Dwarf Therapist

Version 20.4 Guide

Resident Mario

November 2, 2013

Contents

Part I
Preamble

1 What's a Dwarf Therapist?

Dwarf Fortress is not an easy game to play, and many a stoic a gamer has fallen trying to learn how. A mind-bogglingly complex game wrapped in an extremely simple ASCII[1] character-based graphics set, Dwarf Fortress is renown for being impossible to learn, impossible to play, and impossible to master, and so caters to a certain, how to say, esoteric crowd. Luckily the kinds of people that play Dwarf Fortress also happen to be the kinds of people that enjoy laboring over computer programs that, among other things, aim to make playing the game less of a painful experience (and, luckily for you, the kinds of people that think laboring over guides for them would make good LaTeXpractice). Dwarf Fortress's community has lovingly crafted countless tilesets, character mappings, mods, graphical visualizers, companion programs, launchers, debuggers, memory access tools, plug-ins, extensions, program packages, so on, so forth, you get the idea, to extend and empower this (projected) 20 year project of a game.

One such gamer is Trey Stout (or chmod, as he's known on the forums) and one such program is Dwarf Therapist. Initially released way back in 2009, the program solves one of the most basic and annoying problems with the game - the difficulty involved in setting Dwarven labor preferences. In the vanilla game, the only way to set dwarven labor preferences (probably the most important setting there is, dwarf- wise) was to get to the dwarf, get to their labors screen, and then crawl through a tedious menu bumping this labor off or this one on. A starting group of seven dwarves? Not fun, but in grander context of things, doable. 200 of them running amok? No way, jose. Dwarf Therapist solved that problem with !!SCIENCE!!, or more specifically, a program that plugged into the Dwarf Fortress memory and allowed the reading, editing, and committing of dwarven labor changes, all from a pretty (and functional) graphical user interface (or GUI, in nerd-speak).[2]

So no, it's not designed to console your dwarves' psyche, prevent insanity, or relax your brain when you tie it into a knot trying to figure out why that Legendary Armorer ran *into* instead of away from that goblin ambush. Or why that mason just ran off the edge of a waterfall and drowned. Or why that ambusher decided to take a nap beneath your atom smasher. Or how that Werelizard managed to spawn in the middle of an ocean. Or why that crossbowdwarf ran off to bash an ogre fifty times his size with a featherwood crossbow and didn't even stop charging long enough to realize he was also carrying a steel short sword. Or...well, that's enough dwarven logical fallacies for one day (there's a "Dear Urist" forum thread for that). But it *will* make your life easier in another, more mundane way.

So, if all it does is manage dwarven labors, what's so difficult about it that it necessitates a fifty-plus page guide? As it so happens, chmod didn't release the basic version of Dwarf Therapist and leave it there. He continued working on it, polishing off its interface and new features; and then when he retired from maintaining it in 2010, DwarfEngineer took over its development through 2012, whereupon Splinterz revealed his code fork of the program in February 2013, which now stands as the primary, maintained branch of therapy. Years of development have expanded the capacities of the program well past basic labor management and have turned it into a pretty comprehensive fortress management tool, of which labor management is only the most obvious and immediately visible application. It has been made into a robust dwarf management tool, one powerful enough and, yes, complicated enough to sufficiently necessitate an awesome guide to go with it. Which is what you're reading, by the way. Let's get started, shall we?

[1] Technically Code Page 437, an IBM ASCII expansion sometimes referred to as "extended ASCII".

[2] Dwarf Therapist was not the first such program of its kind; before it most players used a similar utility known as Dwarf Manager. But the program became outdated and fell out of use, and Dwarf Therapist became - and has remained - the de facto standard.

2 Installation

The process for installing Dwarf Therapist on your computer differs a little depending on your system.

Windows: Since Windows is the base version for the utility, installing Dwarf Therapist on Windows is easy: just download this ZIP file in the Dwarf Fortress File Depot and then extract onto somewhere close to the game files on your computer. The program will look through your active programs to try and find the game, so you should generally be safe running it from just about anywhere, even a thumb drive, but it's most logical to stick it in the Dwarf Fortress root directory - the one with the game's .exe file. Here is the link: `http://dffd.wimbli.com/file.php?id=7184`.

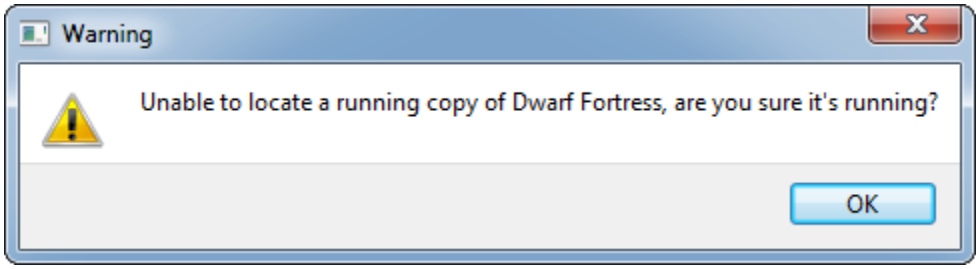
OSX: A working version of Dwarf Therapist has been created, and is available here: `http://dffd.wimbli.com/file.php?id=7842`.

Linux: A Linux-based version of Dwarf Therapist is available for download from the project page: `http://code.google.com/r/splintermind-attributes/`. Note that it is still considered in alpha, however. You can also build it from the source with the instructions available here: `http://code.google.com/p/dwarftherapist/wiki/BuildingDwarfTherapist`.

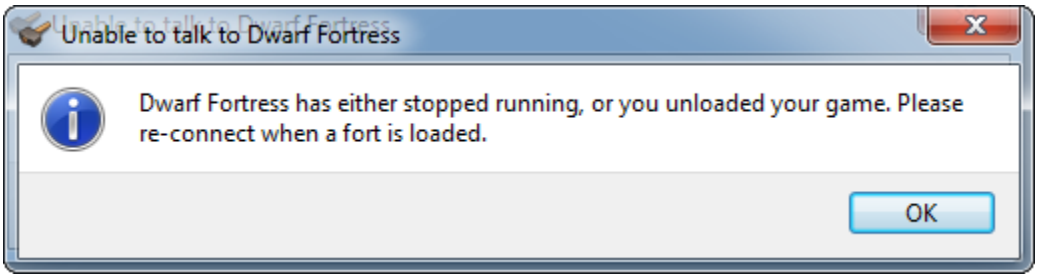
Part II
Basics

3 Connecting to Dwarf Fortress

Dwarf Therapist requires Dwarf Fortress to be running and an active fortress to be open within the game in order to work properly. You can open the utility without the game running or without a fortress loaded, but it will complain to you:



This is because before Dwarf Therapist can do much of anything useful, it has to connect to a running instance of the game; the only function that the utility can provide by itself is the ability to write filter scripts (which obviously can't be applied out-of-game), move the docks around, and change some settings. You'll get another, similar error message if you load a fortress, connect to the game, and then unload it:[3]



To avoid such messages you should generally launch Dwarf Therapist second and close it first, but it doesn't really matter too much - you can connect at any time using the "Connect to DF" button on the main toolbar.

To keep its memory usage down, Dwarf Therapist connects to your game intermittently, loading data into the utility only when a fortress is first opened or when you request to do so. This is the function of the second button on the main toolbar, "Read Dwarves", which will easily be the most used item on your main toolbar.[4] Loading data into the game only takes a fraction of a second to a few seconds (depending on the size of your fortress) each time, and it can be done regardless of whether the game is paused or not - but since you're probably going to be spending some time on the Therapist screen once you switch to it, pausing is probably a good idea.

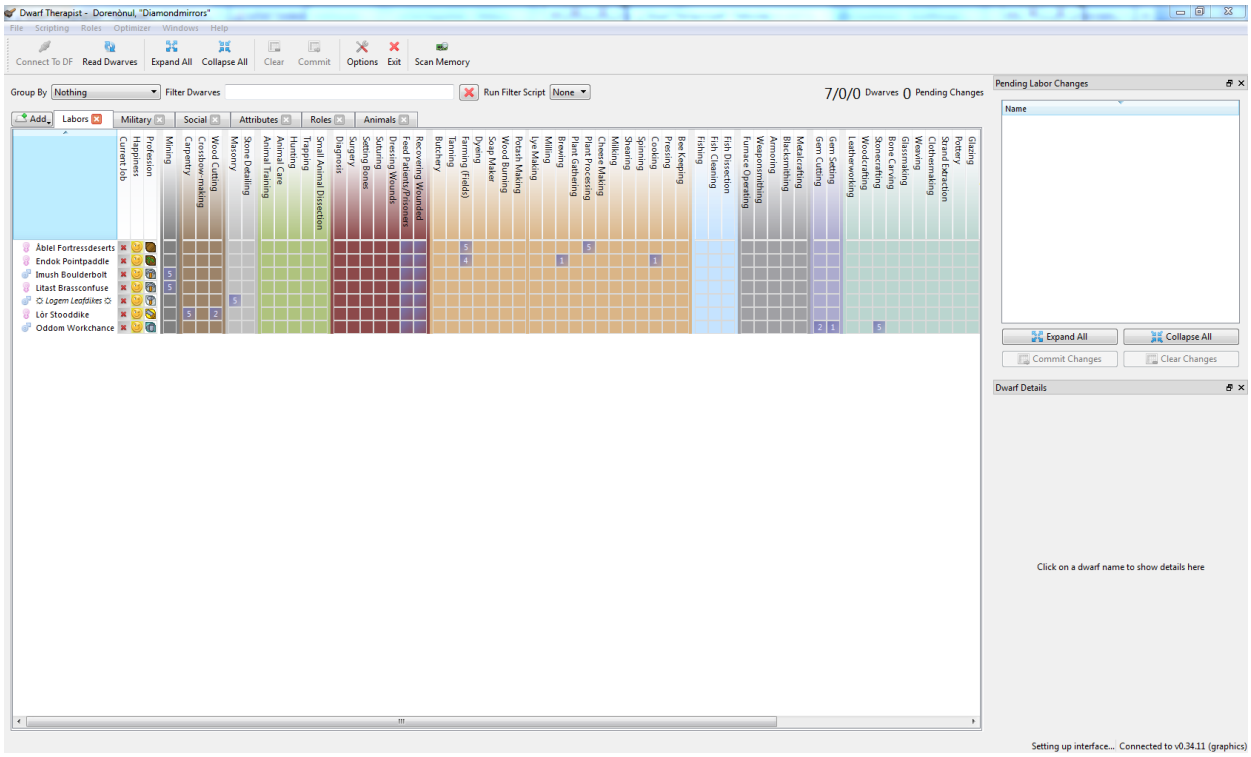
Figure 1: The two most important buttons on the main toolbar.

[3] This behavior can be disabled in the general options. However, this is no "offline" mode: cycling out of a fortress, making changes, loading it again, and then attempting to commit your offline changes will only cause the utility to crash.

[4] Since Dwarf Therapist now lets you view animals, too, the name is a bit of an anachronism. If you mod your game and change the playable race, the button label will actually change with it - one of Dwarf Therapist's many mod-friendly features.

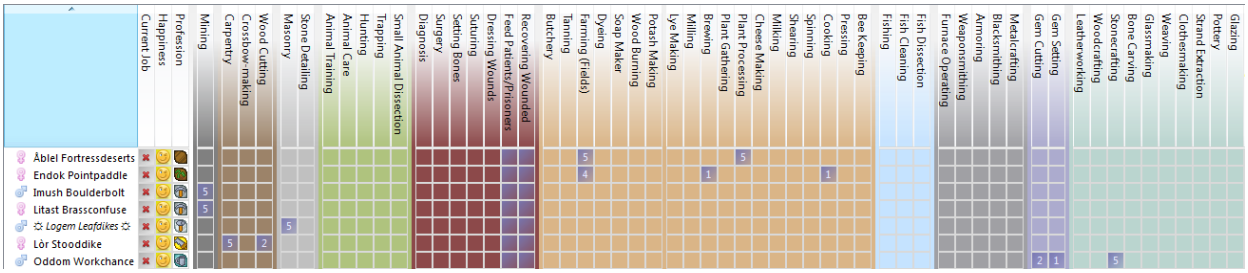
4 Main Display

So now that we've gotten connecting to and loading from the game out of the way, let's create a new fortress and launch Dwarf Therapist for some early dwarven task management. Your screen should look something like this:



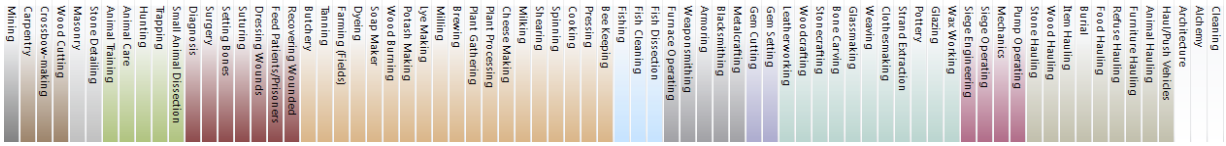
4.1 Labors View

Most of the space on the screen is taken up by the graphic labor management user interface, which is the main window that takes up most of the program's screen place. Since there aren't that many dwarves in our fortress yet, most of the screen is whitespace.



The primary feature of Dwarf Therapist, the labors view is the most immediately accessible part of the program, and it is around this screen that the entirety of the program is built; in fact, changing labor preferences is all of what many people use the program for. Which is fine, really, but in this guide we're interested in learning how to leverage the utility in its entirety, so the labors screen is only a part of what we will cover.

The first thing to note about this view is that there are more labor preferences available in the game then can be comfortably and compactly fit into the screen, necessitating that we contend with a scroll bar at the bottom of the screen on all but the widest of monitors; a second one appears on the right once your dwarves get too numerous (vertically) to fit into the interface.[5] The labors themselves are organized into "tranches", the same way they are in-game, with related labors categorized and sorted under the same color assignments used by dwarves of those professions in-game:



Most of these labors should be familiar to players of the game, as should be their categorical professions: miners, woodworkers stoneworkers, rangers, doctors, farmers, fishery workers, metalsmiths, jewelers, crafts-dwarves, engineers, and (for hauling) peasants. The only novel labors in the interface are alchemy, which was a game feature many a version ago that was replaced by soaping (but is expected to return in a future version of the game), and cleaning, which is novel mostly because dwarves do it so rarely.

To the left of and immediately below the professions list are your dwarves, ready and waiting to receive orders. After their names and genders the first box there is an icon representing their current job - since we jumped into this view immediately after the map loaded, none of them have jobs as of yet, and so it's red Xs all around. The next icon is their current happiness, represented by a red-to-green color gradient: as you might surmise, the greener the better, the happier the dwarf. The last item in the list is their dwarven professions, as listed in the game - a fast view of who's who in the fortress.

Clicking on the columns at the top of the screen allows you to sort the list by that column. Click on the big name spacer and you'll get your dwarves listed alphabetically, backwards and forward; on "Current Job" and they'll be sorted by Job ID; on Happiness and they'll be listed by their happiness level (something that's only indirectly visible in-game); on "Profession" and they'll be listed by profession ID.[6] Finally, hit one of the professions tabs and you'll rank your dwarves by their skill in the task at hand.

