlighted in red in the 3D view, and a selection rectangle - a box with **handles** - will surround the object in the 2D views. When multiple objects are selected, your editing will apply to all the selected objects.

2Switching between scaling, rotation, and shearing modes

There are three selection modes: scaling, rotating, and shearing. Clicking the left mouse button repeatedly on the selection rectangle will toggle between

Tip: Hold down the Alt/paste object to enter different maps! disable the snap-to-grid feature while you're moving or scaling the objects.

these three modes. Using the selection rectangle, you can scale/move, rotate, and shear (slant) the selected objects:

- When you first select an object, it is in **scaling** mode, and has **nine handles**.
- Click on the selection a second time to enter **rotation** mode. It now has handles on the **corners only**.
- Click on the selection a third time to enter **shearing** mode. It now has handles on the **edges only**.

3 Moving and scaling

To move the object(s) to a new location, click and hold the left mouse button inside the red bounding box, but not at the edges. Then drag it to a new location and release the mouse button. If grid snapping is on, the movement will be snapped to the bounding box.

To scale the object(s), move the cursor over one of the handles and click and hold the left mouse button. Then, drag the box to resize it to the desired proportions and release the left mouse button. The 2D and 3D views will be updated to reflect the change in size.

4 Rotation

To rotate the objects, click and hold the left button on one of the four corner handles. Drag your mouse around to rotate the object, and release the left button when you've rotated it to the desired angle.

If Tools|Options|2D Views has the 'Default to 15 degree rotations' set, then all rotations will be snapped to 15 degree steps. This can be overridden by holding down Alt while rotating. If that option is set to off, 15 degree snapping can still be achieved by holding down Shift while moving.

5 Shearing

To shear the objects, click and hold the left button on one of the four side handles. Drag the mouse left/right or up/down to shear the object, and release the left button when you've sheared it to the desired angle.

6 Copying, Cutting, Pasting and Deleting selected objects

These standard features are available through the **Edit menu**.

7 Accessing the object's context menu

Clicking the RMB inside a selected object opens up a menu which echoes many of the same commands available in the pull down menu and the various buttons menus.

1.2.1.4 Grouping and filtering objects

1 Collecting objects into groups

It can become very confusing to try to manipulate numerous objects at the same time. To alleviate this, Worldcraft allows you to group objects together

Note: Filtering can be a convenient way of organizing objects for editing purposes in Worldcraft's Object Properties window. Objects that are hidden by filtering will remain present in the compiled level.

so that they can be manipulated as a single object. Grouped objects will move, scale, rotate, and shear together as a single object.

- To group the objects you have **selected**, press the **Group** button.
- To ungroup the selected object, choose the **UnGroup** button.

2Organizing objects into VisGroups

Furthermore, you can collect your objects into "VisGroups", or "visible groups". You can then specify which VisGroups are visible in Worldcraft's edit windows. This allows you to ignore extraneous areas of the map and focus on the objects that you are currently working on. The **filter control window** shows a list of all VisGroups that have been defined.

To create a new VisGroup:

1. Press the "Edit Groups" button in the filter control window.
2. In the filter control window, press the "New Group" button.
3. Provide a name for the new VisGroup and choose its color.

To add objects to one or more VisGroups:

1. **Select** the objects that you want to be added.
2. Open the Object Properties window.
3. On the VisGroups page, choose which VisGroups these objects should belong to.

3Choosing which VisGroups are visible

To enable filtering by VisGroups, put a checkmark beside **Filter by VisGroups** and then click the Apply button.

With filtering enabled, any VisGroups that are not checkmarked will be made invisible in the 2D and 3D views. You can toggle a VisGroup's status at any time; to view the results, click on the Apply button.

1.2.1.5Compiling the level

1Running the map

Editing Quake levels is done in a standard known as .MAP. Quake, however, requires that the level be compiled into .BSP format.

All Quake editing tools are able to read and write the .MAP format. Worldcraft is no exception, although it also has an additional file type known as .RMF which is used to save additional information that isn't part of the .map standard (such as VisGroup information). See the **File menu** section for details on these formats and their usage.

Note: Whereas QBSP will work even if the level has leaks, VIS requires that the level be completely sealed from the poisonous gas of the gray void.

To build your map into a playable BSP file, press the **F9** key, or choose File|Run Map in Quake. The **Run Map dialog box** will be displayed, allowing each of the following programs to be run at your discretion.

2QBSP

The .MAP file describes all the solids, entities and textures that comprise the level. It is compiled into a .BSP file by the program called QBSP. A .BSP file can then be loaded into Quake and played without further ado. There are, however, two further - optional - steps you can take you make your level better: VIS and LIGHT.

3VIS

If you run a .bsp file after QBSP'ing it, you'll find it generally slow and flickering, especially in larger levels (in fact, entire areas could be gray). This is because Quake is rendering every face which is in it's view, even if that face might be halfway across a level behind ten solid walls.

VIS optimizes a level by making it so that Quake only renders what it can see from any point in the map, thus making the level run a lot smoother. The trade off for having speed in the level is that VIS takes a LONG time to process the file, especially on a large level.

On a slow PC and a large level, you may well be looking at days for VIS to complete. For this reason, VIS is usually used only when the level is considered finished and ready to be released, unless you have access to a fast machine and wish to check it out at full frame rate more often.

4LIGHT

Finally, you can run LIGHT which will calculate shading and shadows for the entire level. If LIGHT is not run, all textures will be fully bright, so LIGHT's main use is to give the level atmosphere.

1.2.1.6 Manually compiling the level

Although the Run Map dialog box is convenient, it's still handy to know how to do it manually in a DOS window to see error messages and such. On a computer with little memory, you may prefer this approach.

1Running the map without Worldcraft

The Run Map feature might not work on all computers, since QBSP (the included program that turns MAP files into BSP files) requires a lot of memory. If you find that the feature does not work, here's how you can do it yourself:

1. Choose File|Export to Quake MAP. This will save your level as a Quake MAP file instead of a Worldcraft RMF (Rich map format) file.
2. Exit Worldcraft, and bring up an MS-DOS Window (use the Windows 95 Start menu, choose Programs, then MS-DOS Prompt.)
3. Change into your Worldcraft directory (for example, type cd\worldcraft.)

Note: QBSP's `-fast` parameter is optional. If you use the `-fast` parameter, VIS the level directly to the target as fast as possible. If you do not use the `-fast` parameter, VIS the level directly to the target as fast as possible. This is very easy to load into Quake with 'map name'.

4. Run RUNMAP.EXE with your map file. RUNMAP takes two parameters: one, the name of your map file, and two, your quake directory. For example, if you were working with a file named mymap and your copy of Quake was located in d:\quake, you'd type:

```
c:\worldcraft> runmap mymap d:\quake
```

2Running the utilities manually

If you prefer to run each utility separately, then you must invoke them yourself on the command line. Although the level can be used at any time after QBSP has created the .BSP file, the utilities should be executed in the following order. Their usages are as follows:

1. QBSP c:/location/name.map c:/location/name
2. VIS [-fast] c:/location/name.bsp
3. LIGHT c:/location/name.bsp

Not every face of a solid is a triangle. In fact, a solid can be any shape, but every face of the solid must possess the same texture. Otherwise the level will not compile.

1.3 Solids

1.3.1.1 Overview

A solid is a fully enclosed convex object which is used to create the level itself along with things such as moving platforms and doors. Worldcraft has four standard types of solids - block, wedge, cylinder and spike - but it really doesn't matter what shape the solid is, so long as it is fully enclosed and has no concave elements.

Often, solids will be grouped together to form a more complex object than the 'convex only' rule allows. When using the 'carve' tool on a solid, you will often find it has been sliced up into multiple solids automatically (all grouped) to avoid concave sections.

Solids can overlap each other without a problem and have no restrictions on size, position and orientation. This freedom is what allows Quake levels to have a much more interesting design than those in Doom, but keep in mind that the number of solids determines how long the level will take to QBSP and VIS, and how fast it will play in Quake.

1 Building the level with solids

Solids are used to construct the physical world that Quake is played in, so the way to treat solids is as you would treat a piece of wood or some bricks.

The advantage of this concept is that it allows solids to intersect each other without causing any slowdown once in Quake. This is because QBSP cuts away all intersecting sections, leaving just the polygons that will actually be seen in the game.

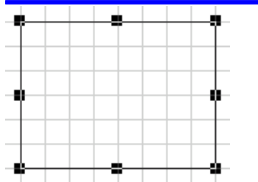
This freedom to intersect removes the necessity to make every wall and point of contact match up perfectly. In fact, it's often considered a good idea to deliberately overlap solids because it will save problems later.

The entire level must be completely self contained, with no gaps - no matter how small - reaching into the gray void of outside. QBSP will probably still work on a level with a leak, but it will not produce a .ptr file and therefore the level cannot be processed by VIS.

2 Texturing the level

Texturing plays a large part in giving a level the quality appeal as those found in Quake. Texturing turns out to be an equally simple task, especially with Worldcraft.

Quake episodes are combined into .PAK files which are located in the /quake/id1 directory. Inside these PAK files is another file called a WAD2 file which holds all the textures for the episode. These WAD2 file can extracted from the Quake PAK files, or downloaded from various Internet locations such as Worldcraft's WWW site.



A box with nine handles.
Handles allow you to adjust
the shape of the box in one
or two dimensions at a time.

Quake has four such WAD2 files, one for each episode. You can also add in your own textures, or remove some just as easily (you'll need a program to do this). Worldcraft lets you use multiple WAD2 files to texture your level, although to use this ability (which is not part of the Quake standard) you will need to use the special version of QBSP supplied with Worldcraft. If you only plan to edit using one WAD2 file, then you can use any version of QBSP that you wish.

When QBSP compiles the level, it will extract the textures you have used in your level from the WAD2 file and add just those to the resulting .BSP file (which can then be played in Quake).

Worldcraft makes texture placement quite simple. Every face of a solid can have its own texture, with accompanying X/Y scaling values, X/Y offset values and a rotation value. For details, see **Applying textures to solids**.



1.3.1.2 Creating solids

1 Overview

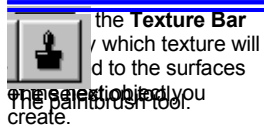
The steps to creating a solid are as follows; they are discussed in greater detail in the following paragraphs.

