



Map paths



Console Commands

- **mpath_show** -- shows existing map paths for your team in developing mode
- **mpath_showall** -- shows all map paths, not just your team
- **mpath_hide** -- The opposite of mpath_show; make the debug view go away
- **mpath_auto** <0/1> -- enables auto mode, while moving you'll place a pathnode every 200 units
- **mpath_create** -- creates a path node at your current position
- **mpath_jumpoff_node** <nodeid> -- will set the parent node for to be used in mpath_auto mode
- **mpath_calculate_active_nodes** -- recalculate node network
- **mpath_move** <nodeid> -- moves a node to your current position
- **mpath_report** -- spams your console with a textual summary of all the nodes in the map
- **mpath_set_obj** <node> <obj> -- set an objective for the given node
- **mpath_save** <mapname> -- save path network to a file
- **mpath_load** <mapname> -- load path network from the given file
- **mpath_set_subobj** <id> -- If you set a node to be controlled by a subobjective, it won't allow the flow of "active juice" unless its team controls that subobjective. Setting a subobjective like this does NOT cause the node to be a "source" of active juice, it just makes it block when the subobjective is not held

HowTo

Map paths are done by using console commands while ingame; no entities required! You don't need hammer at all, or any other tool besides a running copy of the game.

To get an overall idea how those work you could load up any official map and type **mpath_show** in console to see what's behind the arrows you normally see in game.

Map paths are described by a network of nodes. Each node has a position in space, up to two children, and a few other variables which I will cover later. You can create a node at your current position with the mpath_create command. A node's children are the nodes that it connects to. It's like a daisychain of nodes, where 1 connects to 2, which connects to 3, and so on. You can set a node's child with mpath_set_child <parent> <child>. For example, to make node 45 be the child of node 102, you would type "mpath_set_child 102 45". A node can also have a second child, so you can make branches in the path. To do this, use mpath_set_child2. A node can only have two children, so if you want to have a three-way-branch you have to use multiple nodes (like 1 -> [2,3] 2 -> [4,5]) Merging branches together is easy; just make one node be the child of multiple other nodes (and there's no limit on how many can merge together in one place)

An easier way to place a bunch of nodes than using mpath_create over and over is to use mpath_auto 1. It makes you plop down a node every 200 units, just by walking around, and it strings them together for you. You have to use the variable mpath_jumpoff_node to control where your starting point is, though. The first node that gets created will be the child of the jumpoff node. You often have to manually create the very first node in a particular path, set it as the jumpoff node, and then turn on mpath_auto. This is kind of annoying; sorry.

So, there are a lot of different paths that can be defined in a map: there are different paths for each team on each objective. We have to be able to define when they are seen. Team is easy: whichever team you are when you're creating a path, that's what team that path is for. Objectives are more complicated. What you do is set an objective for a particular node with the mpath_set_obj <node> <obj> command. This has two purposes: the first is that it makes a node "active" during that objective. The "active juice" flows to all its children, and to its children's children, and so on forever. So they'll all be active during that objective. The second purpose is that an objective node can block "active juice" from flowing, if it doesn't match the current objective.

The nodes with green on top of them are currently-active nodes. It's currently objective 1. The nodes at the bottom of the screen are active, but the obj2 and obj3 nodes are NOT active. They are blocking the active juice because it's obj 1 right now. That obj1 node in the upper right is active though. In fact, even if the bottom nodes WERENT active, that obj1 node would be active anyway.

This complexity with blocking wouldn't be needed if we just made a *completely new* set of paths for every objective, but that would be a pain. We want to be able to reuse certain sections of path on multiple objectives, but not other sections, and this system is flexible enough to let us do that. Check out the paths on vaccine and you should be able to see how it works.

After you change around nodes, it won't update which ones are active until you tell it to, with the mpath_calculate_active_nodes command.

You can't delete nodes! If you find yourself wanting to, just reuse that node somewhere else. Or make it sit there alone with nothing pointing to it. (if you go into the txt file and try to delete a node that way, you will probably break it)

You can save and load paths with mpath_save <mapname> and mpath_load <mapname>. For example, "mpath_load dys_vaccine". Don't put .txt at the end. They're text files in resource/mappaths/ When you're working on these, be sure to save a lot because there's no autosave and it sucks when you get a crash! There's also no undo so be careful.

Sometimes you need to have tightly-packed nodes in order to get the logic right for what is active during what objective. This can make the particles turn into an ugly blob at that spot, so you can set a node as "hidden" with mpath_set_hidden <id>. A hidden node will still propagate active juice, but it won't draw any particles itself.

You will probably want to bind "mpath_show" to a key; I have it on mouse4. You have to press it a lot to see the names of the nodes.

Now hopefully you know HOW to make paths. Here are some higher-level issues about what paths to make.

Paths for the attacking team are easy: send them to the objective. But remember, when it switches to objective 2, there are people in the objective 1 area. You can't just have obj2 paths consist of spawn -> obj2. Those people hanging around in the obj1 area need a path to obj2. Same for 3, 4, however high it goes. Anyone in a "previous" part of the map needs a path to the current objective. (if you were wondering why there were obj1, obj2, and obj3 nodes right at the beginning of vaccine, this is why)

Paths for the defending team are tricky. Where should you send them? To the objective? To the enemy spawn? Don't give them any paths at all? I think there are different objectives where all three of these are the right answer. It depends on the specifics of the objective. We should probably send them to a "good" place. Think to yourself, "where is the most helpful place for a newbie to mill around" and send them there. Also, if there are any side routes that go to unhelpful places, it's probably a good idea to have a path going AWAY from that route towards a helpful area. For example: on silo 1 defense, you can go to the power jip. If a newbie wanders down there, he is in the Wrong Place and he is being Useless. There should be a path sending him back up the ramp to a useful area. There's a turret near there, and it is often hostile! We probably don't want to send newbies into turrets unless we absolutely have to (like on vacc 2 attack). So that path for the defenders should probably end on that little bridge.

I think it's a bad idea to send defenders towards an objective jip, because they will deck in uselessly. In fact, we probably shouldn't send defenders towards any jip, ever, since newbies decking is so disruptive to teams. Once a player is comfortable enough that they don't rely on mappath, then they can get into decking.

Alternate routes are a little tricky, too. If an alternate route is complex or dangerous, it might be best to not even give it a path. The important thing about paths isnt that they show all the options, but that they give players saying "where the hell do I go" a good answer. Go THAT way.

Retrieved from "http://www.dystopia-game.com/wiki/index.php?title=Map_paths"

Category: [Mapping](#)

navigation

- [Main Page](#)
- [Community portal](#)
- [Current events](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

search

toolbox

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)