Within the grid itself, the numbers within the cells tell you how skilled the dwarf is at that task, while which ones are shaded in and which ones aren't tells you which labors they have enabled. Try clicking on a few of the cells and seeing how the program responds - getting labor changes into the game should be entirely intuitive, even if we're not going over it in detail just yet.

A big part of the interface is in the tooltips, which pop up whenever you hover over a space in the grid. Hovering over the labor column headers will tell you how many dwarves have that labor enabled, while hovering over individual labor cells will display the descriptive (Dabbling, Novice, Adequate, Competent, etc.) and raw (0, 1, 2, 3, etc.) skill that dwarf has in that labor, their cumulative experience and progress towards the next skill level, whether or not that is their highest moodable skill, and (if they are legendary from a strange mood) the name of the artifact they crafted; if the skill has rusted, that will be displayed as well.[7] Hovering over happiness gives you their exact happiness score, over their profession gives you their in-game job title and profession ID (used for sorting), and over their current job gives your the numerical ID of that task (also used for sorting in a similar manner to professions; see **"Using Sorts"**).

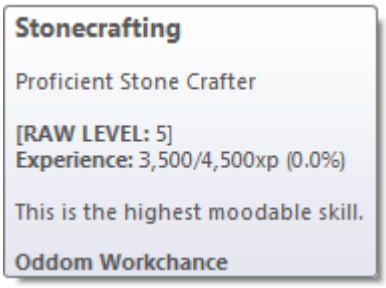
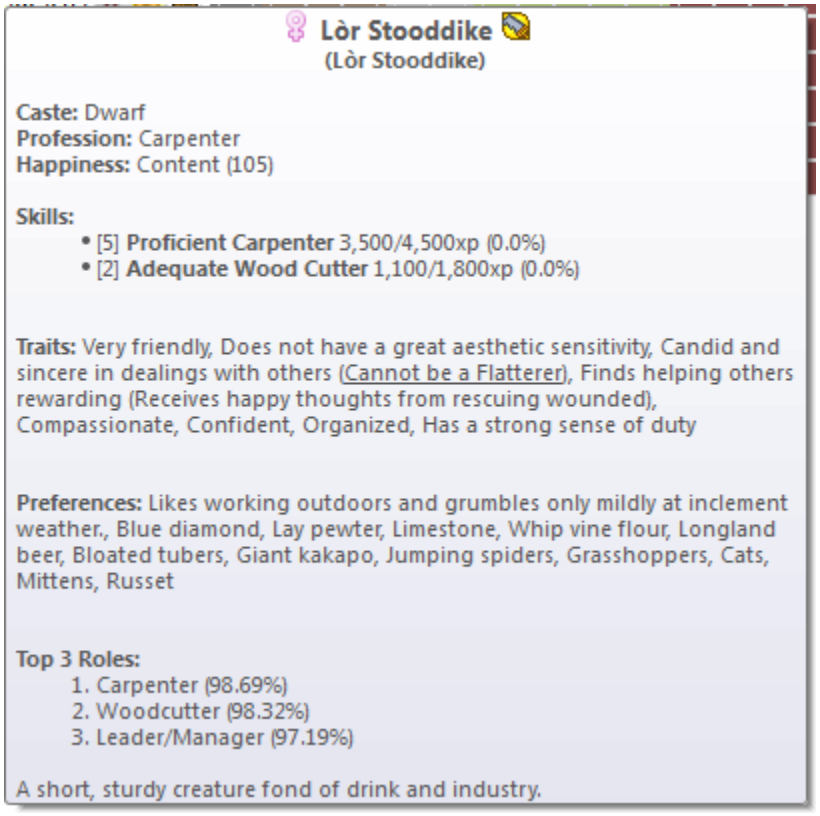


Figure 2: The labor tooltip.

[5] For tips on formatting your display to make it more compact, see the section **"Formatting Your Display"**.

[6] Job IDs and Profession IDs are rather abstract lists that numerically organize of their relative items in no particularly well-sorted order. Applications for and the modification of IDs will be discussed much later in this guide. For further reference, see **"Modifying Game Data"**.

[7] **Skill rust** occurs when a dwarf has not performed the labor associated with a skill for a long time. Dwarves whose skills are "Very Rusty" will eventually see their skill in the labor drain away or even vanish completely.



By far the most information-dense tooltip display here is for the dwarves themselves. Hovering over their names will give you their caste[8] and gender, their profession, their happiness level, their thoughts (color coded green when good, red when bad), their skills (rust will be highlighted), their traits (and anything they can or cannot do as a result of them: this dwarf, for instance, cannot gain experience in the "Flatterer" social skill, and would make a good stretcher-bearer), their preferences, and their calculated "Top 3 Roles", based on their traits (for a detailed discussion on this aspect of the program see **"Roles"**). The caste description that serves as the dwarven tagline also makes an appearance, at the end of the tooltip. Notably, the first line in the preferences section tells you how much cave adaptation the dwarf has experienced: this dwarf is fine with working outside.[9]

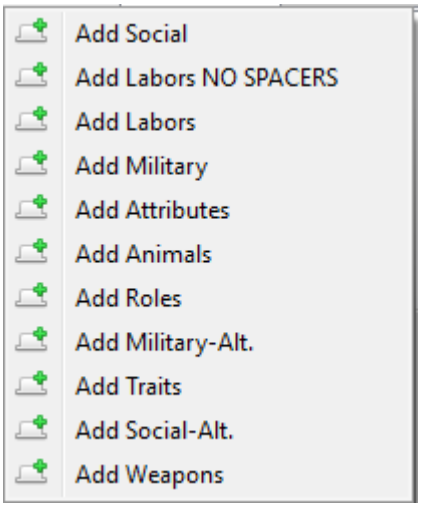
[8] **Castes** (raw file definitions used to define species or sub-species) are not very useful in vanilla Dwarf Fortress, but having a display for them is handy if you're using mods and have fiddled with your playable race.

[9] Dwarves that spend a lot of time underground (on "Dark" tiles) slowly develop cave adaptation, becoming "irritated by the sun" or worse, "nauseated by the sun". If exposed to direct sunlight (a "Light" tile) they will have an unhappy thought in the former and start vomiting too in the latter case.

4.2 Views

Immediately above the labors screen is the screens tab:

Although the Labors view is the core of the interface, and is the tab open by default, there's a number of other screens built into the utility, all of which have their uses. The screens tab has several of these tabs open by default, and more are available under the "Add" menu. The tabs can be moved around, deleted, and re-added as much as you want, and the utility will save their state between instances, allowing you to keep your favorite tabs around where you'd like them. You can also have multiple instances of a tab open - although this isn't a particularly useful feature, it's there. A list of the screens and what they do follows:



Social: Catalogues your dwarves' social skills. The tooltip lists related roles, which are only present for some of the social skills.

Labors NO SPACERS: Same as the option below, but without any spacers between the categorial professions, winning you back some valuable screen space. To learn how to rename tabs, jump ahead to **"Creating Your Own Grid Views"**

Labors: The Labors View is the program's primary view and the one it opens up to by default; for a detailed description see **"Labors View"**.

Military: Allows you to view your dwarves' combat ability. It lists the relevant combat skills in melee, fighter, equipment, miscellaneous, and marksdwarf tranches, then gives you a view of their combat-related attributes, then their compatibility scores for various combat roles, then which weapons they are or are not able to equip based on their relative size.[10]

Attributes: A basic list of attributes. Tranches are physical, then mental.

Animals: Lists domestic, caged, and tame creatures and their attributes. Includes whether or not they are marked for butchering and whether or not they are caged. Distinguishes between children and adults, and lists current training levels.

Roles: Lists the role ratings of your dwarves, a score out of 100 that tells you how adapted they are to that profession. This is a fairly advanced topic covered in detail in the section **"Roles"**.

Military-Alt: Expands on the regular Military view by including role ratings.

Traits: This is a special view that lists raw role ratings; see above.

Social-Alt: Expands on the regular Social view by including role ratings, as well as a non-social trait columns for traits are affected by social traits. For example, Assertiveness is added and grouped with Persuader, because having low assertiveness means it's impossible to increase the persuader trait.

Weapons: Lists dwarves by which weapons they can wield. This view is dynamically generated: in vanilla it acts as a subview of the military view, but mods with large amounts of weapons will look significantly more exansive. Additonally, the weapons view will be tacked on to the military view as well if there are ten or fewer weapon groups.

[10] Theoretically all the but the runtiest of dwarves are able to wield all but a few especially large non-native weapons one-handed; however, because of a bug, in Fortress Mode *no* dwarves can wield two-handed weapons, ever.

4.3 Group By and Filters

Filter scripts are an advanced topic covered in **"Filter Scripts"**, and will not be covered here, but the "Filter Dwarves" input field is quite simple: it just lets you filter your dwarves by name, so as to make it easier to find one you're looking for this way. Click on the red X to quickly remove that filter.

The Group By menu, meanwhile, changes what your dwarves are sorted by in the views, and is fairly intuitive, mostly useful for examining various demographic breakdowns of your fortress. Options are Nothing (default), Age (in groups of ten), Caste (in vanilla, all dwarves, except in the Animals view),[11] Current Job (used to catch idlers), Happiness (by stage, you're better of sorting for this one), Has Nickname (yes/no, useful when you're naming dwarves), Highest Moodable Skill, Legendary Status (yes/no), Migration Wave (the utility's best historical sort, and therefore a go-to grouping), Military Status (active/off duty/can activate/noble), Profession (again you're better off sorting skills), Race,[12] Sex, Squad (No Squad, Squad Name One, etc.), Total Assigned Labors, and Total Skill Level. For more on using groups, see **"Using Groups"**.

When a group is active, the view will be expandable/collapsible, and in the collapsed labors view will display the lowest happiness of the group, as well as (for each skill) if anyone in the group has the skill enabled (which results in a darkened box), or if they all have it enabled (results in a bright green box). This feature can be toggled on or off in the options. Groups consisting entirely of children will appear bright green.

Additionally, there is a population indication in the top-right corner that gives you an at-a-glance view of how many dwarves there are in your fortress, distributed adults/children/babies, and the number of dwarfwise changes you have pending (more on that later).

Figure 3: The population statistics of a mid-sized fort.

[11] When mods are active an additional "Caste Tag" option becomes available, useful for finding hidden or secret castes that may exist.

[12] A creature's race is defined off of its raw file, and will be the same as its caste in most cases. The exception is when a sub-special caste exists - in which case, race and caste does in fact diverge.

4.4 Main Toolbar

Figure 4: Right clicking on the main toolbar allows you to toggle it (and docks, covered in the next section) on and off. It can also be accessed under the "Window" menu in the main menu.

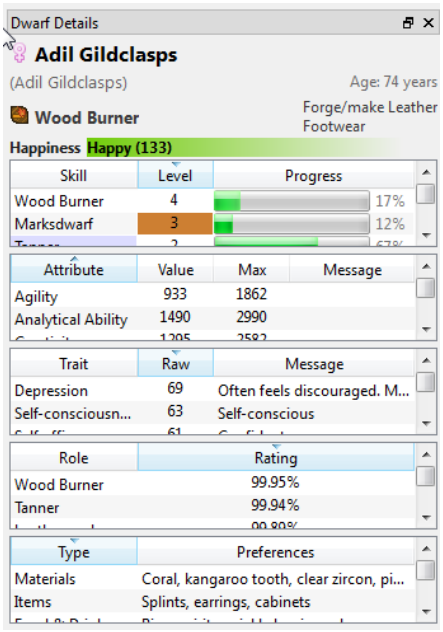
The main toolbar is just that, the main toolbar. Here it is
that the most important "shortcuts" of the program are presented.

- **Connect to DF**: If Dwarf Therapist is not already connected to Dwarf Fortress, this will attempt to connect the program. If it is, this will be grayed out. Shortcut `Ctrl + C`.
- **Read Dwarves**: Reads the current state of the dwarves in your fort into the Therapist. Shortcut `Ctrl + R`.
- **Expand All**: Expands all group views; if Group By is set to "Nothing" this will have no effect. Shortcut `CTRL + >`.
- **Collapse All**: Collapses all group views, doing the opposite of the above command. Shortcut `CTRL + <`.
- **Clear**: Clears all labor preference changes (see **"Managing Your Dwarves"**). Shortcut `Ctrl + E`.
- **Commit**: Commits labor preference changes (see above). Shortcut `Ctrl + T`.
- **Options**: Allows you to adjust settings. Will be covered in the section **"Options"**.
- **Exit**: Exits the application.
- **Scan Memory**: An advanced memory mapping utility that we'll touch upon at the end of this guide: see **"Addendum: Scanning Memory"**.

4.5 Docks

When you first open Dwarf Therapist, the right side of the screen will host a couple of docks. These two marvelous boxes are used to keep pace with what you're doing and modify how Dwarf Therapist works in various ways. Up to a trio of docks can be attached to the right side of the screen, and others can be pinned against other cardinal directions (or simply left floating around). Clicking the window button detaches the dock and leaves it to float, while clicking on the X button closes them, giving the others more room. If you don't have any docks at all pinned to a the right side, the main view will expand to make use of the new screen space.

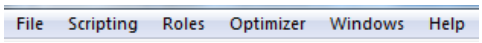
By grabbing and dragging a dock you can actually attach anywhere on the screen - to the left, above, below, or to the right of the main view. By grabbing and dragging it into another dock, you can tab them, so that the dock that is currently open gets to monopolize the screen space. And the current dock configuration is saved between instances of the program, of course. Note that the main menu is actually also a dock, albeit one that must stay independent when docked, and so can be moved anywhere or even gotten rid of entirely. A list of docks follows:



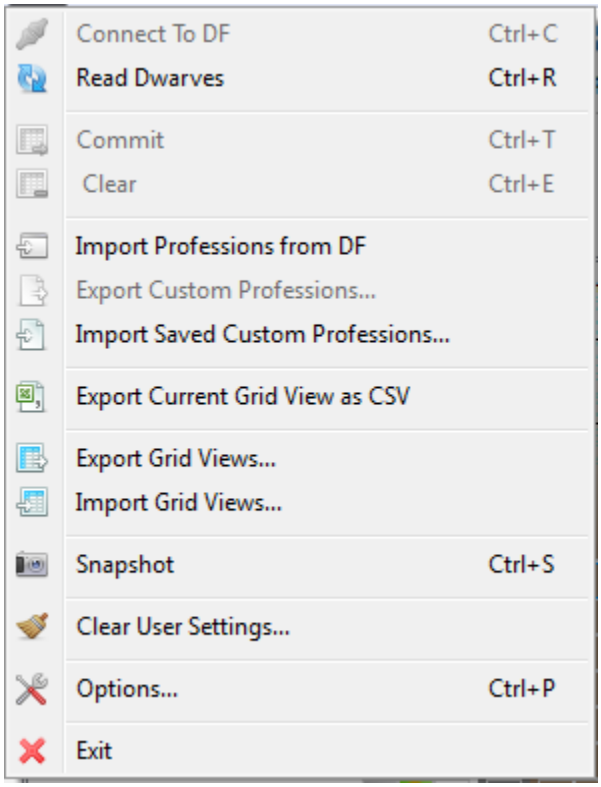
- **Dwarf Details**: This one lists a lot of stuff, so if you use it you should tab-dock it. Dwarf Details provides an expanded view of a dwarf's abilities, providing pretty much all of the information on their capacities in the various views (and then some) in one comprehensive tabular form. You've got name, translated name, age, profession, current job, happiness level, skills table with progress bars, attribute table with maximum trainable value and message text, raw trait value with message text, top ten role ratings, and a table of the dwarf's preferences.
- **Pending Labor Changes**: This dock displays which labors you are turning on or off in a tabular form (it also records the addition and removal of personal nicknames), and allows you to commit or clear your changes independent of the main toolbar. There are also buttons for expanding and collapsing labor changes displayed in the dock, useful when making mass labor swaps. For a discussion on Managing Your Dwarves (I'm getting there, I promise!) see **"Managing Your Dwarves"**.
- **Custom Professions Icons**: This dock allows you to create custom professions and custom icons. For a discussion on creating custom professions see **"Creating Custom Professions"**.
- **Grid Views**: Allows you to open grids from a menu, or create entirely new ones of your own design. See section **"Creating Your Own Grid Views"**.
- **Skill Legend**: Provides a legend for skills display onscreen, and allows you to quickly change it with a drop-down menu. Not really very useful.
- **Preferences**: Lists your dwarves by preferences, and allows you to search through them by object of preference. Clicking on one or more of the preferences allows you to filter dwarves so that only those with that preference are displayed on-screen (and Clear Filter obviously clear the filter).
- **Thoughts**: Similar to the Preferences dock, this brings up popular thoughts by count, allows you to search through them, and allows you to filter your dwarves by them.