1. Select the **block tool**.
2. In any 2D view, hold the left mouse button and drag out a bounding box. The box is reflected in the other 2D views.
3. Using the three 2D views, adjust the bounding box's width, depth, and height until it is the desired shape.
4. Choose the type of solid you want to create using the **object toolbar**, and choose its texture using the **texture bar**.
5. Press Enter to create the solid.

2 Creating a bounding box for the solid

To create a solid, you must first make the box that the solid will be contained in. This allows you to specify the size and position of the solid before it is created.

Create the initial box in any of the 2D views by holding down the left mouse button and dragging the mouse pointer. After you release the left button, a box will appear in the three 2D views. The box will have **handles** at each corner and on each side.



3 Adjusting the shape of the bounding box

Because you have not yet specified its third dimension, the box will initially have default thickness. You can adjust the box as many times as you like using the handles in the three views.

To adjust the size of the bounding box:

1. Place the mouse pointer over top of one of the handles. The pointer changes to a +.
2. While holding the left mouse button, drag the handle to the desired position for that edge or corner.

When you are satisfied with the bounding box, press **ENTER**, or click the right mouse button and choose **Create Object**. The cube will appear in the 3D view. Try moving around in the 3D view (see **Editing maps in 3D**) to view the cube from all sides.

4 Choosing a basic shape

This version of Worldcraft can create four types of solids: blocks, wedges, cylinders, and cones. Use the **Object Toolbar** to select a type of solid before you press **ENTER** to create the object. If the type of solid you've selected can have a variable number of faces, then the rightmost field will allow you to specify the number of faces you want to create.

1.3.1.3 Applying textures to solids

1 What are textures?

Textures create the look of quake. Basically they are the pictures that are stuck on the wall to make them look like brick, wood, metal etc. The textures are divided into different themes. You can get them all from the Worldcraft web page. Put the files into your Worldcraft directory and then add them to Worldcraft's texture collection via the **Tools|Options** window.

2 Choosing a single texture for an entire solid

To specify a texture for every face of an object:

1. **Select** the solids you want to re-texture.
2. Use the **texture bar** to choose the new texture for these solids.
3. Click the **paintbrush tool**.

Textures chosen in this manner will be applied to every face of the solid. For greater control over the textures on each face of an object, you must use the texture application tool.

3 Choosing a different texture for each face of a solid

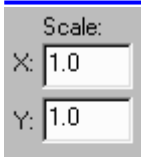
To apply a different texture to each face of an object, you must use the texture application mode. To activate this mode, choose the **texture application tool** from the toolbar, or select Texture Application from the Tools menu. You can then select individual faces of objects directly in the 3D view.

The texture application tool is not available when you're in texture application mode.

To apply a texture to an individual face:

1. Select the **texture application tool**.
2. Select the desired faces by choosing them in the 3D view.
3. Use the **Face Properties** window to choose the desired texture for these faces.
4. Press the **Apply** button.

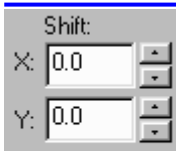
Note: The texture application mode should not be confused with the normal editing mode. Most commands are not available while the face properties window is visible. You should leave texture application mode when you are done adjusting textures.



4Scaling a texture

The **Scale controls** on the **Face Properties** dialog box allow you to stretch or compress the texture bitmap in each axis on the face of the solid. You can specify the scale factor for each of the X and Y axes separately. Scale factors less than 1 reduce the size of the texture bitmap; larger scale factors increase its size.

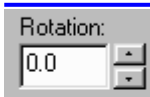
Note: The X-Y scaling axes are oriented at the rotation angle that you specify. Therefore, if you rotate the texture (or the solid) after you have scaled the texture, additional adjustments to the texture scaling may be required.



5Shifting a texture

The **Shift controls** can be found on the **Face Properties** dialog box. In these two fields, you can specify the number of pixels to offset the texture.

Note: Texture offsets are measured with respect to the gridlines, not the edges of your object. Therefore, if you move the object after it has been created, the texture offset may no longer be as desired, and may require further adjusting.



6Rotating a texture

The **Rotation control** on the **Face Properties** dialog box allows you to specify the angle of rotation for the texture.

Note: If you rotate the object after it has been created, the texture will NOT rotate with it; therefore, additional adjustments to the texture's rotation may be required.

1.3.1.4Carving sections from objects

1What is carving?

Worldcraft has a powerful feature called **carving**. Carving is the act of deleting the shape of one solid from another. It's somewhat akin to using a cookie cutter to shape solids. Using subtraction you can create all sorts of other complex solids.

Consider that you've made a six-walled room and want to connect a hallway to it - you can't simply stick the hallway onto a side of the room, because the room's walls are solid; there's no way for the player to pass through the wall

The carve tool.



The selection tool.

into the hallway. This is where carving comes in: you can place a solid within the room's wall and tell Worldcraft to subtract it of the wall - effectively punching a hole right through it!

2Carving other objects using the selected object

Any solid can be used to carve a volume out of other solids - cubes, cylinders, spikes and wedges. To carve using an object:

1. **Select** the object that you want to use to carve other objects.
2. Position the object so that it intersects the other objects.
3. Click the **carve** tool, or choose Tools|Carve from the pulldown menus.

Note: Worldcraft will split a solid into multiple objects if the carving would produce a concave shape. The multiple objects are automatically joined into a single group.



Tip: For instructions on creating standard entities, see [Creating standard entities](#) (section 4.4).

1.4 Entities

1.4.1.1 Overview

An entity is anything that isn't a solid, or is a solid that does something special. Entities range from monsters, doors, trains, weapons, switches, wind plus a multitude of other functions. They are used to make the levels more interesting than simply a walk from the start to the end.

All entities are QuakeC functions which are mostly organized into categories such as "trigger" and "func" (although some, such as "path_corner", belong to no obvious group). While entities are QuakeC functions, you don't need to have any knowledge of QuakeC to use entities.

Entities generally have their own default behavior, but many can be changed by editing or creating "keys" or by altering "flags". For details, see **Overview of keys** below.

Entities are somewhat more involved than solids since they cover a much wider range of features. Entities can be created in two ways which give two distinct results, but unfortunately the only sure way to know which type to use is through experience and common sense.

1 Standalone entities

These are entities that have no solids attached to them. Examples of these are monster_entities, player start points and path_corners (which define the path of a func_train, i.e. a moving platform).

If these types of entities are created with attached solids (see below), the results can be unpredictable; in some cases it won't make a difference, and other times they simply won't work.

2 Entities with attached solids

These are entities that use part of the physical level (i.e. one or more solids) as part of their functioning. Examples of these entities are func_door, func_plat, trigger_multiple etc. The attached solids will not always be visible once attached to an entity also, such as with any of the trigger_ functions.

These entities are actually created by taking a solid (or more commonly in the case of doors, a 'group') and using the 'toEntity' button or the 'Tie to entity' tool in the menus. This will create an entity and attach the selected solid/group to it, at which point an entity type can be selected.

Some entity types will use the shape of the solid for some purpose, but in many cases, the solid itself will appear invisible once played in Quake. The trigger_entities are example of this (although there is one type of trigger in the func_types which does leave the trigger visible) as is the entity that creates the exit portal.

Tip: See the section entitled **Properties** (section 8.6) to learn how Worldcraft handles entity properties.

3Entity keys

Entities have additional properties, called 'Keys' or 'Keyvalues'. Keys accept many different value types depending on the entity used and the particular key. For more information on keys, see **Overview of entity keys** below.

1.4.1.2Entity keys

There are quite a few keys that are consistent across many of the entity types such as 'target', 'targetname', 'wait' and 'sound', although their purpose can change slightly from entity to entity.

You are strongly encouraged to read the 'Quake Map Specs' file located at: <http://www.infi.net/~nichd/qmapspec.html>. This file is the best guide to knowing which keys can be used with what entities (although it shouldn't be taken as 100% correct).

Some of the more common (and important) key types follow:

1targetname

This key is used if you wish to make the entity activate via some other event. A good example is a func_door that you wish to be opened by a switch instead of by proximity (its default). Do not give this key a value if you want the entity to work of its own accord. The 'targetname' key is very closely related to the 'target' key (see below).

The accepted values for a targetname key is any piece of text you wish to identify it by (not starting with a number). id's Quake maps use very simple targetnames like 't1' or 't12' whereas you might find it more helpful to use descriptive names such as 'door1' etc.

You can give several entities the same targetname, meaning that when the trigger happens (whatever it may be) they all activate at once.

2target

Use this key if you wish to make this entity trigger another entity. Continuing the example from above, this entity could be a trigger_multiple which, when triggered, makes the func_door open. The value of the key should be exactly the same as the 'triggername' used in the entity you wish to trigger. You can give several entities the same target value, so for example you could have several switches open the same door.

Triggering is fully automatic in Quake, so once one entity has a target name, and another entity has the corresponding target, it will all work. The type of entity used as the trigger (or to be triggered for that matter) can be almost anything. You can make a door opening turn on a light for example. This targetname/target relationship is the real power behind Quake's level editing.

3wait

This key can be used in many of the entity types to tell it to stay open (or whatever) for a certain number of seconds. The value can be anything starting at 0.01.

Note: This is a screenshot of the Entity Properties window. The text is mirrored and difficult to read, but it appears to describe the 'Entity Properties' window.

4delay

Similar to wait, except it generally creates the pause at the beginning of the action. So we may have our trigger opening the door, except by giving the func_door a 'delay' key the door can be made to not open for a certain amount of time.

5speed

The speed at which the entity moves.

6angle

This is the direction in which the entity acts. The usage varies greatly across the entities, with it being the direction a monster_ starts facing and the direction a func_door opens. Possible values range from 0 (right) through to 360, with special values of -1 (vertically up) and -2 (down). Quake doesn't currently support up/down movement at angles except with func_train entities.

7lip

Lip is one of those bizarre multipurpose keys that turns out to be quite useful. A value of -1 (default) on a door, for example, will open the door (in the direction of the angle) by a distance of exactly the solids width in that direction. A positive value will make it open up less than its width, and large negative values will open it more than its width.

There are many other key types, so it's worth reading the 'Quake Map Specs' as mentioned above. Many entity types also have switches which, for example, tell that entity to only show up in deathmatch play etc. These switches are set in the second folder of an entities properties window.

1.4.1.3 Creating point entities

You create a monster or other entity (herein called items) in much the same way as you create a solid - the difference is that an item requires you to specify just one 3D point. Items also have properties, such as facing direction, which can be adjusted later.

To create an item:



1. Select the **Entity Tool**.
2. Use the **Object Toolbar** to choose the type of item that you want to create.
3. In any 2D view, click on the desired location for the item. The item's volume is reflected in all of the 2D views. Use the three 2D views to adjust the location of the item.
4. Press Enter to create the item.
5. Press Alt+Enter or choose Edit|Properties to access the item's **properties**.

The list of entities is long, and includes ambient effects (light and sound sources), spawn and teleport locations, monsters, weapons and other powerups. Each type of entity fills a different volume of space; the entity appears in the 2D and 3D views as a cube of that volume.

Note: Why can't we already enable SmartEdit? The answer is a solid entity using the Worldcraft tool might create their own Items using Quake's built-in programming language, and Worldcraft can't know about them without some customization to its configuration files. Therefore, it makes sense to allow people to edit the raw keyvalues during the development of these items.

For a description of entities and their keys, see **Overview of entities** (section 4.1).

1.4.1.4 Creating solid entities

Moving solids are actually entities that use one or more solids as part of their functioning. These entities are actually created by taking a solid (or more commonly in the case of doors, a 'group') and using the 'toEntity' button or the 'Tie to entity' tool in the menus. This is just one example of creating a solid entity.

1 Creating a solid entity

Some entity types will use the shape of the solid for some purpose, but the solid itself will not always be visible in Quake. The trigger_ functions are examples of such entities. Other examples of solid entities are func_door, func_plat, trigger_multiple, and so on.

To make a solid entity:

1. **Select** the solid or solids that you want to be moveable.
2. Click the **toEntity** button at the toolbar, or use Tools|Tie to Entity.
3. Choose an entity type.

This will create an entity and attach the selected solid/group to it, at which point an entity type can be selected. The solid(s) are now all part of the entity, and when you select one, all of them are highlighted.

1.4.1.5 Editing an entity's properties

Once you've placed an Item, you can edit its properties - you might want a Monster to appear only on the most difficult skill level, for example.

1. **Select** the item you want to edit.
2. Press Alt+Enter or choose Edit|Properties to edit its attributes.

The **Object Properties** dialog box will be displayed. Using this dialog box, you can change the item's type (its class), the angle it faces, and various attributes called keyvalues. Some classes don't require any keyvalues at all; others have several.

1 Using the SmartEdit view

For those items that do require keyvalues, Worldcraft has a feature called **SmartEdit** that allows you to edit the values without remembering the

Note: The Not in... flags apply to many objects besides monsters - for example health packs and shotgun shell packs could be made more abundant in the easier skill levels.

required keys. When SmartEdit is enabled, the Keyvalues listbox changes to a list of Attributes that are valid for the item's class.

To activate the SmartEdit view, press the SmartEdit button on the Object Properties dialog box. Compare the **SmartEdit view** to the **default view**.

2Setting flags for an entity

Some entities allow you to specify a set of flags, each of which is either on or off. A typical entity's Flags tab looks like **this**.

To access an item's flags:

1. **Select** the item you want to edit.
2. Press Alt+Enter or choose Edit|Properties. The **Object Properties** dialog box is displayed.
3. Choose the **Flags** page.

Each type of item has its own set of flags with unique meanings. For example, the flags for the item monster_army (the shotgun trooper in Quake) have the following meanings:

Ambush, when checked, means that the Monster will only start to attack you after it sees you. When it's not checked, the Monster will try to attack you as soon as it hears your gunshots.

Not in Easy, when checked, means that the Monster won't appear in the Easy skill level. This setting, combined with the other skill flags - **Not in Normal** and **Not in Hard** - allows you to make your maps more difficult for people who think they can take it!

Not in Deathmatch means that the Monster won't appear in Deathmatch games.

Note: This is a list of WADs (Worldcraft Add-on Data) that are available for download from the Internet. These WADs are listed in the Worldcraft directory, and in the Tools|Options menu this WAD should be listed in the path area at the bottom.

1.5 Tutorial



1.1A simple room

1 Creating a new map

This tutorial will show the minimum steps needed to create a new map for Quake. This will be the most descriptive of all tutorials with the others requiring this one to be working.

1. First we must create an enclosed area from which to sculpt a level. From the toolbar on the left of layout, select the Block Tool.
2. In the top view (default position is top right), click and hold the LMB somewhere in it, then drag the mouse down/right until you have a box of around 6x6 grid lines. Let go of the LMB.
3. Now look at either the front or side views (bottom/right and bottom/left respectively with default windows). You will probably see a 1-grid-thickness box with solid black squares at it's corners and along each side. If you can't see the entire box, scroll around the view using the scroll bar to the right of the window, or use the cursor keys on your keyboard.
4. Click the LMB on either the top-middle black square (when the cursor is over a square, it will change to show what type of move you can do - you want it to become a little vertical line with arrows top and bottom). With the LMB held, move the mouse upward until your side view box is around 5 grid lines high.
5. Currently there is only the guidelines of a box in the modeling views - no solid has been created yet. Before we create the box, you may want to reposition the guide-box by clicking and holding the LMB inside the box and moving around (it will snap to the grids). Try to center it around the center of the two dark black lines that run through each view (ie. the origin of the map). This isn't essential by any means - you can put stuff wherever you like - but it might keep things neat for the purpose of these tutorials.
6. Just before we actually create the box, we should think about what texture we want to create it with. Textures can be easily changed at any time, but note that every solid MUST have a texture or the level will not work. So click LMB on the word 'Browse' in the **texture window** on the bottom right of screen. This will open a window of all available textures - pick one that might make a good room texture (i.e. not a liquid).
7. One more thing to check before creating the solid; make sure the 'solid type' selector on the Object Bar says 'Block'. If it doesn't, click once on the text and select Block. Once you're happy with the size, shape, position

Note: The reason the solid was chopped up was because Quake does not allow selective objects.

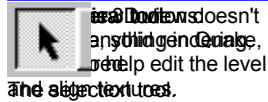
- and texture of your box, press Enter on your keyboard. This will create the solid.
8. This solid is actually our room, but the block is solid so we need to cut out an area inside it to make a room. Click on the **Selection Tool** and then in one of the windows click on/close to an edge of our newly created solid. If you've done this correctly, a bounding box will be drawn around our solid - looking much like the original guide-box - indicating that it is selected. With the solid selected, you can, if you wish, do some more resizing and repositioning at this point.
 9. Open the **Tool menu** at the top of the screen and from there select 'Make Hollow' (you could also hit RMB over the selected object). This will open a window asking how thick to make the walls; 32 is default which is fine, so hit OK. You'll now find that what was once one solid is now six - one for each wall 32 units thick (the default grid size is 64 units which is almost exactly the height of your character).
 10. There is only one more thing required before you can actually check out your room in Quake - a player start point. Click on the **Entity Tool** button. In the **Object Bar**, you should see "player_start_point" which is the default entity type to create. If it shows another entity type, click once on the text then find and select "player_start_point". Now in the top view, click LMB somewhere inside our room. Then go to a side/front view and make sure the green guide is inside the room from that view also - if it is not, hold the LMB on the green box and move it around until it's inside the room. You can place it near the floor if you want, although be sure not to place him IN the floor or you won't be able to move once you start Quake.
 11. Hit Enter to create the entity.
 12. Your level is now ready to play (although in it's current form interest is likely to be short lived). Hit 'F9' (this is Worldcraft's **Run Map** feature) and you will find the compilation window. Select 'Full QBSP' and 'Full VIS' but make sure you have LIGHT turned off. Then hit go.

QBSP will compile the level, then VIS will optimize the level (hardly necessary with a level this size, but it'll be fast with only 6 solids) and then load up Quake with your level in it.

If the compiling stage didn't work, either you messed up the level somehow, or maybe the executables - qbsp.exe and vis.exe - aren't located in your Worldcraft directory as they need to be. If you're stuck, check the **Frequently asked questions** section.

If it compiled with no errors (the errors would flash by pretty quick - if you want to examine them in more detail you'll need to manually compile the level: see **Overview of compiling** for details), but Quake doesn't load up, then check that the path to Quake that you put in the Tools|Options menu is correct.

If it all went well and loaded into Quake, it might just be worth checking if you started in the level at the position you think you placed the start point entity. If not, you probably have your top/front/side views mixed around. Experiment a little. Actually, experiment a lot.



1.5.1.2 Adding some features



1 Viewing the level in the 3D view

We haven't used the 3D window at all yet, so perhaps now is a good time to make use of it:

1. Select the **Camera Tool** button which is third down on the button-menu to the left.
2. Click and hold the LMB where the player entity is top view, then drag the mouse to the center of the pillar and let go.
3. You'll see a dark line in all three views now (you may need to scroll around a little in some windows) with a solid black dot at one end. You will also have noticed that the view in the 3D window has changed; the black dot represents the camera position and the other end of the line shows what the camera is looking at.
4. On the top horizontal button-menu in Worldcraft are the **3d view buttons** which determine how the 3D view is rendered. The first is wireframe (which you are seeing already), the second is solid and the third is textured. It'll be hard to resist pressing 'texture' now so you may as well do it.
5. With Windows 95 set to a 16-bit display, you'll now see the 3D view showing an accurate representation of how the level will look in Quake. Play with **Tools|Options|Texture Brightness** if the 3D view looks too dark/bright. With Windows 95 set to 256 colours, the 3D view may be corrupted (this is a DirectX issue and not the fault of Worldcraft).
6. With the 3D view operating, and with the Camera Tool still selected, you can click and hold the LMB on either end of the camera line and move it around to change the 3D view. There more immediate way to move the camera around is by the use of the mouse buttons inside the 3D window itself (along with the 'Shift' key for zooming in/out). See Camera Tool for details on its usage.



2 Selecting objects in the 3D view

Selecting objects has been covered thoroughly for 2D view, but once a map becomes cluttered the 3D view can actually be best for selecting. Try selecting some of the objects like:

1. Turn on **solid rendering** for the 3D window by pressing the gray block button from the MapView bar. The 3D view will now show solid colours.
2. Select the **Camera Tool**. Now move the camera view around until you are looking at the object you want to pick (see the section describing the **Maptool Bar** for full details on camera manipulation). Once you've played with the camera a bit, it'll become quite simple to use.
3. Now go to the **Selection Tool** and simply click on the object you wish to pick. It will turn red. Continue to hold the LMB if you wish to cycle through all objects currently under the cursor (very handy Worldcraft feature).





The selection tool's grid size is 64, because we made it 32 thick (when we made the selection box). The selection box won't snap exactly where you want it. You can either hold down Alt while you move (remove snapping altogether), or hit the 'T' key on the keyboard which will actually halve the grid size to 32 making the snap work as desired.