4.6 Menu Bar

The last piece of Dwarf Therapist we're going to analyze is the iconic menu bar (or taskbar), present on almost every real application ever written. I'm going to give a brief list of what's in it here, and direct you to the sections for specific functions when appropriate.



File



Connect to DF `(CTRL + C)`. See **"Connecting to Dwarf Fortress"**.

Read Dwarves `(CTRL + R)`. See **"Connecting to Dwarf Fortress"**.

Commit `(CTRL + T)`. See **"Managing Your Dwarves"**.

Clear `(CTRL + E)`. See **"Managing Your Dwarves"**.

Import Professions from DF. See **"Exporting and Importing Professions"**.

Export Custom Professions. See **"Exporting and Importing Professions"**.

Import Saved Custom Professions. See **"Exporting and Importing Professions"**.

Export Current Grid View as CSV. Exports the current view as a "comma exported value" spreadsheet, which is a basic `TXT` that can be read by almost any spreadsheet software.

Export Grid Views. Exports, in Dwarf Therapist's `DTG` export format, a chosen selection of grid views to a chosen file location. See **"Exporting and Importing Grid Views"**.

Import Grid Views. Imports `DTG` grid views from the disk. **"Exporting and Importing Grid Views"**.

Snapshot `(CTRL + S)`. Takes a snapshot of the currently active Dwarf Therapist window, and stows it away where you tell it. Baically an extended version of the `PRNT SCR` key.[13]

Clear User Settings. Deletes all user settings and then exits Dwarf Therapist. This restores all settings in the program back to default and erases all data, which is why it has a warning screen - it can be very damaging if you have complex scripts and other goodies programmed into the utility.

Options `(CTRL + P)`. Also provided on the Main Toolbar; see **"Options"**.

[13] This option is functionally limited to full-window shots, and so should not be used too extensively - the screenshots for this guide, for instance, were done with Greenshot.

Exit. Immediately exits the program.

Scripting
Provides facilities for generating Filter Scripts to apply to your dwarves. For a detailed discussion on Filter Scripts, see **"Filter Scripts"**.

Roles
Provides facilities for creating, modifying, removing, importing, and exporting custom roles. For a detailed discussion on Roles, see **"Roles"**.

Optimizer
Allows the creation, modification, deletion, importation, and exportation of Optimization Plans. For details, see **"Optimization Plans"**.

Windows
No, not the operating system. Allows you to modify the docks and main toolbar displays in a manner similar to right clicking on the main toolbar. For more information on Docks, see **"Docks"**; for more information on the main toolbar, see **"Main Toolbar"**.

Help
Project Homepage. Provides a link to the project homepage: `http://www.code.google.com/r/splintermind-attributes/`.

Discussion Forums. Provides a link to the main Dwarf Therapist forum thread: `http://www.bay12forums.com/smf/index.php?topic=122968.0`.

Request Feature / Report Bug. Provides a link to the Dwarf Therapist issue tracker: `http://www.code.google.com/p/dwarftherapist/issues/entry/`.

Donate. For buying the poor developer a beer through PayPal.

About. Brings up a small splash screen giving you the Dwarf Therapist version number, some accreditation links, and a link to check for updates.

5 Managing Your Dwarves

Now that we've finally pinned down all the wayward bits of Dwarf Therapist and explained, though in some advanced cases quite briefly, what each of them does, we are in a position to discuss the game's primary source of utility: its ability to change the game's dwarven labor preferences, without having to deal with the game's clunky dwarfwise interface. This section will cover the basic tenets of dwarven management and demonstrate why even when Dwarf Fortress updates that have been in the works for a year or more are released, many people still refuse to play until Dwarf Therapist is updated to match. If you've already gotten dwarven labor management down pat, you can skip ahead to **"Options"**, or to **"Advanced Features"** if you're itching to try out the program's more advanced features.

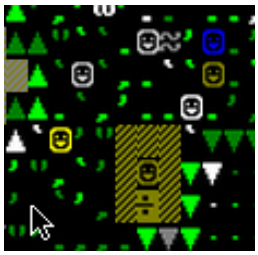


Figure 5: Dwarves frolic by the caravan.

5.1 Making Labor Changes

Now let's return to the seven humble dwarves we touched upon at the beginning of this guide, illustrated above. We've got two miners, a carpenter/woodcutter, a mason, a stonecrafter/broker, and two farmers. The woodworker is soon to be off cutting wood, and the two miners are soon to be off digging - but what should the other four dwarves do? The facilities for their professions haven't been built, and there's nothing to haul around yet. This is a recurring problem, but I look around and see that there are plenty of bushes lying around that can be stripped for some free early food (and seeds for an above-ground farm, later on). So I designate some plants for gathering, and then change the labors to get my dwarves to do some work for me.

Individually designating dwarven labors for changes is as simple as clicking on the boxes that correspond with that dwarf and that task in the labor view. The box with either fill or unfill and will be surrounded by a bright red border, and the exact nature of the labor changes will be added to the pending labor changes dock if one is present on your screen and notched onto the Pending Changes counter near the top right of your screen; the Clear and Commit buttons on the main toolbar will greenlight as well. The changes that we would like to make and how they appear on-screen are highlighted on the left. To revert a change you're making - if, for instance, you accidentally toggle Plant Gathering on for your Carpenter, when what you really want him to be doing is chopping trees - just click on it again to revert it to the previous state.

We've made some labor preference changes, but right now they're only hanging around in the "Pending Labors" queue in Dwarf Therapist. To make them actually appear in the game, we have to **commit** these changes. Here's what the changes we want to make look like in the Pending Labor Changes dock, expanded for clarity and collapsed for compactness (which is what the buttons do, if you didn't know already):

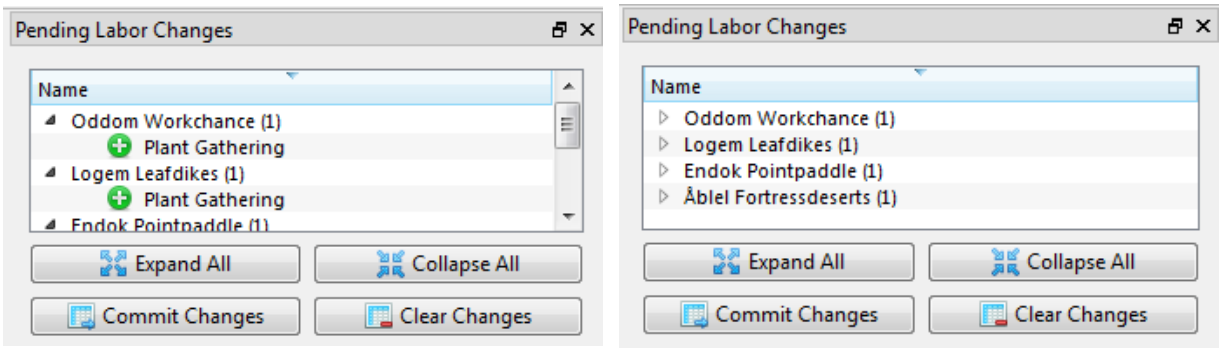


Figure 6: You can remove the need for collapsing changes by giving the dock a bit more breathing room.

Hitting "Clear Changes" will clear all of these changes and wipe Dwarf Therapist to its previous state, while hitting "Commit Changes" will send the new labor orders through to the game, updating Dwarf Therapist to reflect the changes and producing immediate results. These buttons are available on both the Pending Labor Changes dock and on the Main Toolbar, however they also have hotkeys: `Ctrl + E` and `Ctrl + T`, respectively. Since these are two of the most common operations you'll be conducting with the program, it's probably a good idea to remember these hotkeys - it will save you a lot of time (refer to **"Hotkeys"** for the list of available hotkeys).

Dwarves start off with certain labors enabled: labors they have above-dabbling level skill in, hauling labors (including the two medical ones), and cleaning are always going to be set to on. The presence or lack of hauling labors especially is a concern: if we want the dwarf to be entirely focused on their primary tasks, these should be turned off, but if you want them to help with hauling away loose stone, feeding the wood stockpiles, and so on, then these should be turned on. For instance, I always dedicate my miners 100 percent to their task, and turn off all of their other (non-cleaning) labors. But there's a lot of small hauling labors, and turning them on or off individually is a pain. One solution is to hold down the mouse button and drag your cursor across the labors: this will toggle every labor you pass through on or off. But it's still not that fast, and it's pretty easy to mess up and accidentally toggle a nearby dwarf's labors on or off with it, which requires backtracking, which is a waste of time.

Thankfully, there's a better way! Go to any one of the labors in the tranche and right-click on it. Lo and behold, an option to toggle all jobs in that grouping on or off appears! I'm going to use this now to quickly and seamlessly turn off hauling for my miners, and dedicate them to their labors. This toggle feature is mostly useless for professions labors, but devilishly handy for designating hauling on or off when and where you need it.

Figure 7: Mischief, managed.

5.2 Using Groups

Now, picking out who's who on the labors view is easy enough when there's just seven dwarves to deal with. But it becomes quite a bit more problematic when there's say, seventy of them running amok:

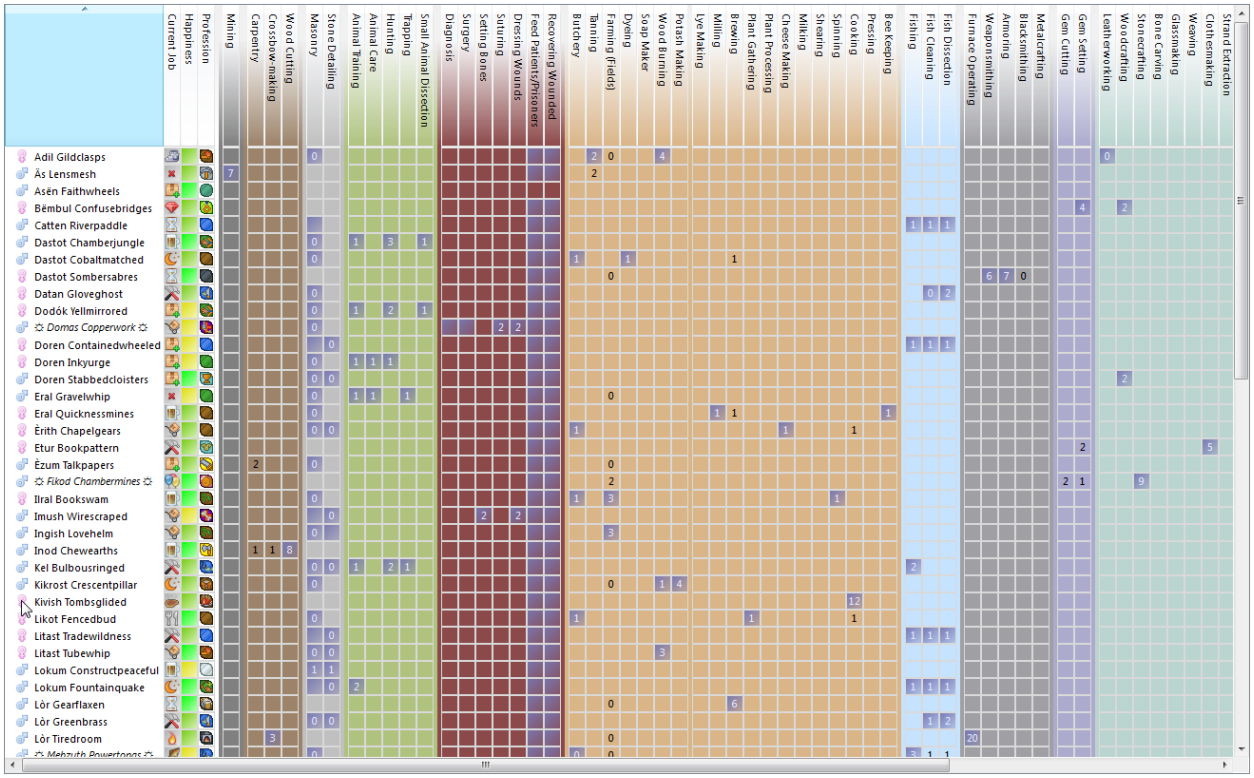
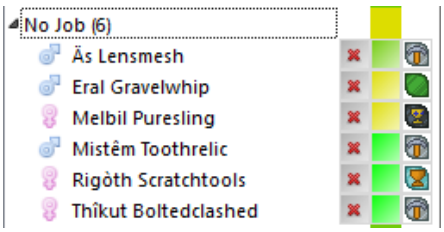


Figure 8: Who's who? Beats me.

This is when two of the built-in sorting features of Dwarf Therapist, Groups and Sorts, become useful (you can also use filters, but that's a more difficult topic for later: see **"Filter Scripts"** if you're impatient).

First, let's talk in terms of Groups. Groups are described in brief in the **"Group By and Filters"** section of this guide, somewhere way above here: refer to it again if you need a refresher. Right now I return to the game screen and discover that six of my dwarves are idling - not bad in fortress of this size, but I've always been one to keep my dwarves' hands as busy as possible; don't want them making friends in my dining room and then tantruming about it later. Maybe it's a lost cause - it appears all the useful ones are out partying - but we'll try anyway.[14]



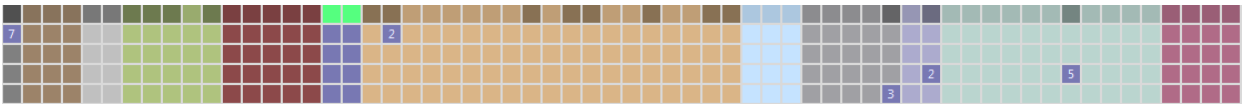
In a large enough fortress, trying to find out which dwarves are twiddling their thumbs is a fantastically annoying chore. Luckily, we can simplify the task for ourselves by grouping our dwarves by their current job. Going to the "Group By" drop-down menu and selecting "Current Job" will nicely clump our dwarves by - what else - their current jobs. After a little bit of snooping, I'm able to figure out what the problems are. We've got three jobless Miners with nothing to mine, a Woodcrafter (but really Thresher) with nothing to process, a Metalcrafter with nothing to craft, and a Ranger with nothing to haul. The Ranger will very soon find himself work, a spot of designation gets the Miners up and

[14] Actually you can stop a party dead in its tracks and return the partying dwarves to their labors by freeing the room in which the party is taking place, and then redesignating it. The more you know...

at it again,[15] and I add some processing jobs to an idle Farmer's Workshop. Bam; idle hands, managed! Ah, but what's this:

Migrants! What are we going to do with them? After letting them get used to my fortress surroundings, I open up the Dwarf Therapist, group by Migrant Wave, open up the new guys, and am now ready to do some easy-to-make-out management; as an added bonus it'll tell you exactly how many dwarves you're getting this season. Dwarves are loaded into the memory the instant the message appears, so they don't actually have to be on your map yet for you to start working with their labor designations - a neat feature.[16]

Now, whenever you group your dwarves the top row will consist of a collapsible group name and header, and a series of labor boxes that, based on whether they'll filled out or not, tells you which labors have been enabled within the group. Since these are new migrants and I haven't make adjustments yet, it basically tells me what skills the random number god has gifted me with this immigratory wave:



By clicking on these headers you can enable or disable a labor for an entire grouping. Since we don't want the new migrants idling, but haven't yet made facilities for most of them to use, let's do what I always do when a new migrant wave arrives: mass designate stuff for them to do, in bulk. Well, it just so happens that I've got a wall that I've been meaning to build, and there's quite a lot of fortress surface that needs to be smoothed out (and don't even get me started on the many pictures of cheese that we need, but lack!). So I click on the Masonry and Stone Detailing headers, toggle them on for the entire wave (you can also right-click on the headers), commit, and voila - stuff for them to do!

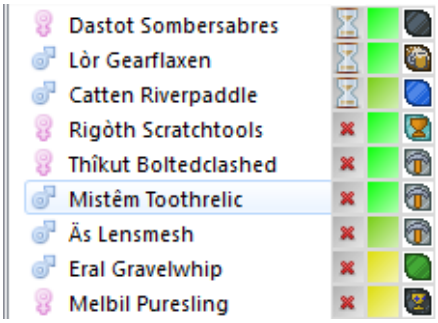
Now, these are just two of the most immediately useful scenarios for which grouping comes in handy, and there are quite a few more groupings that you can make and display. Additionally, if the correct option is enabled, these groups carry across all of the views, not just the current one! So if you want to see a demographic breakdown of your dwarves, group them by Age or Sex. To see which of your dwarves are Legendary, group them by Legendary Status (or hit Highest Skill for a more inclusive view). To check up on happiness, hit Happiness. To start working with Nicknames, hit Nicknames (this will prove useful in **"Assigning Nicknames"**, still ahead of us). To see what the chances of you getting a Legendary Armorsmith are, hit Highest Moodable Skill. To sort them by Military Status, click that, or Squad to check up on individual squads. To sort by Profession, hit profession. Use Collapse All and Expand All to switch between detail levels without having to manually toggle groups; for expedience you'll probably want to remember the associated `CTRL + <` and `CTRL + >` shortcuts.

However, some of these functions we've named Groups for are better and more easily approached by using Sorting; and that's what the next section is about.

[15] I cannot recommend the utility DFHack, and specifically its digv command, enough. It makes digging designations *so* much easier!

[16] Though it's generally reliable, when loading in dwarves from an auto-pause, the program may occassionally miss a couple. If you want to "play it safe", wait until the first few dwarves have made it onto your map.

5.3 Using Sorts



Grouping is a powerful tool, but it can be annoying - the items are sometimes sorted oddly, and opening and closing individual groups requires double clicking: one click too many. For a lot of situations, it makes sense to use the utility's other sorting tool: the...um...Sort.

Let's return to the problem of the six idling dwarves, and approach it with sorts in mind. When you click on one of the column headers at the top of a view, it will sort the content in ascending order against that column. The "Current Job" column is numerically sorted by the "Job ID", an internal list number assigned to that job. There's no particular order to how Job IDs are assigned because there's not truly superior logical way to organize them, but what's important to know is that idling is assigned a value of "-1". Sorting the column once will put idlers at the bottom, then sorting again will flip it around and bring them to the top - perfectly positioned for labor manipulation. Dwarves on break are given a value -2 and, conveniently enough, are also displayed - on top of idle ones.

Sorting by happiness will list your dwarves by their numerical happiness level, grading your dwarves down from estatic green to suicidal red - a far superior solution to the blunt and categorical Group equivalent.

The Profession column works in a similar manner to the Current Job column, sorting by another abstract numerical system, this one known as the "Profession ID" list. This (and the many variety of professional icons) makes it much harder to recognize sorted professions from one another than grouped ones, so grouping is the clear winner when it comes to this particular task. The professional listing does have one useful function, though, in that it more immediately lists peasants at the top (or bottom) of the list. These ID lists are not immutable entities, and can in fact be edited: for information on how, see the section **"Modifying Game Data"**, in the appendix.

Hovering over individual professions tells you how many dwarves have that labor enabled, and then clicking on it will sort your dwarves by their experience and skill in that category. This is much quicker and cleaner than performing the same operation with groups:

Figure 9: Anyone can pick plants, but only true farmers can plant seeds.

The ability to seamlessly list your dwarves by their competence at a task is one Dwarf Therapist's key sources of utility, and has many obvious applications when you need to find dwarves for a task you have in mind - like, say when building a marksdwarf squadron:

You can go even further with **"Filter Scripts"** and **"Roles"**, but those are advanced topics that we'll leave for later. If you right click on a column you can change the sort method - this is a more advanced role-based capacity, and will be discussed in greater detail in the section **"Using Roles"**.

5.4 Assigning Nicknames

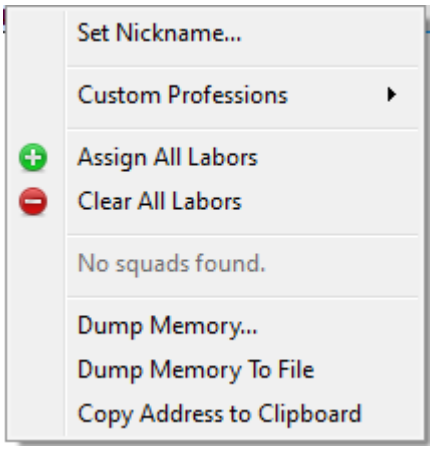
So you've spent a good ten minutes shifting around labors, building facilities, and generally getting your latest migratory wave to work at the various things that need work in your fortress. You've bootstrapped a metalworking industry, started making some potash, and are now weaving clothes. Feeling content with yourself, and maybe just a little tired because it's somehow two in the morning and you've been sitting here for five hours now, you save and log off so you can go get some sleep. In the morning you wake up, pour yourself some cereal, check the time (still Sunday, thank god), stretch your arms, and go right back to playing Dwarf Fortress. You open up Dwarf Fortress and...argh! What's this! Why are all these silly peasants making crappy armor while your legendary armorsmith is idling! Who told that farmer he could take over the cook's job! And most importantly, why is there still stone lying around everywhere!!!

So perhaps playing Dwarf Fortress until two in the morning isn't good for your fortress (never mind your sleep cycle). But there's another way to stay on top of the roles your dwarves are supposed to be playing, one with which you can be relatively sure that, if you come back in a week's time instead of a day's and have forgotten all of the various itty-bitty configuration details that the fortress survives on, you'll be able to (more) easily pick them up again and keep right on playing. The solution is to name your dwarves.

Dwarven nicknames are often used rather jestingly by players, since you can name *any* dwarf pretty much *anything*, even calling your King "Giant Poo Poo Head" and your Great Potash Maker "Unfortunate Accident". Dwarves don't know the difference and thus don't mind, but it's worth a cheap laugh from the player if "Giant Poo Poo Head" goes to clean up the blood stain left by the demise of "Unfortunate Accident". However, they can actually be a pretty powerful tool if used right. By nicknaming your dwarves by what their profession within your fortress is or will be, you'll be able to more easily keep track of who they are when you bump into them in the labor manager or on-screen. Their professional name might be "Woodworker", but to you they're "Furnace Operator", and using nicknames allows you to keep track of that while their professional name catches up to their new role.

Let's look at one such fella for which a naming would be useful, a certain Medtob Swinwind.

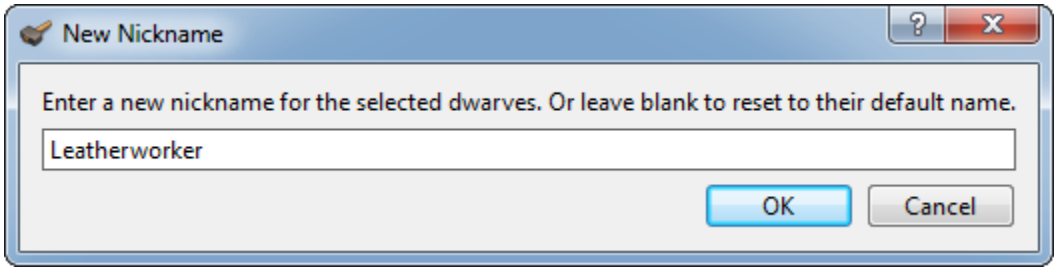
Mebtob has a few assorted skills - he's an adequate Tanner and Fish Cleaner, and a Novice Fisherdwarf and Fish Dissector. However, none of these skills are really useful to me: I've got enough fisherdwarves and fish cleaners already,[17] tanning is only useful every once in a while, and fish dissection is a near completely worthless skill. What I *do* need, however, are some leatherworkers - I just bought a shipment from a caravan, and want to turn it into a complete set of backpacks, waterskins, and quivers for my military to peruse. Unfortunately the random number god has not blessed me with any professional leatherworks - which is fortunate for Mebtob, since he's now going to move past peasanthood to become the fortress leatherworker.



For now, though, he's still a regular old fish cleaner, and if you bump into him in the hallway or look at him in Dwarf Therapist after a week away or a really long night, you won't have a clue what he's there for. So to make that job easier, let's give him a nickname. Right click on Mebtob to bring up his personalization menu, displayed at right. We haven't discussed this item yet: note the Assign All Labors and Clear All Labors commands available here, which are useful for, say, clearing off all of Mebtob's now non-essential labors before assigning him leatherworking. Custom Professions are

[17] Because of a bug, fish stocks do not replenish, which means that they will inevitably go bust, leaving your fishery workers with nothing to do.

covered in **"Creating Custom Professions"**, and the memory dump stuff is in **"Addendum: Scanning Memory"**; for now we want "Set Nickname":



New nicks are actually treated the same way as labor changes: once you've chosen a new nametag for your dwarf to go by and clicked on "OK", the name change will be added to the Pending Labor Changes dock, awaiting committal:

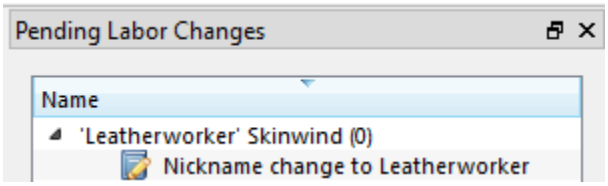


Figure 10: Ah yes, compatriot "Leatherworker Skinwind".

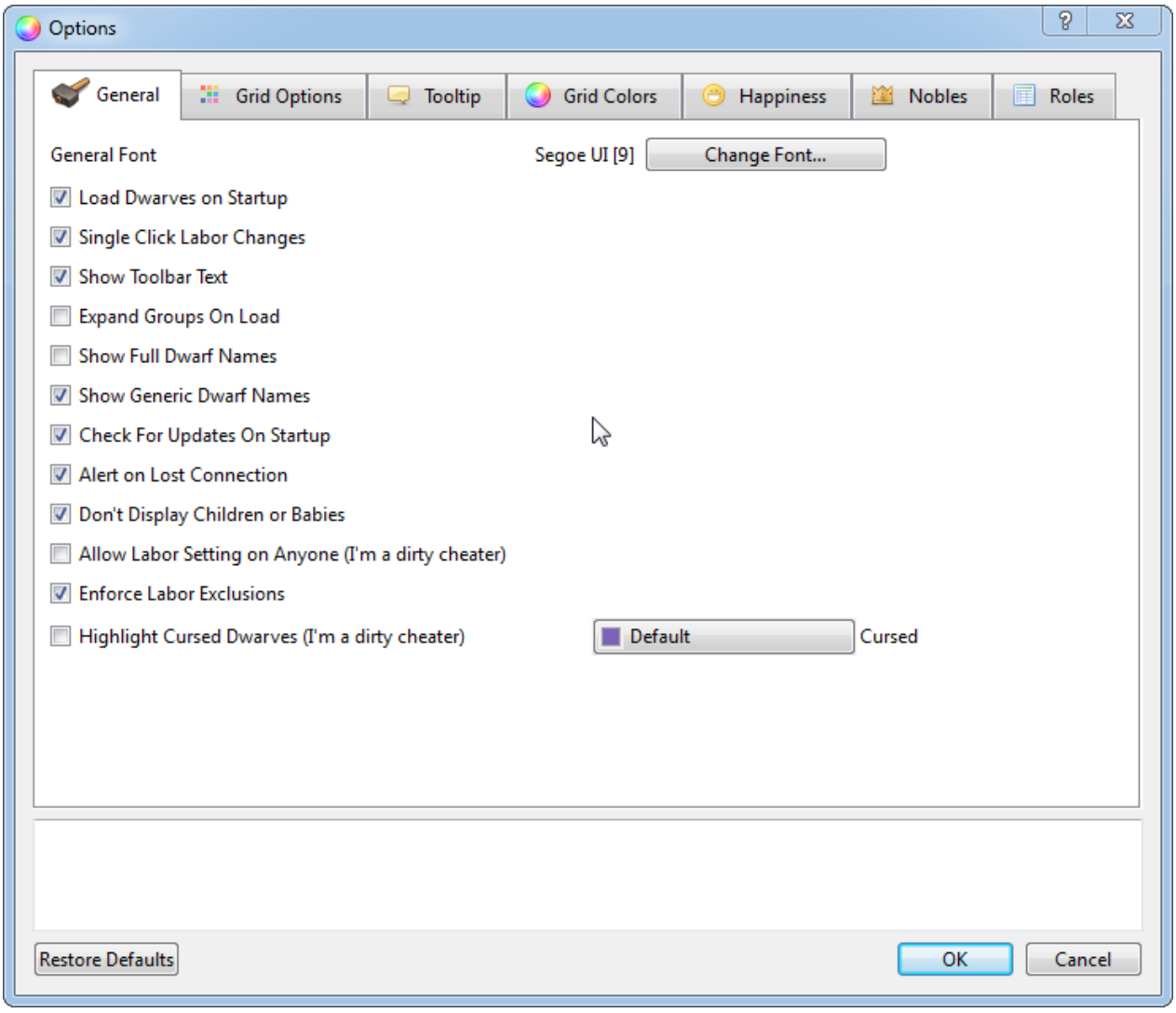
There's one more thing worth mentioning: you can assign dwarves to squads through the right click menu! No more going through the military screen trying to figure out who's who and who's available. That's it for basic labor management. We're going to go over many more advanced features in the **"Advanced Features"** section of this guide, but for now let's move on.