3 Adding more objects

As cool as our room is, it can be made a lot cooler. Adding in extra solids is as easy as it was to make the original room:

1. Select the **Block Tool** from the button-menu to the left.
2. From the **Object Bar** (just above the views area) you can now select another type. Choose a cylinder.
3. In the top view, draw out a 2x2 box, then from a side/front view, pull it's mid-top and mid-bottom squares up/down so that the bottom on the box sits on the ground, and the top touches the roof. Position it in the room, but don't place it where your player entity is.
4. Before making the solid, go into the **texture window** again (bottom right of screen) and select another good pillar texture (rock or something).
5. Create the solid by pressing Enter.
6. Press **F9** to check out the new map.

You'll now find a whopping great pillar in the middle of the room. This demonstrates how simple it is to add extra details to the map.

4 Adding some lights

At this point, when you run the level it will still be looking fully bright with no shading or shadows. Basically it looks like crap, so perhaps a few well placed lights will bring in some much needed atmosphere.



1. Select the **Entity Tool** from the button menu to the left of screen.
2. In the **Object Bar**, click once with the LMB then scroll down and select 'light_fluoro'.
3. Place the light entity around where you have your player start entity. Perhaps just a little above where the player one is (remember you'll need to manipulate it's position in more than one view to line it up correctly).
4. Press **F9**, but now in the compilation window select 'Full LIGHT'. Without this setting, any light entities in the map would still have no effect. Then press OK.

This will produce a very dark level except around the start position which will be subtly lit. It's easy to see that a lot of lights will be required to get nice even lighting around a large level. The alternative is to just boost up the amount of light produced by the one entity.

1. **Select** the light entity, then click the RMB with the mouse in the center of it's bounding box and select 'Properties' at the bottom.



The selection tool.

2. The **properties window** is where you alter entities. Turn off 'SmartEdit' and click on the 'light' key. This value is default at 200, so make it 600 (three times as bright).
3. Close the properties window and hit **F9** to run the level.

1.5.1.3 Enhancing the room

1 Modifying the textures

Now lets see an interesting effect of certain textures:

1. Choose the **Selection Tool** then select the pillar solid by clicking LMB on/close to one of its lines. If you accidentally highlight the room solid, then just click somewhere there are no solids to deselect everything (or press Alt+C).
2. With the pillar solid highlighted, click LMB on the **texture window** again. Now select the water01 texture or similar.
3. Changing the current texture to water hasn't automatically changed the texture of our selected object. Press the **Paintbrush tool** to apply the texture to our object.
4. Press **F9** to run the level.

The pillar now has the animating water texture wrapped around it which looks kind of odd free-standing in thin air. In Quake, try running into the pillar to see what happens.

2 Making objects overlap

So far we've been basically aligning objects neatly by the grid, but Quake's style of level editing means that this isn't necessary. Just as an example:

1. Deselect any objects that might be selected, then **select** the pillar solid.
2. In the top view, move the pillar object so that it's center right on top of a wall. This will mean part of the pillar is inside the room, and part is outside.
3. Press **F9** to run the level.

If the pillar is still water, you'll be able to see from this test that Quake has nicely clipped away that part of the pillar you don't see.

3 Rotating and shearing

Now let's apply **rotation and shearing** to some of the solids:

1. With the pillar still selected (or select it again if it isn't already selected), click LMB once inside it's bounding box.
2. The bounding box should now have solid circles at each corner which indicates that it's now in 'rotate' mode. Press and hold the LMB on one of the black dots and move the mouse around.
3. The pillar will be rotating around. Release the LMB once you've seen how it works.
4. Click once with the LMB again in the center of the bounding box.

Note: This operation will produce 6 triangles and 6 bounding boxes. Selecting one of the triangles will allow you to place it with Worldcraft several places and glue the others together, preferable.

5. The bounding box will have changed again to show solid black squares halfway long each edge (with none at the corners) which shows that it is in 'shear' mode. Press and hold the LMB on a square and move the mouse around.
6. This will shear the object. Release the mouse button when you've seen how it works.
7. Click once inside the bounding box again to return to move/ resize mode.



That demonstrates one way to produce solid shapes that aren't part of the predefined set. The other way is to use 'carve' to use one solid to slice away part of another.

4Carving a volume out of solids

1. **Create** a new solid. We're after a 'block' type solid, located so that it half sits inside the pillar, and half sits outside (in the room). Press Enter when it's in the desired position (texture isn't important).
2. Now **select** the room object.
3. Select **Tools|Ungroup** from the menu (or from the button-menu, or even the floating menu you get if you click the RMB over the selected object).

The **Carve Tool** carves any object that it is touching based on whether their bounding boxes are touching. Since the room group is surrounding everything, its bounding box is also touching everything else meaning that if we did a carve with those 6 solids grouped, the walls would become fragmented (try it if you want to see the effect). By ungrouping them they each get their own, smaller, bounding box which gets around the problem. You could just as easily move the room group well out of the way to do the carve, then move the room back once carving was complete. See **Tools|Carve** for further information.

4. Now deselect anything currently selected, and **select** that new block you created that was sticking into the pillar.
5. Select the **Carve tool** to carve a block out of the pillar in the shape of the cube.
6. With the cube still selected, use **Edit|Delete** to remove the block from the map (it was only used for carving, which is why texture wasn't important).
7. Press **F9** to run the map.

If you walk over to the pillar (wherever it is in your map at this point) you should see the shape of the block cut neatly into the side of the pillar.

1.5.1.4Joining two rooms

1Creating a hallway by carving

For this tutorial, we need two rooms that are not yet joined. Making two rooms is simply a case of following the **first tutorial** twice. However, do

Note: Keep in mind that the room walls can be any thickness - it makes no difference in the compiled level. The only important rule is that the level must have no gaps (or "leaks") to the outside void when it comes time to run VIS.



NOT create two player start point entities. With two hollow rooms created, connecting them is achieved by cutting out doors from each room, and either lining them up so that the two doors touch each other, or creating a hollow connection between them.

We'll assume with this tutorial that the rooms are a few grid lines away from each other, and at the same height. Due to the openness of this particular task, the tutorial may be hard to follow.

1. Select the **Block Tool** again, and create a 'block' type solid connecting the two rooms to each other. The solid should be placed where you want the doors to connect, and it should meet the outer wall of each room. Overlapping it into the walls of each room is fine; however, it should not extend into the interior of either room.
2. Once the solid is created (with an appropriate **texture**) it can still be **manipulated** into position via the various move, resize, rotate and shear functions covered in the previous tutorial.
3. Now create another solid - the texture of this one is unimportant - and place it inside the solid you just created. It must be visibly thinner than the first solid in height and width; however, it should be longer than the first solid, extending into the interior of each room at the ends. This latest solid will be used to cut a hole through the wall of room 1, through the larger solid placed between both walls, and through the wall of room 2.
4. Because carving is based on bounding boxes, we should ungroup the solids making up each room if they are grouped. To check, select one of the solids making up the walls of a room, and choose Tools|Ungroup. Do this for each room.
5. Select the thinner solid that we created last.
6. Select the **Tools|Carve** menu item (or choose the **carve button**).
7. Delete the thin solid (press the Del key, or choose **Edit|Delete**).



If the second solid was thinner and longer than the first solid, then you should find that an open walkway now exists between each room. As long as the interior of the hallway remains completely sealed from the outside void, the level will work.

If both rooms actually share a common wall, then one simple **Carve** operation is all that is required. If the rooms aren't neatly aligned with each other, it might be simpler to first carve door holes in each room, then construct the corridor by hand - first a floor, then each wall, then a roof - with several solids and the 'shear' operation.

Note: You can use the Search
VisGroups, you can also save
groups to the file. If you
if you only have one
VisGroup, then
information boxes
is displayed in the File
menu for details on file
formats.

1.5.1.5 VisGroups

1 Classifying objects into VisGroups

VisGroups are a unique feature of Worldcraft that makes editing large cluttered levels much easier, so they are worthy of our attention in this tutorial. When you start to work on large levels, VisGroups will be your best friend.

The idea is that you create a whole bunch of “VisGroups” which relate to particular areas you might want to selectively turn on/off all in one go. An example VisGroup list for a simple two room level might be:

```
room1
room2
doorstuff
furniture
furnitureroom1
furnitureroom2
```

Note that some of the groups listed might seem to overlap. This is one of the useful functions of VisGroups: any item can be in any number of groups.

2 Setting up VisGroups

Let's setup the above VisGroups ready to be used (this is just a demo of setting up VisGroups, but if you happen to have such a room ready it can be applied easily):

1. On the right side of the Worldcraft window at the top is the **Filter Bar**. At top/right of it is a button labelled ‘Edit groups’ which will invoke the VisGroups editor. Press it.
2. Now select ‘New Group’ - a new group is created.
3. With that group selected, you can edit the name of the group by typing in the text box top/right. Type in ‘room1’.
4. Follow steps B and C for all the VisGroups listed above.
5. With the VisGroups created, you are free to begin assigning objects to them. Select an entity, solid or group then open it's properties window via the Edit->Properties menu (or by clicking the RMB over the selected objects and selecting the bottom tool).
6. Go to the ‘VisGroups’ folder which for an entity will be the 3rd folder, and for solids it will be the only folder.
7. Here you choose which groups you want this object to be associated with. These groups have nothing to do with Quake, they are simply to make editing easier. If the chosen object is a chair in the first room, for example, you'd probably place a ‘tick’ next to; room1 furniture furnitureroom1
8. Place each item of the two rooms into appropriate groups.

Our VisGroups are now ready for use. As you create new objects you should add them to the appropriate VisGroups or create new VisGroups where needed.

Note: You don't need to make everything visible again to run it in Quake. If you hit **F9** and examine the compilation options, you'll find a toggle which tells it whether to compile invisible objects or not.

3Enabling VisGroup filtering

In the **Filter Control window**, you will see a list of all the VisGroups along with tickboxes indicating which groups are visible. By default, Worldcraft doesn't have VisGrouping enabled, so no matter what groups are ticked you will not see any difference.

1. Tick some VisGroups you wish to work on. Perhaps you are planning to add something new to room 1, so having anything from room2 in the editing windows would just slow things down and make it hard to see whats going on. For now, tick just the 'room1' group which should at this point have anything associated with room1 placed in this group.
2. Place a tick in the 'Use groups' box located just below the list of groups. This will tell WC that you wish to use VisGroups although you won't see any changes yet.
3. Click 'Apply'.