5.5 Addendum: Managing your Animals

There's one more thing to be said before we move on. While most of the views in Dwarf Therapist besides the labors view are non-interactive, one is - the Animals view - and so it's worth mentioning here. The Animals view provides information on the name (duh) and owner of the creature, their "profession" (really whether they are children or adults), their training level (nothing is displayed if they are tame), whether or not they are designated for or can be butchered, whether or not they are in a cage, and their physical attributes. I say that this menu is interactive, and it is: non-wild, non-pet creatures can be designated for butchering here. This makes a partial replacement for the Animal Status screen in-game - although you cannot (yet) designate creatures for taming through it.

6 Options

The Options menu is available from a few different places: either a button on the main taskbar, through the "File" menu in the main menu, or with the `Ctrl + P` hotkey. This brings up a menu:



The options menu is extremely well-documented: if you're ever at a loss for what a particular checkbox or radio button does, hover over it and look for its description in the text box at the bottom of the dialogue. It's all quite intuitive and well-explained, so instead of wasting a ton of time and space describing it, I invite you to explore the menu, and then come back to here.

Good? Ok, let's leverage our options to improve how Dwarf Therapist looks and runs a bit.

6.1 Formatting Your Display

A big problem I have with the default configuration of the current version of Dwarf Therapist is how unnecessarily complicated and bulky it is. Dwarf Therapist is a *utility*, and thus its beauty is in its usability - not the fancy and non-functional faded-out boxes and headers and dividers that eat up tons of screen space. So in this section we're going to leverage the program's high degree of customizability to simplify its appearance and increase its information density; if you like how it looks as it is, skip ahead to the next part of this guide, **"Advanced Features"**.

Figure 11: My horizontal scroll bar at the beginning of this exercise.

Remove Spacers

To remove the perfectly non-functional spacers from the long-form Labor view, where they matter, peruse the **Views**. Drop in the "Labor NO SPACERS" view from the menu and then delete the old one. Of course having "NO SPACERS" stare at us is quite annoying; we'll look at ways to fix this later in this guide.

Figure 12: Getting better...

Turn off Gradient Shading on Headers

To remove the gradient shading in the column headers and return them to plain coloration, uncheck "Gradient Shade Column Headers" in `Options > Grid Options`.

Figure 13: Much more readable.

Turn off Gradient Shading on Cells

To remove the gradient shading present on table cells that are toggled on, uncheck "Gradient Shade Cells" in `Options > Grid Options`.

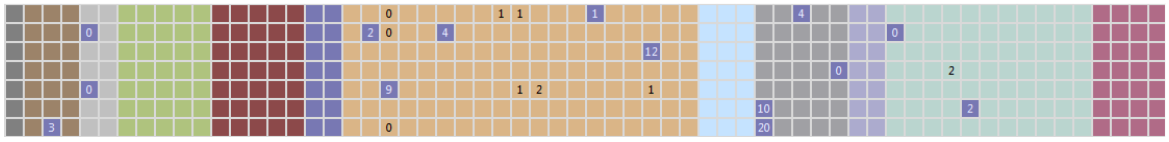


Figure 14: Easier on the eyes.

Synchronize Scrolling between Views

To enable this feature, check "Synchronize View Scroll Positions" in `Options > Grid Options`. This adds some functionality to your views, preserving the current scroll position between them and letting you examine a single dwarf (or if they're sorted the same way, a group of dwarves) through the lens of multiple different views. They don't have to be sorted the same way for this to work, though occasionally it near-misses (bringing you to the dwarf just above the one you care about in the view, for instance).

Highlight Highest Moodable Skill

To enable this feature, check "Highlight moodable cells in labor/skill columns." in `Options > Grid Options`. This one's optional: it gives you extra information, but knowing what moods your dwarves are likely to have isn't terribly useful since the selection is random, and you can get the information at a glance through a Group. Though it doesn't say it, this option also highlights dwarves that are legendary because of a mood with a different colored box. You can change the colors in `Options > Grid Colors`.

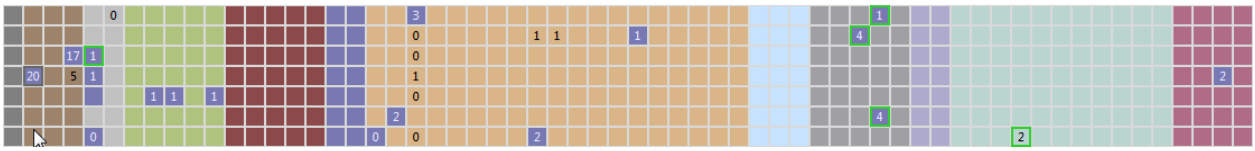


Figure 15: Moodable in green, already mooded in brown.

Show Highest Moodable Skill in Tooltip

To enable this feature, check "Show Highest Moodable Skill in Tooltip" in `Options > Tooltip`. This will add a "Highest Moodable Skill" entry to the dwarf tooltip - no real reason not to have it.

Make the Main Toolbar More Compact

To shrink the main toolbar a bit so that it doesn't take up so much vertical room (and thus give your view and your docks more of it), uncheck the "Show Toolbar Text" option in `Options > General`.

Highlight Nobles

To turn this feature on, hit "Highlight Nobles" in `Options > Nobles`. The default color is orange, and you can further pick specific colors for specific roles. Personally I set chief medical dwarves to dark red, bookeepers, managers, mayors, and brokers to magenta, royals to dark magenta, and military types (militia leaders, champions) to dark green.

Role Information in Labor Columns

To turn this feature on, hit "Role Information in Labor Columns" in `Options >Roles`. We've barely talked about roles so far, and they're covered in-depth in the section on them, thoughtfully titled **"Roles"**. For now it's just more information for you to peruse.

Move, or Remove, your Docks

Docks are a very useful thing, displaying useful information in a compact form. However, if you want to make the most use of your space: go without. Removing all docks from the right side of the screen so that the edge of the view window is now plum against the right edge should be enough to let you see the entirety of the grid on a moderately large screen.

Figure 16: On my (21-inch) monitor, it was enough to get rid of the scroll bar completely.

If you want to get rid of the scroll bar but want to keep your docks, too, another option is to place them at the bottom. This is a dock-heavy option, however, because the docks are not optimized to use horizontal space well - so giving the Pending Labor Changes the entire lower fourth of the screen is no better than giving it the same amount of space, vertically, on the sidebar.

Remove the Main Toolbar

The main toolbar is very skinny: if, like me, you have space on the right edge (or can spare a sliver off the top), you can simply and safely stick it there. But if you want to reclaim that last ounce of screen space, by all means, remove it: all of its commands are hotkeyed anyway.

Play with Fonts, Spaces, and Font Sizes

If this still isn't enough to get rid of the horizontal scroll bar, and you really, really want to get rid of it, you can play around with the fonts, font sizes, spacer options, and grid sizes in `Options > Grid Options`, and lower the sizes of things until the bar is no more. Obviously, the smaller the font and grid size, the harder it is to read information of the utility - even reducing it to 15px has a noticeable effect.

Here is what my display looked like at the end of this exercise:

Cleaner, more informative, and most importantly: no horizontal scroll bar!

Part III
Advanced Features

In this section of the guide we're going to really dig into Dwarf Therapist, working with powerful elements of the program that aren't as immediately apparent as toggling labors on and off, giving stuff nicknames, and other such simple and obvious applications.

7 Roles

7.1 What's in a Role?

So far we've mentioned "roles" and the "role" they play in Dwarf Therapist only in passing. Since we're now going to discuss them in more detail, the first question we have to ask is, what's in a role? Have you ever had two miners work side-by-side from the very beginning of the game, but discovered that one reaches legendary status before the other? This happens when one dwarf is better adapted to that role than his fellow, and the differences between them become increasingly obvious over time. "Roles" is Dwarf Therapist-speak for the holistic weighing of the various elements of job performance, of which skill is only the most immediately visible and obvious element. Thus in order to understand what roles are, we must first get comfortable with what these elements are, and where they come from.

Attributes are, subjectively speaking, the most important hidden role modifier. No two dwarves are alike, and all dwarves have certain attributes that are attached to them from birth.[18] Where one dwarf might be naturally weak have superb spatial sense, another might lack analytical ability and be susceptible to disease, but be unnaturally strong. These attributes themselves fall into two categories, physical and mental; they can be assesed under the "Attributes" view, and they're described in detail on the DFWiki: `http://dwarffortresswiki.org/index.php/DF2012:Attributes`. Since they're pretty important for understanding roles, here's a quick list of the ones that affect job performance:[19]

Physical Attributes

- **Strength**: Alters the damage done in melee (increases velocity of weapon swings), increases muscle mass (thicker muscle layer also resists damage more), and increases how much a creature can carry. Higher strength also increases the speed with which a creature, even a naked creature, may move. Movement speed is important for pretty much every task, but to a varying degree.
- **Agility:** This attribute increases the speed at which a creature works in the same way as strength – a creature with maximum agility and strength can move around three times faster than a creature with minimum agility and strength.
- **Endurance:** Reduces the rate at which dwarves become exhausted, important for physically demanding tasks.
- **Toughness:** Reduces physical damage. Used by physically demanding tasks.

Mental Attributes

- **Analytical Ability**
- **Focus**

[18] Attributes, alongside appearance modifiers, are inherited through dwarven genetics.

[19] It should be noted that all creatures, not just dwarves, have attributes (for instance, the attributes of animals attached to your fortress are visible on the "Animals" view). The attributes of creatures that are not tied to your fortress - wildlife, unwelcome visitors, hostile sapient creatures, caravan traders and guards - will be hidden from you.

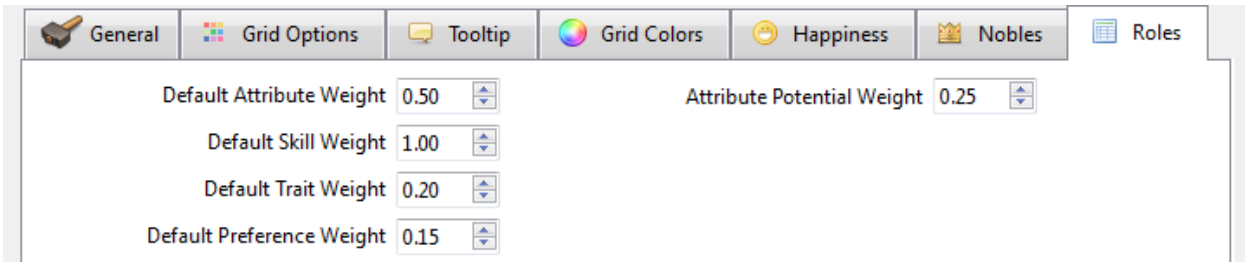
- **Willpower:** Willpower directly reduces exertion and pain effects, useful for physically demanding tasks.
- **Creativity**
- **Intuition**
- **Patience**
- **Memory**
- **Linguistic Ability**
- **Spatial Sense**
- **Kinesthetic Sense:** Most skills involving any movement at all (lots of them), and many non-skilled tasks as well are affected by Kinesthetic Sense.
- **Social Awareness**

Physical attributes will increase or decrease over time, depending on whether a dwarf uses or doesn't use them in their day to day tasks. Thus your Miner will become very tough and very strong while working on the job, and would make a good recruit for your military in their next, ahem, role.

Another modifier is a dwarf's **personality traits**. For the most part personality traits only affect social skills and appointed jobs that involve working with others - expeditions leaders, managers, brokers, and mayors use them, as do high nobles. There is one, perseverance, that is thought to affect the length of breaks that your dwarves will take, and might thus be important for all skills - your dwarves will obviously do less work overall if they're always on break. However, Toady has never confirmed this.

The final hidden element of a dwarf's role is his or her **preferences**. Dwarves have innate preferences for certain items, materials, organisms, and even colors and shapes (I want my coffin to be a *yellow square*, you hear me!). Dwarves like seeing things they like - they'll get a happy thought from it - and they like working with them even more, producing above-average quality goods when working on items or with materials that they like. For this reason a dwarf that likes beds will make a better carpenter than a dwarf that like breastplates and vice versa. Having the right sort of preferences can be an important bonus in a dwarf's work, meriting weight in the role calculations.

Now that we've examined all of the elements besides skill that are weighed into roles, the next logical question is: how are role numbers calculated? As it turns out, this is not a trivial question. There are, essentially, three ways of "thinking about" - and hence using - roles. Let's first examine th default behavior, which can be seen and modified in the "Roles" Options menu (`Options > Roles`):



Dwarf Therapist takes a dwarf's attributes, skills, traits, and preferences and weighs the ones that matter for a certain labor by the amounts inputted here. It then performs a **cumulative distribution function** on the results, giving you how fit that dwarf is for that role as a percentage calculated against the *current dwarves in your fortress*. The result of this statistical trickery is that while the result is a *current* rating, it's *non-transferable* (a 75% rating in one skill is not directly comparable to a 75% rating in another) and *non-preservable* (role ratings will resettle every migrant wave, or every time a child grows up). It's also razor

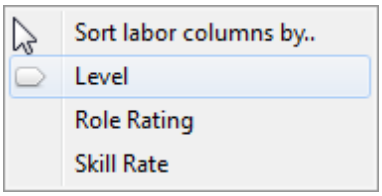
sharp: since perhaps nine out of ten average dwarves are unskilled in a particular labor, and since the CDF is based on probability density, a dwarf that happens to be even a novice at a labor will shoot up to 90+ percent compatibility, and reaching proficient rank is usually enough for any dwarf to hit 100 percent. For the purposes of this guide, I call this rating **skill rank**.

Changing the second option in `Options > Roles`, "Default Skill Weight", to 0 removes it from the equation. With skills no longer counting for anything, the role rating now shows what I refer to as **basal compatibility** - the results of a CDF crunch of your dwarves' attributes, traits, and preferences alone. This is a useful view of your fortress "in retrospect", as you might discover that the weaponsmith you appointed from scratch in wave one is completely outclassed by several dwarves in waves two, three, and four. Of course by that point that dwarf would have put on so much distance in terms of skill that it doesn't matter anymore - but it's still an interesting statistic to know. It's also much less statistically sharp than skill rank is, because base stats don't vary nearly as much as skill does.

The last and most useful function that roles serve is one that answers the question "Which of these dwarves will reach legendary status the quickest?" In such a distribution an unskilled dwarf with minimum stats will be rated at zero percent, an accomplished one with middling stats would be a fifty, and that legendary +5 dwarf with maximum stats and perfect preferences and whatnot would be a one hundred: a **true role rating**. So, how do you turn this wonderful ranking on?

Well... you can't. It's a pipe dream. There are simply too many variables involved in job completion length and skills gain and no one's ever fully done anything beyond rudimentary research on the topic - it's very complicated and not nearly as glamerous as combat weapon penetration rates. Even the weights that the program uses are the result of conjecture - there's just no hard !!SCIENCE!! to base the whole thing on as of now, and that's the reason that we have to use skill rank, drawbacks that it has, as a substitute.

7.2 Using Roles



We've talked before about the utility of clicking on the labor headers and having your dwarves sorted in ascending order of experience. But did you know you can actually change this behavior? Indeed, right-clicking on anywhere on the labor headers will bring up a menu that allows you to change the current labor sorting behavior, as seen on the right.[20]

The first option, Level, is the default option, and that makes sense. The third option is pretty much the same as the first, except that skill rate (which the amount of skill gained every time a labor in that category is completed) is modified by, you guessed it, the dwarf's role, and so will vary a little from straight up experience level - but experience is still the most important indicator of relative skill gain, so it won't change too much. However, the one we want right now is Role Rating. Click on it. Now, whenever you sort a column, the dwarves will be listed by their adeptness for that labor. Assuming you turned the role tooltip display on, as we did in **"Formatting Your Display"**, then you can hover over a labor box to see the exactness of the match - if you skipped that section, you can toggle the behavior on now by going to `Options > Roles > Show role information in labor columns`.

If you play around with sorting the columns now, you will notice two things. The first is that dwarves with the labor enabled and those with it disabled are considered, and listed, separately from one another: under the ascending sort, dwarves active with that job come first, followed by those that are not. The second is that in terms of basal compatibility skill is (mostly) irrelevant; beyond the fact that we took it out of the calculations in the previous section, this indicates that physical attribute gain from performing labors that use that attribute is simply not that high. And so it may be that your primary mason is buried nearer to the bottom of the list, with a compatibility of 14%, with a long list of mostly dabbling dwarves that outrank him:

[20] You can also change the sorting behavior of the dwarf column, to sort by ID (a somewhat useful alternative to grouping by migrant wave) and age (useless). Unfortunately, it will not save this behavior, and attempting to reverse the sort will send it right back into alphabetical order.

Figure 17: Lucky number 15? Well, maybe not.

In this case the reason that the dwarf that became my chief mason was so incompatible with his job is that he was part of my starting seven, and so his attributes were random - and the random number god did not favor me in this case. It's important, however, to realize when basal compatibility is important and when they are not. While basal compatibility will tell you which of two comparable dwarves would be better at a certain job, they are no replacement for hard-won experience, and so that dwarf, incompatible though he may be, is many times better than any of the laymen that pretend to also do his job. Once you put three or four levels of experience points between two dwarves, their roles lose all relevance.

So then, when are roles important? As it so happens, my fortress just experienced a migratory wave, and something that my fortress critically lacks right now is metalworkers. I'm a big believer in expensive, well-decorated furniture for my dwarves, and I've got a skilled gem setter running around encrusting beds in locally-mined sapphire, but he's not keeping up with the demand. I have a lot of extra fuel, some gold bars, and a couple of metalsmith's forges idling around, so I decide that the solution to my problem is to put two of these new migrants to work as dedicated metalcrafters, studding every piece of jewelry they can get their hands on in gold. Since no one has a skill advantage in this arena (my fortress has zero metalcrafters), and all but two of my dwarves are pretty much useless anyway, I want to pick out the best two dwarves available from this wave. How do I go about doing it? By combining two things we've learned so far, grouping and sorting, with a third - roles.

Try doing this exercise yourself - you need not even have a fresh migrant wave. First, group your dwarves by migration wave, collapse them, and open the latest wave. Now, with the sorting method set appropriately, hit "Metalcrafting" to sort these dwarves by role score. Now look down your list.

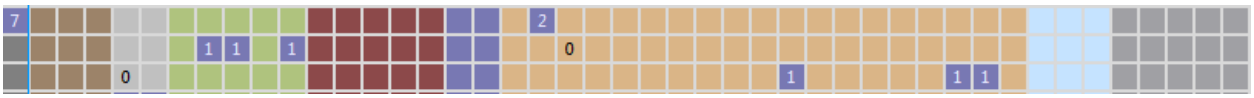


Figure 18: This migrant wave is *almost* fresh. For extra clarity, unnecessary details have been omitted.

In my case, the first dwarf on the list was an Adept miner with a score of 98.1 percent; but mining is a useful skill and I'd rather he cut rock for the fortress, so I keep going down. The next option is better: a ranger whose only real usable skill is hunting, and a farmer with some assorted non-essential skills, both of whom are easily replaceable. I right click on their names and hit "Clear all Labors" to quickly wipe their workloads, designate metalcrafting for the pair, and then hit "Commit" to send the changes to the game. I even give them nicknames, to be extra sure I won't forget about them:

Figure 19: Notice the change is sorting behavior to reflect changes in labor designation.

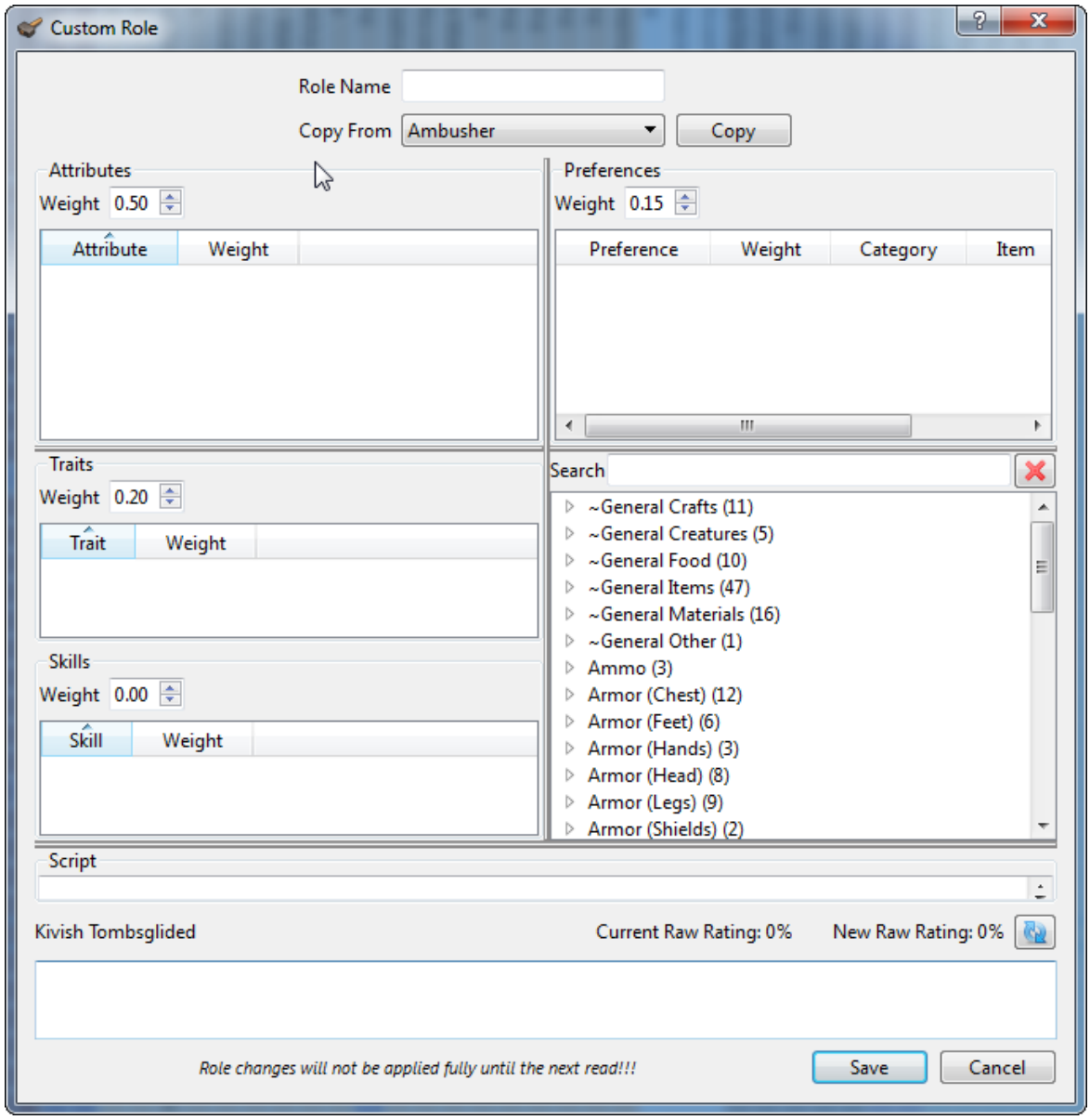
Voila! I have now committed two well-suited dwarves to their new tasks. Works flawlessly, right? Well, *almost*. There are other, similarly replaceable dwarves also in the fortress that would make better candidates for this job, but hunting them down is a pain. I could find the best-suited dwarves in the *fortress* by turning off grouping and then sorting the column, but then I'd bump into another problem: which dwarves are replaceable, and which ones am I training up or holding onto with certain roles in mind? In this respect, your tools are crude - *maybe* you remember exactly or can deduce from the labor settings what role each dwarf is set for, but in a fortress of 200, you probably don't. In large fortresses with many industries, it's simply not possible to easily keep track of so many variables at once.

We can solve this problem by using nicknames aggressively, designating all dwarves committed to a task with a nick, and then using the "Has Nickname" grouping in the view. Setting so many nicknames is tedious, however, and if you don't like using them then you're out of luck. Or are you? There is another,

more powerful solution that will come later in this guide: filter scripts. For more details on this, see **"Filter Scripts"** for the scoop.

7.3 Creating Custom Roles

If you right click on "Roles" on the menu bar it will bring up a menu with several role-modification related entries on it. Dwarf Therapist allows the creation of custom roles through this dialogue, and you can also export and import other player's custom roles definitions. If you don't like a default role, you can overwrite it by building and saving a custom role with the same name.[21] Click on "New Custom Role" and it should bring up this dialogue:



[21] You can also do this by exiting Dwarf Therapist and browsing to and modifying the `game_data.ini` in the program's install directory: see **"Modifying Game Data"** for more information. A word of warning: your changes will not be preserved should you update Dwarf Therapist.

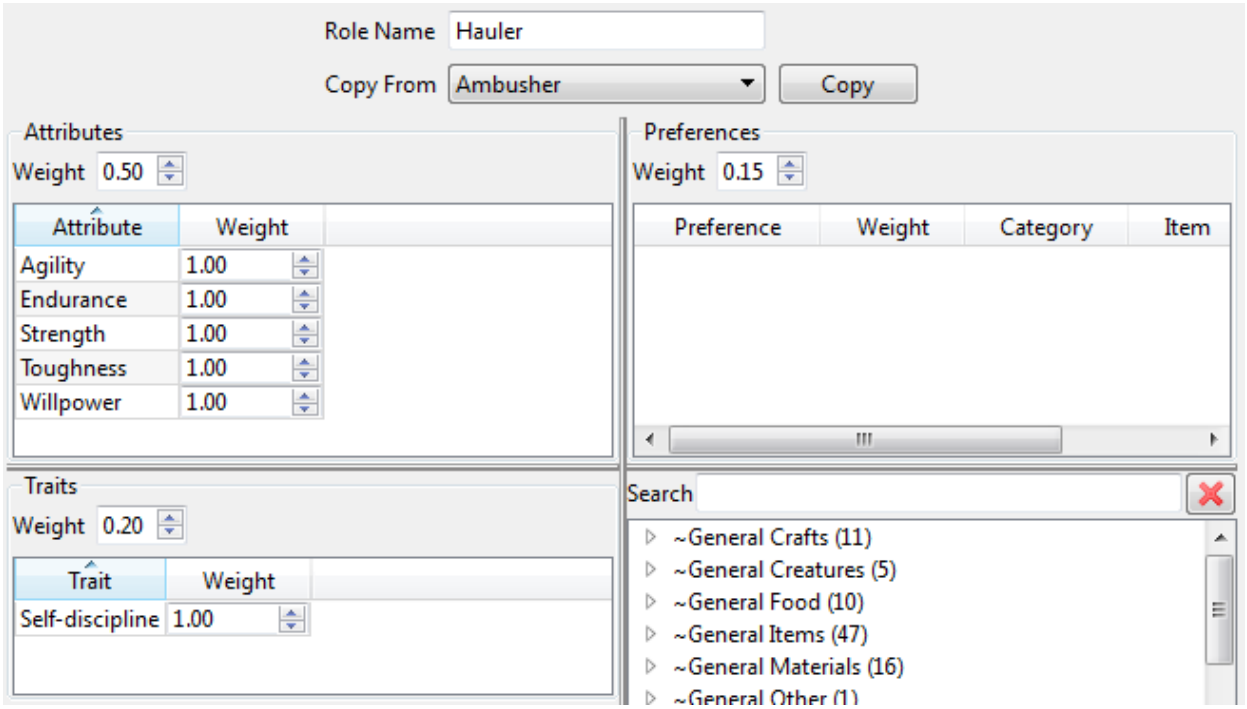
This is the editing interface for creating a custom role, and clearly demarkates the four components of a role. There is a scripting option available - this is a plug-in for **"Filter Scripts"** that we'll talk about later. And finally, the italic text at the bottom of the window tells us that roles changes will not be read in until you reread your dwarves, so we will have to do so after we're done here.

Dwarf Therapist ships with a pretty comprehensive role library - it even has lawdwarf (weak) and liar options, and provides role information for jobs that are not implemented into the game, such as alchemy. However, there is one feature that it lacks, and that is a role setting for haulers. Ye peasants do their job best when they're quick and strong, but there's no role for such a thing available! The solution, of course, is to write our own.

Start by copying role settings from a similarly physically demanding job - say, mining. Scroll down to mining in the"Copy From" menu (or hit M while the menu is active to get there more quickly) and click "Copy". This will copy all of the role settings for minters into the dialogue. Our haulers need not be lovers of picks, so let's remove it from the Preferences screen - right click on it and hit "Remove Selected". Skills are irrelevant, too (especially for an unskilled hauler), so remove that as well. Actually, for a hauler that's going to be carrying pretty much everything in the fortress back and forth, preferences are unimportant - they'll bump into pretty much everything - so let's drop stone, too.

Hauling is an unskilled job, so the only things that matter for it is the speed and endurance of the dwarf. That means that the Strength, Endurance, Willpower, Agility, and Toughness attributes are at a premium, and the rest can be gotten rid of. The only Trait that matters to use is self-discipline - don't want them taking two-season breaks, after all - so let's add that into the mix, too. And, of course, let's give your new role a name: "Hauler" seems appropriate.