After clicking on Apply, Worldcraft will look at all the settings in the VisGroups menu and then apply them to the editing window. This means that any objects what haven't been set to belong in the 'room1' VisGroup will be turned invisible making editing of room1 a lot easier.

4Changing which VisGroups are visible

After editing the new object for a while you might decide that having all the other furniture in the room is making it still too cluttered, so:

1. In the **Filter Control window** (over the right of view), turn off 'room1' and turn on 'roomsurrounds1' with maybe 'doorstuff' if it'll be helpful for editing the new object.
2. The 'Use VisGroups' toggle should still be active from before, but if not turn it back on.
3. Hit Apply.

The furniture will disappear from view.

Finally, when the new object is completed and you want to view the whole level again:

1. On the **Filter Control window**, deselect the 'Use VisGroups' option.
2. Hit 'Apply' to make the change happen.

1.5.1.6A moving door

1Creating a door

The func_door entity is one of the more versatile functions available for level editing. The following example shows how to use it to create a basic door that opens up when you walk close enough to it.

1. Load in or create the simple room as described in **tutorial number one**.

Note: This is a variable for a function that opens a door by a certain length, see positive or negative values.

2. Now **create a block type solid** positioned in the room where you want the door (for test purposes it would probably be easiest to just place it in the center of the room) and at the size you want. Don't worry about fancy door trimmings just yet, this example will be free standing.
3. **Select** your door solid, and then select the 'toEntity' tool on the **Object bar** (or use 'Tie to entity' from the menu or the RMB-menu).
4. The door solid is now attached to an entity and it's the entity we wish to manipulate to become a door. After selecting 'toEntity' the **properties window** would have come up automatically. Change the entity type to 'func_door'.
5. In the 'angle' settings top/right of the properties window, make it -1.
6. Close the properties window and press **F9** to run the map.

Running the map, you will find the door sitting where you created it, and the door will open vertically upward if you walk close to it.

2 Adjusting the door's movement

By default, a func_door (or func_plat etc.) will open a distance of exactly its width in the direction of its angle. To make it open twice as far:

1. You'll need to remember how tall you made your door. Lets assume you made it 80 units high.
2. With the door solid/entity **selected**, click the RMB inside the bounding box (which will open a menu) and select the bottom option which is 'properties'.
3. One of the **keys** is called 'Lip' which does different offset related things depending on the entity type. With a func_door it can be used to alter the amount the door will move; a positive value will make the door open less than it's full length, and a negative value will make it open more. Put in a value -2 x the height of the door (in our example case, that will be -160).
4. Press **F9** to run the map.

And there is a door that opens twice its actual height.

3 Connecting the door to a trigger

As with almost all entity type in Quake, func_door can do much more than simply opening when you walk close enough. Triggering the door to open by a switch is a common adaptation that Quake levels use a lot:

1. **Select** the door solid/entity and open its properties window.
2. Turn off 'SmartEdit' and click on 'add' to add a new key.
3. Make the key name 'targetname' and give it a useful value such as door1.

Note: 'targetname' is one of the most used **key types**. By giving an entity (almost any entity) a targetname key, you are making that entity activated by some other trigger (and a trigger can be almost anything). If you ran the map now, you'd find the door wouldn't open, since the door now requires a trigger of some sort.

Note: Remember that the entity is NOT a solid that can be triggered via the trigger_entity type. It is a trigger which is visible or invisible. The trigger_entity type is a point in space that is a copy of the entity. The entity is a solid that is located just below the position as the trigger. Solids make it appear visible.

4. **Create a new solid** making it about 16 units cubed. This will be our trigger so place it somewhere obvious and at a height such that the player can run into it. Use an appropriate texture that identifies it as a trigger (i.e. lava wouldn't be that useful).
5. **Select** the new solid and press **toEntity** and make the entity type 'func_button'.

Note: There are numerous trigger types, and the func_button one we used is the only one that keeps the solid visible - that is, all the trigger_entity types will use the solid shape only as a detection box to tell if the trigger has been set off, and the solid itself will be invisible (which is why func_button isn't part of the trigger_group).

1.5.1.7A teleporter

1 Creating a teleporter

A teleporter consists of two entities; one has a solid attached to it which is the part you step into, and the other is a standalone entity that marks where you teleport to.

1. Start by **creating a tall solid** - about big enough for a player to walk into.
2. Select this solid and from the menu select **Edit|Copy** (or from either the button or RMB menus).
3. Select **Edit|Paste**, then **select** the newly pasted solid and move it well out of the way (for now).
4. **Select** the original teleport solid (make sure it's at a good position in your room) and press **toEntity**, then make the entity type a 'trigger_teleport'.
5. Turn off SmartEdit and add/edit a key called 'target' and give it a value something like "tele1".
6. Now **create an entity** and place it where you want the teleporter to throw you. The entity type should be an "info_teleport_destination".
7. **Select** the newly created entity and open its **properties window**.
8. If the entity type isn't already info_teleport_destination, make it so right now.
9. Change the angle parameter (top right of the properties window) to the direction you want the player to be facing when they pop out.
10. Turn off SmartEdit, then add/edit a key called 'targetname' and makes its value equal to the 'target' value you gave the trigger_teleport (tele1).
11. Both entities are now set up correctly and will work. But the teleporter will be invisible, so **select** the copied solid you made back at step 2.

Note: path_corner entities differ relative to where you place them and since certain of them are also used in the paintbrush console box. The reason is that the first stop point to line everything up perfectly apart from careful placement.

12. Open the **texture Browse window** and find that wobbly black-with-white-dots one that we all recognise to be the teleport texture.
13. Hit the **Paintbrush tool** to apply this texture to the selected solid.
14. Move the solid so that it sits at exactly the same position as the teleporter solid entity.
15. Press **F9** to run the map.

You'll see that the teleporter has its characteristic wobbly texture, and by running into it you'll be transported to the location given by the destination entity, and oriented in the direction given by the angle parameter.

1.5.1.8A moving platform

1 Creating a train

Trains function a lot like teleporters in that they have a solid attached to an entity as the main section (the train itself) and 2 or more standalone entities to give a path to the train. This example will create basic train moving between three stop points.

1. **Create a thin solid** to use as the train itself (the train can be more complex by using multiple solids and then grouping them together into one object). Give it an appropriate **texture**.
2. **Select** the solid and press **toEntity**. Make the entity type a 'func_train'.
3. Turn off SmartEdit and add/edit a key called 'target' and give it a value of something like "stop1".
4. The train is complete, but now we must tell it where to go. Create a standalone entity (use the **Entity Tool**) and place it where you want the train to start. The entity should be created as a 'path_corner' type although it can be changed in the **properties window** at any time.
5. With the first stop created, **select** it and open its **properties window**.
6. Create/edit a key called 'targetname' and give it a value equal to the target value you gave the train itself (stop1).
7. Create/edit a key called 'target' and give it a value of 'stop2'.
8. Close the properties window and now create another entity of the path_corner variety. Place it where you want the train's second stop to be (keeping in mind the bottom/left rule).
9. Select the entity then open the properties window. Edit/create a key called 'targetname' and give it a value of 'stop2'.
10. Edit/create a key called 'target' and give it a value of 'stop3' (are you beginning to see a pattern?).

11. Close the properties window and now create yet another entity of the path_corner variety. Place it where you want the train's last stop to be (keeping in mind the bottom/left rule).
12. Select the entity then open the properties window. Edit/create a key called 'targetname' and give it a value of 'stop3'.
13. Edit/create a key called 'target' and this time give it a value of 'stop1' (ie. this will make a train loop back to it's start position again).
14. Exit the properties window and press **F9** to run the map.

When you enter the level, the train will be sitting waiting at the first stop point. Run up to it and jump on. A sound will play, followed by a pause (alter the 'delay' key value to change the pause length) and then it will travel to the second stop point. At the second stop point it will pause briefly again (length is set by the 'wait' key, although it seems to be impossible to remove the "stop" altogether) and then move on to the next stop.

And then from the third stop point it will move back to the first stop point again. Once it's reached where it started, by default it will just do the circuit again, but if you give any of the path_corner entities a 'wait' key of value '-1' the train will remain at that corner.

Trains aren't as flexible as they first seem; the requirement that they **MUST** stop - no matter how briefly - at each path_corner means that continuous motion isn't possible, and you cannot give rotation of any type to your train. This makes things like Duke Nukem's trains and swinging doors impossible to achieve without getting into QuakeC. (Hexen 2 and Quake 2 are expected to address this shortcoming.)

1.5.1.9 Frequently Asked Questions

1 How can I run maps that I've made in Quake?

A: Worldcraft includes a feature that will build your map into a playable BSP file, and run Quake. You can activate this feature with the **F9** key, or use File|Run Map in Quake. This feature might not work on all computers, though, since QBSP (the included program that turns MAP files into BSP files) requires a lot of memory. If you find that the feature does not work, here's how you can do it yourself:

1. Choose File|Export to Quake MAP. This will save your level as a Quake MAP file instead of a Worldcraft RMF (Rich map format) file.
2. Exit Worldcraft, and bring up an MS-DOS Window (use the Windows 95 Start menu, choose Programs, then MS-DOS Prompt.)
3. Change into your Worldcraft directory (for example, type cd\worldcraft.)
4. Run RUNMAP.EXE with your map file. RUNMAP takes two parameters: one, the name of your map file, and two, your quake directory. For example, if you were working with a file named mymap and your copy of Quake was located in D:\Quake, you'd type:

```
c:\worldcraft> runmap mymap d:\quake
```

2 What's the scale of maps in the editor to how they're displayed in Quake?

A: You get a feel for this over time, as you've edited some maps and tried them out in the game. A measurement that might help in the meantime is that the size of the player is about 25 units wide and 60 units high.

3 When I start up Worldcraft, I don't see anything in the 3D views. What's wrong?

A: Make sure that you've loaded a MAP file or created something to be displayed. Remember, solids aren't created until you press the ENTER key after drawing them!

If that's not the problem, try turning off Hardware Acceleration for your 3D card. Even if you don't have a 3D card, try this option - go to the **Tools** menu, choose **Options**, select the **3D views** tab, and uncheck "hardware acceleration." You will need to restart Worldcraft for the changes to take effect.

Note: Matrox Millennium cards have real problems with Microsoft's Direct3D, which is what Worldcraft uses to display your maps. If you have a Matrox card, try to get new drivers from Matrox or Microsoft - these drivers weren't available at the time this was written, but might be available by now.