This is what the important settings in the interface should look like at the end of our little exercise:



Hit "Save" to save your new custom role. Now, whenever you go to the Custom Roles dialogue box, hovering over "Edit Custom Role" or "Delete Custom Role" will give you the option to perform that action on your new role. Custom roles can be used in custom views and in labor optimization - we'll see an application for them later - but cannot yet be bound to custom professions or to labors.

7.4 Exporting and Importing Custom Roles

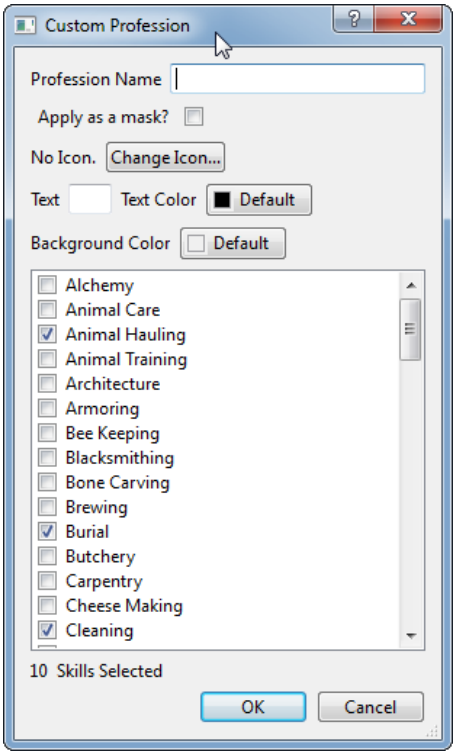
It's also possible to export custom roles, which creates a Dwarf Therapist export file (`.dtp`) that can saved and then passed along to another installation of the utility by importing it. You can export your custom roles and then upload them on the web (preferably the Dwarf Fortress File Depot: `http://www.dffd.wimbli.com/`) to share them with other players, or you can download one from the web and then import them into Dwarf Therapist so that you can use them too. By the end of this guide we're going to generate a lot of custom bits of code for the program to use, including a couple of custom roles, but for now let's just ignore the importing and exporting options, and talk about something else: custom professions.

8 Custom Professions

Dwarf Fortress has a *lot* of professions, and even more skills, available to your dwarves. In fact, learning what each profession and skill corresponds too is one of the hardest part of learning to play the game, because there's just so many of them. You can add to this sprawl by creating your own professions, assigning multiple labors to one unified profession. This ability has its uses, and in this section we're going to use it to unify hauling labors.

Whether your dwarf is hauling around sacks or focused on their primary task is usually an all-or-nothing thing: either they're committed to their job, or they're not. It's not very useful to have dwarves only perform *certain* hauling tasks, unless you're getting into the extreme end of fortress management and optimization, so most players of the game have them either all on or all off on a dwarf-by-dwarf basis. We're also going to assign the "Hauler" role that we built in the previous section, **"Creating Custom Roles"**, to this new, unified profession.

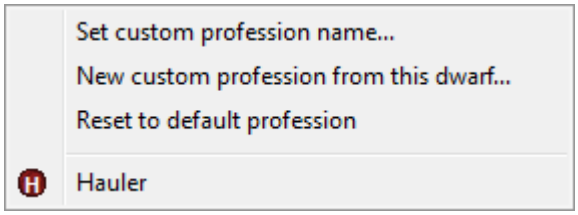
8.1 Creating Custom Professions



The dialogue for creating custom professions is located in one of two places, and you can get there either by hitting "Create Custom Profession" at the bottom of the Custom Professions dock, or by right clicking on a dwarf, going to "Custom Professions", and hitting "New custom profession from this dwarf". This second method starts you off with the labors that that dwarf had enabled checked, and so is faster if you have a dwarf already set to the labors you want - and after a quick profession sort I discover that I do, in fact, have a peasant hanging around my fortress, so I open it up off of him. Lo and behold, the ten hauling "subskills" are already selected. I name the profession "Hauler" and give it a red circle icon, one of the spare ones in the utility's default set that's not already used by a professions. To make it clearer that this icon is for haulers, I'm going to use the text and text coloration option provided in the window to draw a white H over my icon. You can also add a background color, but the regular icon set keeps a transparent white background, so I will too. I hit ok and the profession is now listed under "Custom Professions" in the Custom Professions and Icons dock (which is the only way to edit or delete a custom profession, fyi). To make it appear in the display, you have to commit your changes and reload your dwarves.

To work the way we intended, make sure you don't check the Mask option in the window - this will change the behavior so that the custom profession only masks the dwarf's ordinary profession, and it will not allow you to make designation changes. You can

remove a dwarf's custom profession designation with "Reset to default profession" and then a commit to the game.



You can create a mask more directly by right clicking on the dwarf and then going to "Set custom profession name" under the custom professions menu, and then inputting a name: once committed this will change the dwarf's profession to a custom one. This is a by-dwarf operation that basically mirrors how custom professions work within the game, replacing the dwarf's professional name with the new string but not changing anything else, without labor assignments or a custom icon, and with no facilities for transferring the profession to other dwarves. Thus it's like a nickname applied to the dwarf's profession, and in fact provides the same "who's this dwarf?" functionality that we previously used to make nicknames useful. However, since nicknames can be more easily seen at a glance within the Dwarf Therapist utility (and don't get in the way of this feature's more advanced version), I recommend sticking with them for this function.

For now, the functionality of custom professions is fairly limited. What you've created is essentially a triggerable mask. Although the profession icon has changed and although the dwarf will be listed under their new profession within Dwarf Therapist, actually triggering the profession is annoying: you have to right click on the dwarf, then click on "Custom Professions", then click on the profession as it appears in the menu. This will toggle the labors associated with that profession on, and turn off all others in the process. You have to do so much work that this shortcut becomes anything but!

8.2 Exporting and Importing Professions

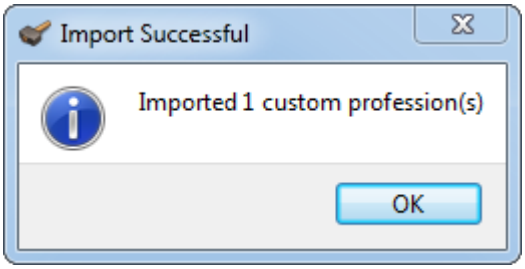


Figure 20: Binmaker extrordinare.

Custom professions is actually one of the two Dwarf Therapist features we discuss (alongside nicknames) that are present within the base game, although custom professions have less utility within the game - they simply replace the dwarf's professional name. If you want to assign the same profession to multiple dwarves you have to do so manually, retyping it in each time. If you assign a custom profession to a dwarf within the game, but don't import it, his professional name will change in Dwarf Therapist but nothing else will (essentially the same as setting a mask). To make the profession more available to Dwarf Therapist, allowing you to more easily assign it to multiple dwarves, give it an icon, and use it as a labor designation shortcut, you have to import it. To do so, hit `File > Import Professions from DF` - it'll give you a quick confirmation window telling you how many professions you've imported.

Much as with roles, custom professions can be made within the utility and exported, or taken from someone else and imported - this is actually going to be a common theme between the various customizable functions we're discussing in this section of the guide. There are even dialogues that let you pick and choose which professions you want to export or import. Both options are available from the `File` menu.

9 Custom Grid Views

So far we've stuck with the views that the utility generates for us - labor, military, etcetera - and the most interactive thing we've done with them is switch our labor screen to the beautifully structured, unfortunately named NO SPACERS version (that is, if you've read **"Formatting Your Display"**). However, it's also possible to define your *own* grid views - here we'll use it to make our own, better, labor management view.

9.1 Creating Your Own Grid Views

Unfortunately, unlike the rest of the advanced utilities we're discussing in this section of the guide custom grid views can only be created through its associated dock, so if you don't have it open already you're going to have to do so at least for the duration of your editing.

Inside the dock you will find the grid views that the utility comes preloaded with, but they will be grayed out. You can create a new, editable grid view in one of two ways: either by hitting "Add New Grid View", which gives you a blank, or by right clicking on and copying one of the views that is already hardcoded in, which will generate you an editable copy of the view. Copy the Labors NO SPACERS view now, then right click on it and hit "Edit" (you can also copy it again or delete it entirely in this menu). This will open the Grid View dialogue:

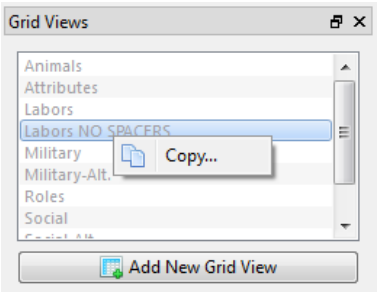
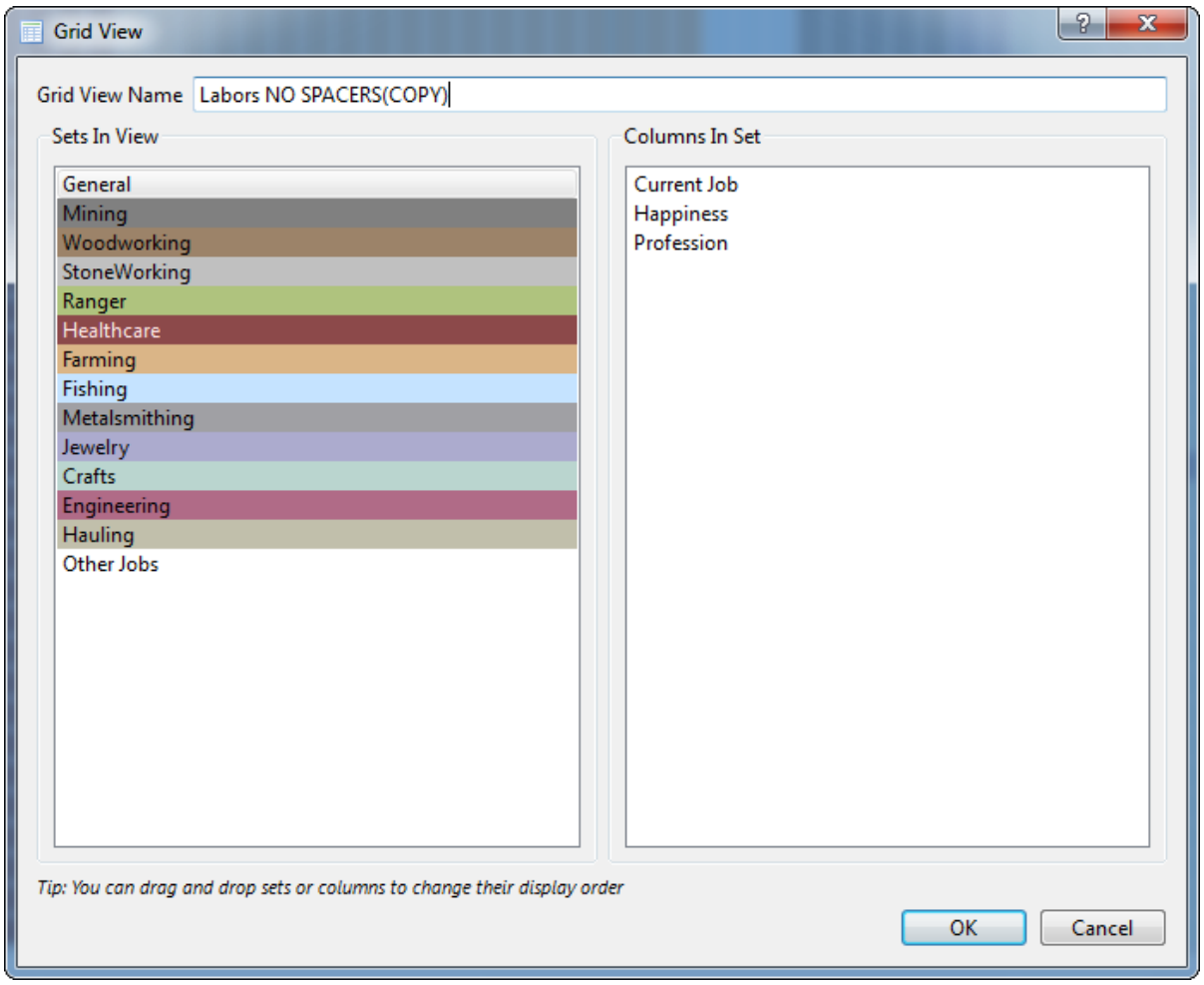
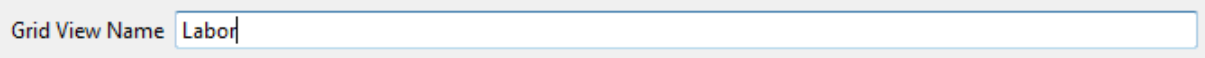


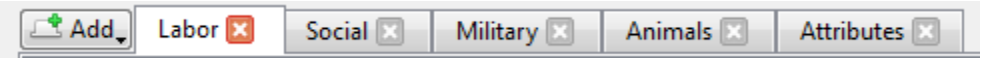
Figure 21: The Grid Views dock.



The very first thing you should do is this:



Now if you add that view from the "Add Views" drop-down menu and delete the old one:



NO SPACERS, banished!

The Grid View window is organized into two columns, sets on the left and component columns on the right. Set names are not displayed anywhere on-screen, so they're mostly organizational, but ordering your columns causes the program to automatically apply the set's chosen coloration to all columns in the set. One column in the game is immutable - your dwarves by default, or your animals if you've copied the animal view instead. This will always appear on the left edge, and does not appear in this listing.

Right clicking on a set allows you to either modify it or delete it entirely. Editing the set allows you to change its name (again, merely organizational) and its coloration - there are a number of predefined options, but by going to the `[...]` option you can choose and use any hex value.[22] Clicking on an empty space in the box creates a new, empty set for editing. To rearrange the position of a set, simply grab and tug it up or down.

The columns themselves are color-codes to match their sets, but if you edit them you'll be able to override the base set color and set your own. Right clicking anywhere will also bring up a complete menu list of possible columns to add to your view. You can also add or remove spacers here - since they don't serve too great a function, if you don't have the horizontal sceen space for them in a view, removing them might be extremely helpful.

Now, let's make some modifications to our new and improved "Labor" view.

Figure 22: Our Labor View at the beginning of this exercise.

The first thing we should do is remove alchemy. Assuming you're not using mods, it's not implemented into the game, and so it has no place in the labor view. Go to Other Jobs, right-click on Alchemy, and hit Remove.

There isn't ever a real reason to remove Cleaning as an enabled labor, either - when your dwarves actually perform that job, it's a blessing - so let's remove that too. This has the benefit of allowing you to remove all labors from your dwarf with the right click option without removing cleaning as well.

Though Architecture is held in-game to be an administrative job, despite having a labor assigned to it, it's really an engineering one. Now that it's left all on its lonesome on the right edge of our screen, we have an excuse to move it under the engineering banner. I'm placing it between mechanics and pump operating on the banner.

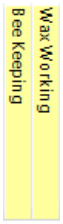
Animal care is similarly unimplemented in vanilla at the moment - your animals either heal up, or they die, and there's nothing you can do about it. So it's safe to send it the same way as alchemy, because even

[22] For some reason choosing a new color sometimes doesn't change the color of the header. Going through the menus and just pressing "OK" again fixes this bug.

if some of your dwarves do have skill in it they'll never get to use it.