1.6Menus

Menus in Worldcraft stick to the standard style laid down by Windows 95. To open a menu, you can either click with the left mouse button (LMB) on the required menu title (at the top of the Worldcraft window), or use the Alt+{underlined letter} hotkeys.

By clicking once with the LMB on a menu title, it's selection menu will open and stay open until you do another LMB click (whether it be inside the menu or anywhere else). This method allows you to browse through each menu by moving the mouse around, and to select an item by clicking on it with the LMB. The other method is to click the LMB on the desired menu title, but continue to hold the button. An item is selected by positioning the mouse over it and releasing the button.

For further information on menu selection and menus in general, refer to Windows 95 documentation.

1.6.1.1File Menu

1New (Ctrl+N)

Opens a blank map. Used when you wish to create a new level from scratch. This will not destroy any level currently loaded in Worldcraft, it will just bring up another set of windows. This ability to open multiple levels allows cutting and pasting between maps. See the MENU|WINDOW:' for further information on using multiple maps.

2Open (Ctrl+O)

Presents the standard Windows 95 Open File dialog box which allows you to select an existing quake map to load in. As with New, this will not destroy any currently loaded map.

The selector at the bottom of the file dialog box allows you to filter files between .MAP and .RMF. The .MAP file is the standard Quake editing format and is used by the QBSP program to create the actual Quake .bsp file (a .bsp file can be loaded into Quake and played). A .MAP file is compatible with all other Quake editing tools available.

The .RMF format is Worldcraft's own special format ,which allows extra information to be saved which is not part of the .map standard (such as 'VisGroup' information). It's recommended to use the .RMF format for general editing (since you can easily save them out in .map format if you wish) although if you don't intend to use any of Worldcraft's special features there is no harm in saving as .MAP.

Note that if you use Worldcraft's 'Run Map' feature, it will automatically save a .MAP file.

3Close

This will close the currently selected level that you are editing (if there are multiple maps loaded, the current one is the one which has a blue top border). This function is the same as pressing the LMB on the 'X' icon on the top right

of the editing area. If any changes were made to the current level since the last save it will ask you if you wish to save before closing.

4Save (Ctrl+S)

If the current map already has a name, this will simply save the level with that name. If it is a new map which hasn't yet been saved, it will prompt you with a file dialog box to give the project a name. See Open for details on .map and .rmf formats.

5Save As

This command will present a file dialog box even if the current map already has a name. Handy if you wish to save the map but don't wish to save over the top of the previous copy.

6Export To Quake MAP

This command will save your level in an editor-neutral MAP format. This is the format used by QBSP, and is suitable for converting your level. This format is also created by the Run Now command.

7Run Map in Quake (F9)

This does many things, but the end result is that by pressing this button you can play your level in Quake with no further fuss. Many factors determine whether it will be successful; these are discussed in the description of **Run Maps dialog box**. When you run this feature, Worldcraft will automatically save your script.

8Exit

This will exit Worldcraft. Any loaded levels which have been modified since their last save will give a request to save. Exit is the same as pressing LMB on the 'X' at the very top right of the Worldcraft window.

1.6.1.2Edit Menu

The Edit menu presents the usual commands for editing, along with some Worldcraft specific commands.

1Undo (Ctrl+Z)

Will undo the previous command. Worldcraft will remember consecutive commands and allow you to undo several of them at once; the number of undo steps is limited by the setting in the Options menu.

2Redo (Ctrl+Y)

Will redo the command that was undone using the Undo command.

3Cut (Ctrl+X)

This will delete the currently selected solid(s)/entity(s) and keep a copy in the clipboard buffer.

4Copy (Ctrl+C)

This will copy the currently selected solid(s)/entity(s) into the clipboard buffer.

5Paste (Ctrl+V)

This will paste whatever is in the clipboard buffer to the current map. The pasted object is centered in the two previously selected windows.

6Paste Special...

See the section **Paste Special** for details.

7Delete (Del)

This will remove the selected solid(s)/entity(s) from the map, and will not keep a copy in the clipboard buffer. A deleted item is lost forever.

8Clear selection (Alt+C)

Will clear all selections.

9Select all

Will select everything in the level being edited.

10Properties

This command is where all the serious level creating and editing stuff happens. See the **Properties window** for full details.

1.6.1.3Map Menu

The map menu contains various options and menus relating to the map.

1Snap to grid (P)

Moving objects around will snap them to the grid. This is on by default, and allows objects to be lined up more easily.

The grid snapping can be overridden at any time by holding down 'Alt' while moving. Doing this will ignore the current Worldcraft grid settings and instead snap to the smallest units that Quake can recognize.

Where the object chooses to snap to on the grid is determined by where you clicked on the selected object to move it around. If you clicked on it near the upper lefthand corner, then the object will snap to the inner top/left corners of the grid. Likewise, selecting the object towards the bottom/right will snap to the grids inner bottom/right corner.

2Show grid (G)

The grid is by default turned on. Here you can turn it off if you don't wish to snap to the grid and it's cluttering the view.

3Grid settings ([and])

These two sub-options will make the grid smaller or larger by a factor of two. Note that there are two sets of hotkeys available for grid setting.

4Map properties

This contains information about the map which Quake will use. This is related to object properties and is also covered in the **Properties window** section.

5Show Information

Shows basic statistics on your level. Solids are simple solids, PointEntities are lone entities, SolidEntities are entities which have one or more solids attached to them, and Faces is the number of polygons making up the scene.

To learn about solids and entities, see the Overview of Levels.

6Check for problems

This is a useful command which will analyze your map for common errors. A full list of the problems that Worldcraft can detect will be added to this documentation in a later release.

1.6.1.4View Menu

The view menu contains all Worldcraft viewing related commands.

This allows you to determine what elements are present on the Worldcraft layout. Details and descriptions of the particular elements can be found in the **overview of toolbars**. The following can be selectively turned on and off (descriptions relative to default Worldcraft layout):

Object Bar: Thin horizontal bar located second row from the top.

MapTools Bar: Thin vertical bar located on the left of the layout.

MapOperations Bar: Thin horizontal bar located on the top row to the right.

MapView Bar: Thin horizontal bar located on the top row to the left.

Filter Control: Large box window on the right of layout to the top.

Texture Bar: Large box window on the right of layout to the bottom.

Status Bar: Thin horizontal bar located at the bottom of the layout.

12D X/Y Y/Z X/Z

The tick indicates which view the currently selected editing window is showing. Each editing window can be changed to another view at any point (so it's possible to have multiple windows looking at the level from the same direction) by changing the selected item.

23D

There are three render modes for the 3D view window:

wireframe: Views all solids in wireframe mode, and views entities as boxes. The size and colour of entities are determined by the type of the entity. Solids are shown in blue unless they belong to a VisGroup which have their own colours. The currently select items are shown in red.

filled polygons: Solids are viewed in solid form with entities rendered as solid boxes. Colours are determined by the same rules as with the 'wireframe' render method. Some basic lighting is applied to give the objects some shape.

textured polygons: This mode uses Microsoft's 'DirectX' to render the level with texturemaps applied. This view presents basically a 1:1 view of how the level will look inside Quake (with the exception of lighting) and is very useful for aligning textures.

Note: There is a limitation with DirectX while using 256 colours. The symptom is corruption of either the 3D view, the texture menu, or both. To avoid this if it causes problems, set your system to at least a 16bit display via Windows 95: Start|Settings|Control Panel|Display (and then Settings).

3Autosize 4 views (Ctrl+A)

Will resize all four windows to equal sizes. Further window sizing details can be found in the WINDOWS topic.

4Maximize/Restore active view (Z)

This command will be implemented in a future release of Worldcraft.

5Hide Items

Selecting this will hide all entities from the view. By default this is off.

6Center views on selection (Ctrl+E)

Center the X/Y/Z window views around the currently selected items.

7Hide selected objects

This command will make the selected objects invisible. This visual control is not related to the VisGroups feature, and should not be confused with VisGroup filtering.

8Show hidden objects

This command will make the all hidden objects visible.

1.6.1.5Tools Menu

The tools menu contains raw editing and manipulation commands.

1Carve (Shift+Ctrl+C)

This command is very powerful and allows more complex shapes than the few default ones Worldcraft contains. The operation is a mathematical 'Boolean subtraction' which means that it subtracts one object from another.

To use Carve, place to solids in an arrangement such that they overlap each other. Then select the solid you wish to cut with - the knife - and select the Carve option. The knife will remain unaltered, but the other solid(s) will have the shape of the knife solid removed from them.

Due to the way Quake levels are created, the solid that was carved up will most likely be now a combination of several smaller solids, in which case they will be grouped together.

Any solid who's bounding box is even partially in the 'knives' bounding box will be carved. This means that if you have a big environment around the solids you are carving, and all the solids making up the environment are grouped, the environment will get carved too. The solution is to do all carving outside of the main level area where it can't do any harm, or you can use a blank map for working in from which you can copy the results back to your level map.

For step-by-step instructions on how to carve an object, see **Carving volume out of solids**.

2Make Hollow (Ctrl+H)

This will hollow out the selected solid. The solid will be divided into a group of several smaller objects that make up the same shape but enclose a cavity. The thickness of the walls will be equal to the number of units that you specify.

If you specify a negative number for the wall thickness, then the walls will be constructed outward from the outer face of the solid, rather than inward. This allows you to build a solid (or group of solids) in the exact shape of the desired interior cavity, then construct walls around it using the Make Hollow command.

3Group (Ctrl+G)

Group will join together all selected solids and entities into one unit. Selecting and deselecting any item contained in a group will select and deselect the WHOLE group. The only method to select an individual item contained within a group is to select the group and then select 'ungroup' (see below).

There are two useful purposes for grouping objects. Firstly, because it can make moving elements around much easier if it's to be done often (instead of having to select each element time and again). The other use is to easily create complex doors, platforms and trains that move as one unit.

Grouping will destroy any entity settings that a solid might contain.

For details on grouping, see **Grouping and filtering objects**.

4Ungroup

This will split a grouped unit into it's individual solids and entities. Ungrouping will destroy any entity settings that the group may have had.

5Tie to entity (Ctrl+E)

This will tie a particular solid, or group of solids to an entity. This is used to make everything from doors to trains and triggers. See the index for descriptions of solids and entities. Details on the entity window can be found in the section describing the **Properties window**.

6Move to world (Ctrl+W)

This is the reverse of 'Tie to entity'. It removes any entity settings the selected solid(s) have and reverts it/them to a normal solid.

7Texture application (Shift+A)

Allows pixel-accurate placement of textures on any selected faces. Faces are the individual sides/facets of a solid and can be alter individually or as groups. If the current view mode for the 3D window is 'textured polygons' the changes made in the face will happen dynamically in the 3D view allowing textures to be lined up very accurately with little effort.

For details about using the texture application mode, see **Texture application window**.

8Align Objects

If several objects are selected, the align options will align all objects to the object that is the furthest in that direction.

As an example, we have 4 objects we can see from the front view but they are all at different heights. If we select all 4 objects and then choose Tools|Align Objects|To Top, the three lowest objects will move up to align with the highest one.

9Prefab Factory

This command allows you to access the Prefab Factory. The Prefab Factory is a set of libraries that contain useful objects that make building your level easier. You can also define your own libraries that contain objects you have created yourself. Each object must reside in its own .MAP or .RMF file. Libraries in the Prefab Factory also show up on the **Object bar**.

10Create Prefab

This command will create a new prefab using the selected object.

1.6.1.6 Window Menu

The window menu contains features relating to Worldcraft's layout.

1 New Window

Will create a new set of 4 windows for the current map and place them in front of the existing windows. Any change done in either set of windows will instantly update in the other set.

To change between sets, or to see behind the current working windows use the left-two icons on the top right of the work area (i.e. the second set down on the top right of the Worldcraft screen). The '_' button will minimize the current window and the icons that appears as two overlapping squares will shrink the current view.

2 Cascade

This will shrink all non-minimized windows and display them neatly cascaded in the work area starting at top left.

3 Tile

Will arrange all non-minimized windows so that they all have an equal amount of room on the work area, but do not overlap.

4 Arrange Icons

Will arrange all minimized window icons neatly along the bottom of the work area.

5 Messages

This command opens the Messages window, in which Worldcraft will report errors and other troubleshooting messages. Most users will have no need to use the Message window.

The bottom part of the Window menu contains the name of each map currently loaded. A tick is placed next to the currently active map, and by selecting any of the maps listed that map is made current. If the selected map is minimized it will be expanded.

1.7 Toolbars

Many of the often used commands found in the menus are also available as buttons in onscreen menus. These button-menus are divided into several sub-groups which can be individually moved around the Worldcraft screen, or even removed altogether.

To turn on/off any selected button-menu, use the toggle switches in the View|Screen Elements menu.

Moving button-menus around is achieved by clicking and holding the LMB inside the border of a menu (but not on an actual button), and then dragging it around the screen. Worldcraft will detect where a menu is currently being moved to, and change it's orientation accordingly.

Upon releasing the LMB, the menu will be dropped into position and Worldcraft will automatically rearrange all other windows and button-menus to accommodate the newly placed menu.

If dropped in a border section, the button-menu will become part of the border elements, but if dropped into the work area, it will become a standalone menu.

Note that any changes made in window/button-menu arrangement will be permanent and will be saved as default.

The following refer to default positioning.

1.7.1.1 MapView Bar

Contains three sets of buttons:

1XY, YZ, XZ View

These refer to the three possible planar views. If a 2D view is currently selected, the button corresponding to that particular view will be highlighted.

You can make the currently selected window into a different view by selecting another button. Even the default 3D view located top left can be turned into a planar view, and you can have multiple showing the same planar view.

By default, the top right window is XY, the one below it is XZ and the bottom left window is YZ, with the top left window being the 3D view.

2Wireframe, Polygonal, Textured View

These three buttons refer to the render method of the 3D view. See the **View Menu** topic for further descriptions of each render method.

3Toggle Grid

Turns the grid on and off. See **Map|Show Grid** for further details.

1.7.1.2 MapOperations Bar

Contains five groups of icons relating to map editing and manipulation.

1Carve

See **Tools|Carve** for details.

2Group, Ungroup

See Tools|(Un)Group for details.

3Align Left Right Top Bottom

See Tools|Align Objects for details.

4Cut Copy Paste

See the **Edit menu** for details.

5Run Map! (F9)

See the section on the **Run Map dialog** for details.

1.7.1.3Object Bar

Contains four sections relating to object status and creation.

1toWorld

Will remove any entity settings of the selected solid(s)/group. See Tools|Move To World for further details.

2toEntity

This will attach the currently selected solid(s)/group to an entity. Entities are covered in the Properties topic. See Tools|Tie To Entity for further details.

3Categories

When the **Entity Tool** is active, no categories are available. With **Block Tool** selected, there will be at least one category available: the "Primitives", which contain Worldcraft's four basic solid shapes: block, cylinder, spike, and wedge. If any Prefab libraries have been defined, then the library names will also be listed here.

4Object

This list contains all objects that belong to the chosen category. For entities, the list contains all entity classes available. For solids, if the 'Primitives' category has been selected, then this list contains the four types of primitives. If a Prefab library has been selected, then this list contains all objects in the library.

5Faces

This selector becomes active if the **Block Tool** is active, and either 'cylinder' or 'spike' is the current solid type. The value, which can be changed by either keyboard or clicking LMB on the little arrows), represents the number of side faces the cylinder/spike will be created with.

1.7.1.4MapTools Bar

The main map creation button-menu contains 4 sections; viewing and selection, creation, textures and layers.

1Selection tool (Shift+S)

Used to select object(s) thus making them current. Selecting objects can be done from both 2D and 3D windows by clicking the LMB on the required objects. Clicking on or near the lines of objects is the most sure way of selecting the correct object.

Selecting multiple objects can be achieved by holding down 'Ctrl' when selecting additional objects.

If selecting in a shaded or textured 3D view, by holding down the LMB in a selected area will cycle through all objects contained under the pointer.

A selected object is rendered as red in the 3D view, and is bordered by a red bounding box in the 2D windows. If several objects are selected, the bounding box expands to cover all objects.

To deselect all objects, click the LMB in an area of the 2D windows containing no objects, or use the Edit|Clear Selection tool (Alt+C).

For details on the use of selecting objects, see **Selecting and adjusting objects**.

2Magnify (Shift+G)

Used to zoom into and out of a 2D view. Clicking the LMB in a 2D view will zoom into the area under the pointer, and clicking RMB will zoom away from the area under the pointer. The same functions can be achieved by the '+' and '-' keys on the keypad.

3Camera (Shift+C)

The camera tool is used to alter the 3D view and can be used in one of two ways.

If activated for the first time, you can draw a line in any of the 2D views; the start of the line, which is drawn as a solid black dot, will become the camera position, and the end of the line will be the place the camera looks at. Length of the line is not important. Once this line has been drawn once, it can be altered at any time by (with the camera tool selected) clicking LMB on either end of the line and dragging them around in the 2D views.

The second method is a little more intuitive as you are working directly in the 3D view. With the camera tool selected, you can use the LMB and the RMB with mouse movement to directly affect the view of the 3D window. Speed of update depends on map size and the current render method for that window.

The position and orientation of the camera is used only for editing and does not relate to anything in Quake. The default position for the camera is high above the level looking directly down at position 0,0,0.

For details on the use of the camera view, see **Navigating maps in three dimensions**.

4Block Tool (Shift+B)

Used to create solids in the map. Once selected, the 'Solid type' selector (second row from top) becomes active and allows you to specify the type of solid to be created.

See **Selecting and adjusting objects** for details on manipulating created solids.

See **Overview of Solids** for further information on solids.

5Entity Tool (Shift+E)

Used to create standalone entities for the map (i.e. entities that have no solids attached to them). Once selected, the 'Entity type' selector (second row from top) becomes active and allows you to specify the type of entity to be created.

See the **Properties** topic for details on manipulating created entities.

See the **Overview of entities** for further information on entities.

6Toggle texture application (Shift+A)

See **Tools|Texture Application** for details on the use of this command.

7Apply current texture

Will apply the current texture to all selected solids. This tool works on the whole solid, so for individual faces the 'Toggle texture application' tool should be used.

1.7.1.5Filter Control (VisGroups)

VisGroups is a powerful tool for limiting visible objects in the map. This allows for much easier editing on large levels but does require that you use the .rmf file type for saving your maps since the .map format doesn't include the VisGroup information.

The 'Filter Control' button-menu allows the filtering and creation of VisGroups. For this menu to be used, objects must be set up with VisGroup settings which is covered in the section describing the **Properties dialog box**.

1VisGroups

This large box at the top/left contains a list of all created VisGroups along with their on/off status (identified by a tick). You can individually turn on/off each group, although the changes don't take effect until you use 'Apply' (see below).

2Filter by

With this tick box unticked, VisGroup filtering is totally turned off. With it ticked on, filtering is activated according to the settings in the VisGroups box. The change of this toggle only takes effect when you hit 'Apply' (see below).

3Apply

This will re-filter everything according to the current settings of the VisGroups settings and the 'Filter by' toggle.

4Edit Groups

Takes you into the VisGroup editor. This editor can also be accessed via the VisGroup folder in the properties window.

5Groups

The large area to the left lists all VisGroups for the current map, along with their status (status is defined by a tick - on means the group will be visible if filtering is activated). There are no VisGroups by default, so to use them you must create some.

6Selected group name

The name of the currently selected VisGroup. This can be edited by keyboard.

7Color

The colour used to draw all objects of this VisGroup in the 2D and 3D windows. Selecting the 'Color' button opens the color selector window from which you can choose from a selection of predefined 'Basic colors' or create some customized ones to your liking.

8Remove

Will delete the currently selected VisGroup from the list.

9New group

Will create a new VisGroup in the list. By default this name will be 'new group' which should be edited to something more appropriate such as 'Room1' etc.

1.7.1.6Texture Bar

The texture bar selects the current texture. Any solids created are made using the current texture, and the 'Apply current texture' button in the 'MapTools Bar' will use the current texture to change the selected solid(s).

See **Tools|Texture Application** for details on its use.

1.7.1.7Status Bar

The status bar is divided into 6 segments each giving various pieces of information depending on which window is selected and the current tool.

1



solid with 6 faces @-128, -192 64w 64l 64h

2Help

Provides a description of menu commands and toolbar buttons.

3Selection

Will give information on the current objects selections. If a single element is selected, it will give details information such as the entity type if it is an entity, or the number of faces if it is a solid. If multiple objects are selected, it will list how many.

4Coordinates

Will list the current cursor coordinates for the window.

5Bounding box/rotation/shear

When an object is selected, this area will display the bounding box size (i.e. the minimum sized box which can be drawn around the object(s)). The values are Width Length Height.

When you begin to do a move, resize, rotate or shear, the box will display information on the move.

If the type is 'move' (there will be solid black squares at each bounding box corner along with one halfway down each side and the cursor will be clicked inside the box) there will be no display.

If the type is 'resize' (bounding box will appear the same as for 'move' but the mouse will be clicked on one of the black squares) it will display the X/Y bounding box size for the current view.

If the type is 'rotate' (there will be solid black round dots at each bounding box corner) the area will display the rotation amount.

If the type is 'shear' (there will be solid black squares halfway along each side of the bounding box) it will display the amount of shearing.

6Zoom

Shows the amount of zoom in the current 2D window.

7Grid

Shows whether grid snapping is on or off, and the current size of the grid.

1.8 Windows

All editing is performed in the window views which may be either 2D or 3D. These windows can be arranged to your preference by using either the 2D and 3D options in the View pulldown menu, or the corresponding tools on the 'MapView bar'. See the **View Menu** for further details on configuration.

1 Traversing Windows

Resizing of the windows is achieved by clicking and holding the LMB on the cross bars separating each view; if selected in a vertical or horizontal section, movement is constrained to Y/X movements respectively. Selecting the bars in the very center where they cross over allows free movement.

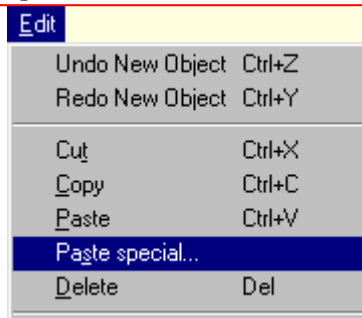
To get the windows back to default size, use the View|Autosize 4 Views menu tool (Ctrl+A).

The currently selected window can be expanded to full size by pressing the 'z' hotkey. Pressing 'z' again will shrink it to the original size.

If scroll bars are turned on (Tools|Options|2D views), the 2D windows can be scrolled around by clicking the LMB on the arrows, on the slider bar (for interactive scrolling), or in the space above/ below the slider. The arrow keys on the keyboard also perform the same operation.

Further details can be found in the section describing the **MapTools Bar**.

1.8.1.2 Paste Special



This menu allows multiple pasting of the clipboard contents with offsets and rotations. This relieves the work in creating something such as a picket fence where you can make one rail, then copy it multiple times with an offset for each new object.

1 Number of copies to paste

The number of new copies to paste.

2 Start at center of original

With this in its default position of on, offsets and rotations are relative to the location and rotation of the original item in the clipboard. With it off, the new objects will be created relative to the center of the two previously selected windows.

3Group copies

Turning this on will group all resulting objects together. See Tools|Group and Tools|Ungroup for further details on grouping. The default is off.

4Offset (accumulative)

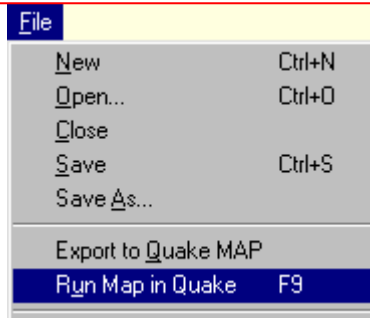
The three boxes here allow offset values for X, Y and Z. With each additional paste, the next object will be offset from the previous one by these values which are given in Quake units.

The '<' buttons beside each box will automatically insert the current grid settings.

5Rotation (accumulative)

These settings work in the same fashion as the Offset values, except that they apply rotation with each successive paste.

1.8.1.3Run Map in Quake (F9)



By selecting this feature, and after giving the project a file name if one doesn't yet exist, another window is presented with various compilation options:

1Additional Quake parameters

Here you can add extra Quake command line parameters such as -listen etc. which will be used when Quake is run.

2Run QBSP

QBSP is the program which compiles a .map file into a .bsp file which can then be loaded into Quake and played. You have the options of not running QBSP (which is only useful if the .bsp file already exists and you only wish to do other things to it), 'Normal' which will compile the entire map into a new .bsp, or 'Entities only' which will also require that it has previously been QBSP'd with 'Normal', only this time it will compile just the entities.

3Run VIS

VIS optimizes the level allowing it to run faster and smoother with no visible disappearing walls and textures. This is usually only done once a level is basically finished and ready for final testing since on a decent sized level it can take a LONG time (even days on very large levels and a slow PC).

The options are; no VIS, Fast (it's speedy, but isn't optimal) and Full which is generally only used for final release.

4Run LIGHT

Light is another optional step in the compiling process. Without running Light, every texture in the level will be fully bright, meaning that there will be no

shading or shadows. If you have any light entities in your level, running Light will give your level proper shading and shadows (if you run Light with no lights in your level, you will only see black).

The options are: no Light, Normal (is fast to calculate but shadows may be noticeably jagged edged) and Extra which will go a long way to removing jagged edges on shadows.

5Don't run Quake

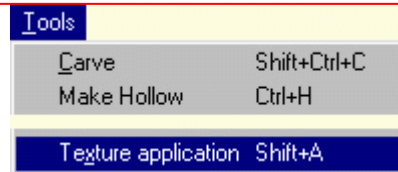
With this deselected, if the compilation is successful Quake will load up with your level ready for testing. Select this box if you don't wish Quake to load.

6Save visible objects only

With this in it's default off position, all objects in the level will be compiled, but with it selected on only visible objects will be compiled.

Note: Do not confuse the VIS utility with Worldcraft's **VisGroups** (visible groups) feature.

1.8.1.4Texture Application (Shift+A)



Allows pixel-accurate placement of textures on any selected faces. Faces are the individual sides/facets of a solid and can be alter individually or as groups.

If the current view mode for the 3D window is 'textured polygons' the changes made in the face will happen dynamically in the 3D view allowing textures to be lined up very accurately with little effort.

1Scale X Y

Will stretch the texture of the selected face(s) in the X and Y directions by the desired factor.

2Shift X Y

Will offset the texture by the selected number of pixels. The arrows to the right of each box can be used to offset with the mouse LMB. The longer you hold the button, the faster the stepping will become.

3Rotation

The rotational orientation of the texture on the face(s). The mouse can again be used to change rotation via the arrows to the right of the input box.

4Texture group

Worldcraft allows you to use multiple WAD files (a WAD the textures) and this selector lets you choose to browse ALL textures, or just those from a selected WAD.

5Current texture

The text field contains the name of the currently selected texture. Clicking the LMB on the button to the right presents a thin window containing all the

textures (based on the selection in 'Texture group') from which you can select a new current texture.

The currently selected face(s) will not gain this texture - use Apply to do this. All solids made are created with whatever is the 'current texture' at the time.

6Revert

Worldcraft, up to a point, remembers the previously selected texture, and 'revert' will make it the 'current texture' again.

7Apply

Changing the 'current texture' will not instantly change the selected face(s) to use that texture. Click LMB on Apply to give the selected face(s) this new texture.

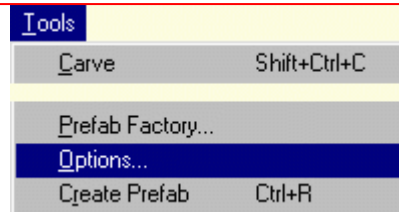
8Browse

Present a full screen view of the available textures (based on the selection in 'Texture group') for friendly texture browsing. Double click the LMB to select a new texture.

At the very bottom/left hand corner of the window is a selector which changes the viewing of the textures:

- 1:1 each texture is viewed at its full size
- 32x32 each texture is zoomed into a 32x32 box
- 64x64 each texture is zoomed into a 64x64 box
- 128x128 each texture is given a 128x128 box although no scaling is done

1.8.1.5Options



For setting up Worldcraft. All paths can be typed in by hand or by using the Browse option.

1Quake Location

This should point to your \quake directory (it may be something like c:\quake or c:\games\quake). This is used by the **Run Map** (F9) command.

2Maps location

Select the directory you wish to do your work in. This can be any directory you wish and doesn't necessarily need to be in the Worldcraft or Quake directories.

3WAD files

WAD files contain the texture maps for Quake levels. Worldcraft allows multiple WAD files to be used although by using this feature means that you must use the QBSP program supplied with Worldcraft and cannot use any of the other versions around. **Add** will let you add a new WAD file and **Remove** will delete the selected WAD from the list.

4Texture brightness

This slider determines how bright the textures will appear in the 3D view of Worldcraft. It has no effect on the appearance of textures in Quake.

5Display a crosshair under the cursor

With this option enabled, Worldcraft will draw a crosshair under the mouse cursor in the 2D views.

6Default to 15 degree rotations

When rotating a solid, having this option on will mean it will snap to 15 degree moves. This snapping can be overridden by holding down the 'Alt' key while rotating.

If this option is off, rotation can still be snapped to 15 degree movements by holding down 'Shift' while rotating.

7Display scrollbars

Turning this option off will remove the scrollbars from the editing windows. Movement can still be achieved even with this off by either clicking and dragging or using the cursor keys.

8Draw Vertices

By default every corner (vertex) of a solid is drawn as a little dot. Turning this option off will remove the dots.

9Hardware Acceleration

Some buggy 3D accelerators, such as the Diamond Stealth 3D and the Millennium, can cause problems for Worldcraft. If you find Worldcraft crashing consistently, try disabling this setting.

1.8.1.6Properties

The **properties window** is the center of all Quake activity. While good looking levels are certainly essential, it's careful use of entities that makes a level interesting/challenging to play, and that's what properties are all about.

In the properties window there can be up to 3 folders available depending on the type of object selected.

1Class Info

This folder will be available if the object is an entity, or a group of solids tied to an entity. Here you can choose on entity types and what keys will be used by that entity.

2Class

Class refers to the type of entity used. The selector can be opened by clicking in the text area or on the arrow to the right.

3Member of Groups

The large area to the left lists all VisGroups for the current map, along with their status (status is defined by a tick). There are no VisGroups by default, so to use them you must create some.

By placing a tick next to a VisGroup, you are placing the currently selected object(s) into that group. There are no limits to the number of groups you can

place each object into, so for example, a group of solids could belong to a group called 'Room1' as well as belonging to a group called 'doors'.

Every solid/entity/group created in Worldcraft has its own VisGroup settings which can be edited by the VisGroups window of the object 'properties'. The window can be opened by one of many ways: Edit|Properties menu, Alt+Enter hotkey, or RMB on a selected object in the 2D views. An object must be selected to use this window.