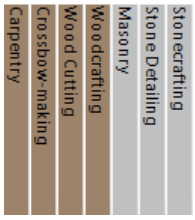
Feeding prisoners and the wounded is always a priority job that is or should be enabled on all dwarves by default, pretty much even the grumbly ones - the rare -20 happiness penalty is minor compared to the cost of a tantrum spiral. So these two columns are safe to remove, too.

Now that we've removed useless labors from the display, let's do some organization amongst the tasks that are left. Farming in particular is a mightily large category, and I often get lost looking for my Weaver somewhere between my Potash Maker and my Presser, or even start designating the wrong labors on a dwarf! I have a similar problem with the labors under the "Crafts" header. Let's divide these sections between a few different categories.



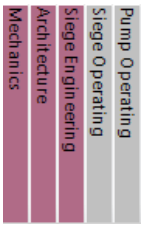
The first new category I create is beekeeping. Beekeeping is rarely used by players because it's an extremely labor-intensive and limited form of food production that produces low-value goods and is heavily bugged. Assuming the bugs with the industry are resolved, it might become more useful in future versions of the game, but for now we'll relegate it to the sideline. Beekeeping and Wax Working are both jobs that are only ever used in this industry. The new color of choice is obvious - yellow. Open up the customization menu and select the weaker yellow of the two that are available in the menu. This is still pretty bright, though, so let's weaken the colors a teeny bit further. In the end the color I used had an RGB value of 255-255-157 (alpha 255), but you can adjust it to your liking.

Woodcrafting is better off alongside its peers in the woodworking column.



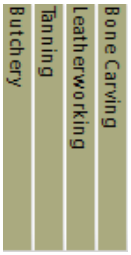
In a similar vein, stonecrafting is better alongside masonry and stone detailing.

Pump operating and siege engineering fit a certain theme, so while we're doing all this rearranging let's give them their own column to the immediate right of their current location. I use the default light grey color for this column.



Let's also move Siege Engineering to the right edge of the engineering set, to match Siege Operating opposite it.

Butchery, tanning, leatherworking, and bone carving are all components of the meat and leather industry, and so belong togethor. For this one I used a muddy leathery brown, RGB values 170 170 127.



Since there's no Other column anymore, let's set the hauling labors to their base white to make more sense of the skill-less labors.

Pottery and Glazing work as a pair. For these two I choise a light pink, RGB value 255-225-240.

Shearing, Spinning, Weaving, Clothesmaking, Dying, and Strand Extraction all have to do with the clothesmaking industry. For this one I used a downsampled version of one of the base pastel blues, RGB 155-205-255.

Glassmaking becomes the odd one out of the old crafts column, but folds nicely into jewelry - glass is just a fancy gem that can also be turned into furniture after all.

The remaining parts of the large farming cluster were a little trickier to organize. I ended up dividing it into two tranches with some color variation betwixt them (in color order, default, 218-199-175, default again, 85-170-127, 193-191-171, 85-170-127 again).

Then I toned down the medical task colors because the text is really hard to read in that dark a color. I moved this one down all the way to 140-112-114.

Now some name changes were in order. Why is it Farming (Fields)? Let's just set that to Farming. Why is it Haul/Push Vehicles? Whenever you are using a vehicle you are pushing it; when you are hauling it to and from a stockpile it falls under Item Hauling. Let's just change that to Push Vehicles. While it is true that bowyers are only really crossbow-makers, as they don't make anything else, "bowery" is a much shorter and more elegant name: let's change it to that.

Finally, glassmaking and bone carving are kind of lopped onto their sets, and siege operating folds into siege engineering, so I fade those halfway down the coloration scale to individualize them.

The finished result:

It's organized, easy to view, nicely colored, and removed of all un-necessary bits; half an hour well spent. There's a full spread on the next page. Although we didn't use the capacity here, any custom roles you create will appear under the "Roles" selection menu, and can thus be added to custom grid views you create - you can very easily add a Hauler role fitness column (we defined this custom role as an example back in the section **"Formatting Your Display"**) to your Labor view, for instance. Unfortunately one thing you *cannot* do yet is is add labor columns for custom professions - this behavior will be added in the next update.

9.2 Exporting and Importing Grid Views

The ability to export and import grid views is actually very important, and both options are available from the File header under the taskbar. In the above section I spent about half an hour organizing, modifying, and documenting the changes I made in order to create a more useful and more readable professions grid. You can spare yourself at least some of the trouble by simply downloading and importing my exported grid view (I say at least some because there's bound to be some things with it you want to tweak) from the Dwarf Fortress File Depot: `http://dffd.wimbli.com/file.php?id=7880`.

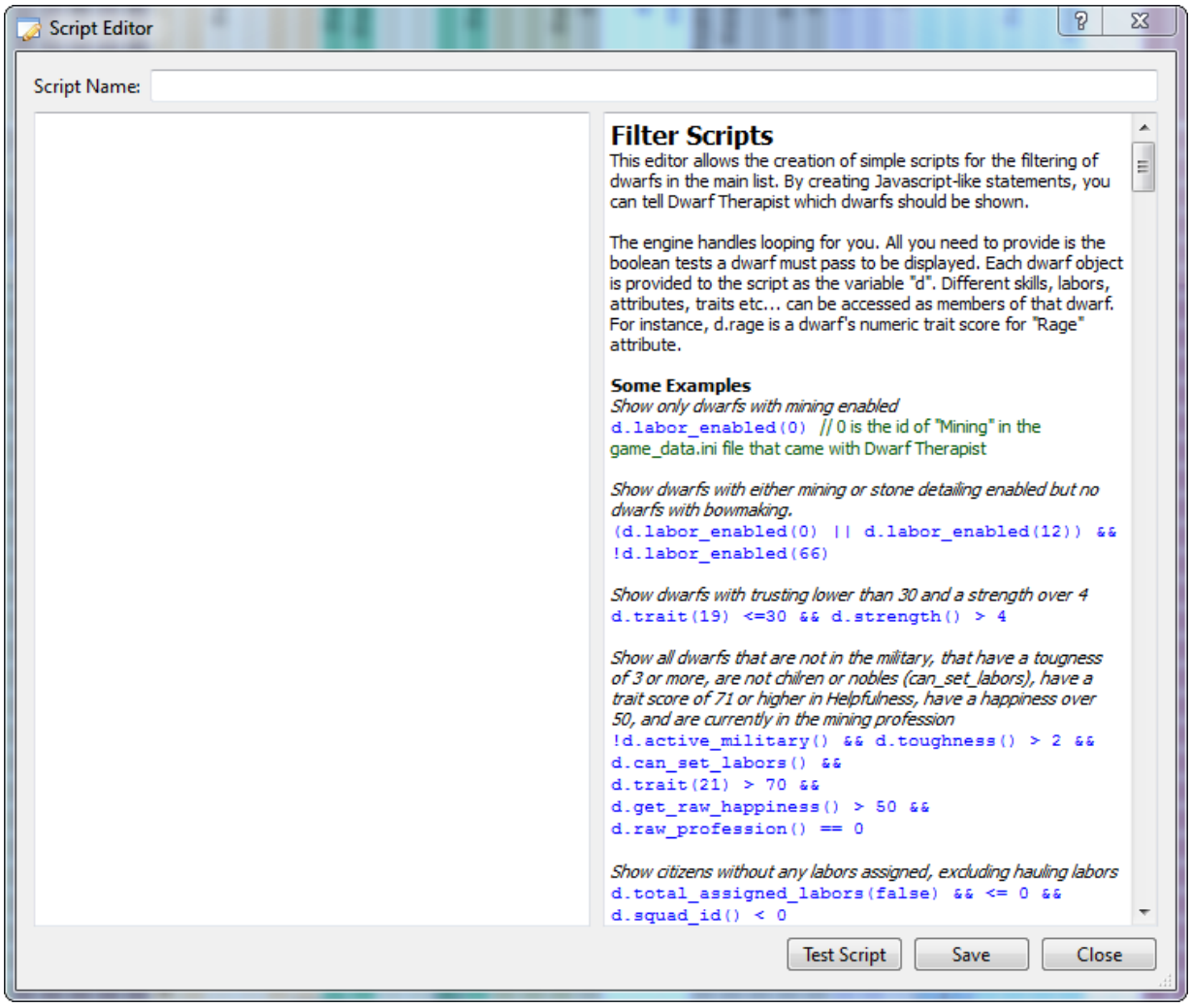

Dwarf Therapist - Ingishlocun, "Bodicerisen"
File Scripting Roles Optimizer Windows Help
Group By Nothing
Filter Dwarves
Run Filter Script None
70/13/5 Dwarves 0 Pending Changes
Add
Labor
Social
Military
Animals
Attributes
Current Job
Happiness
Profession
Mining
Carpentry
Bowyery
Wood Cutting
Woodcrafting
Masonry
Stone Detailing
Stonecrafting
Animal Training
Hunting
Trapping
Small Animal Dissection
Diagnosis
Surgery
Setting Bones
Suturing
Dressing Wounds
Farming
Soap Maker
Wood Burning
Potash Making
Lye Making
Milling
Pressing
Plant Gathering
Plant Processing
Brewing
Cooking
Milking
Cheese Making
Butchery
Tanning
Leatherworking
Bone Carving
Fishing
Fish Cleaning
Fish Dissection
Furnace Operating
Weaponsmithing
Armoring
Blacksmithing
Metalcrafting
Gem Cutting
Gem Setting
Glassmaking
Shearing
Spinning
Weaving
Clothesmaking
Dyeing
Strand Extraction
Pottery
Glazing
Mechanics
Architecture
Siege Engineering
Siege Operating
Pump Operating
Bee Keeping
Wax Working
Stone Hauling
Wood Hauling
Item Hauling
Burial
Food Hauling
Refuse Hauling
Furniture Hauling
Animal Hauling
Push Vehicles
Adil Gildclasps
Ås Lensmesh
Asën Faithwheels
Bëmbul Confusebridges
Catten Riverpaddle
Dastot Chamberjungle
Dastot Cobaltmatched
Dastot Sombersabres
Datan Gloveghost
Dodók Yellmirrored
Domas Copperwork
Doren Containedwheeled
Doren Inkyurge
Doren Stabbedcloisters
Eral Gravelwhip
Eral Quicknessmines
Èrith Chapelgears
Etur Bookpattern
Èzum Talkpapers
Fikod Chambermines
Ilral Bookswam
Imush Wirescraped
Ingish Lovehelm
Inod Chewearths
Kel Bulbousringed
Kikrost Crescentpillar
Kivish Tombsglided
Likot Fencedbud
Litast Tradewildness
Litast Tubewhip
Lokum Constructpeaceful
Lokum Fountainquake
Lòr Gearflaxen
Lòr Greenbrass
Lòr Tiredroom
Mebzuth Powertongs
Mebzuth Webbedpost
Medtob Skinwind
Melbil Puresling
Connected to v0.34.11 (graphics)

Part IV
Expert Features

10 Filter Scripts

Dwarf Fortress provides two different, easy-to-use tools for organizing your dwarves. In this section we'll introduce a third, much more advanced tool for the task: filter scripts. With custom filter scripts you can filter out those dwarves that you don't need to see right now from those that you do in a much more powerful, and much more refinable, way than grouping or sorting allows. However, it's also a challenging thing to do, as it requires that you learn and master the program's syntax for the task; indeed, if you do not have experience with writing programming code you could very easily become lost here.

Go to the dedicated Scripting item on the menubar and hit `Add New Filter Script` to bring up the script creation and modification dialogue:



As you can see, on the right side there is a fairly detailed demonstration of what filter scripts are meant to do, and then further down the list we get a list of methods we can use and then several tabulated references for labor, trait, skill, and attribute ids. You can actually collapse this away to the side or expand it to take up the whole view by dragging the right edge of the box.

Filter scripts work by generating a boolean test for your dwarves and then iterating through them. Dwarf Therapist handles looping for you, so your job is simply to write a useful filter: basically, at the end of the application of your filter to a dwarf, we want to end up with either a "true" or "false" statement. If the result is false, the dwarf will not be displayed; if it is true, they will be. All of the commands that you can enter into the script editor are all meant to be called on the "d" object, the abstracted dwarf in question, through the dot operator: `d.is_child()`, for instance. Here's a table of the commands available to you:

Command	Returns	Description
`is_child()`	boolean	Returns true if d is a child, else false.
`is_adult()`	boolean	Returns true if d is an adult, else false.
`is_animal()`	boolean	Returns true if d is an animal, else false.
`profession()`	string	Returns the basal profession string of a dwarf.
`raw_profession()`	integer	Returns the raw basal profession ID number of a dwarf.
`custom_profession_name()`	string	Returns the raw custom profession string, NULL if none.
`nice_name()`	string	Returns the nickname string, NULL if none.
`noble_position()`	string	Returns comma seperated list of the dwarf's noble positions.
`get_raw_happiness()`	integer	Returns the raw hapiness value of the dwarf.
`attribute(attribute_id)`	integer	Given the ID number, returns a dwarven attribute value.
`active_military()`	boolean	Returns true if the dwarf in an active military squad.
`squad_id()`	integer	Returns squad ID (from 0 by order of formation).
`can_set_labors()`	boolean	Returns false if dwarf is a child or baby.
`labor_enabled(id)`	boolean	Is this labor (by id number) enabled on this dwarf.
`is_labor_state_dirty(id)`	boolean	Returns whether there are pending changes for this skill.
`labor_rating(id)`	integer	Return's the dwarf's skill in a labor.
`trait(trait_id)`	integer	Returns raw trait value by ID number.
`total_assigned_labors()`	integer	Returns the total number of assigned labors for this dwarf. Optionally takes a boolean value: if set to false, does not include hauling labors.
`skill_level(int skill_id, bool raw, bool precise)`		Returns a floating point value, or "float". Raw uses the uncapped skill level, precise returns a decimal skill level.

In addition to being able to call on attributes through `attribute()`, you can also call on them directly by name:

```
int strength()                int focus()                   int spatial_sense()
int agility()                 int willpower()               int musicality()
int toughness()               int creativity()              int kinesthetic_sense()
int endurance()               int intuition()               int empathy()
int recuperation()            int patience()                int social_awareness()
int disease_resistance()      int memory()
int analytical_ability()      int linguistic_ability()
```

Unfortunately there are no equivalent shortcut for traits or for labors.

As for logical construction, Dwarf Therapist uses standard boolean operation notation. `||` stands for OR, and two statements linked in such a way will return true if either or both are true. `&&` means AND, and will only return true if *both* statements are true. Statements based around a `==` operator will be evaluated for equivalency, and will return true if they are, and false if they are not. Statements surrounded by `()` operators will be evaluated ahead in PEMDAS order, much like they are in algebra. Finally, adding a `!` before a boolean statement negates it - false becomes true and true becomes false. Integers can be compared with `<` and `>` operators as well as the `==` one. Strings must be surrounded in double quotes, `"Like This"`.

10.1 Writing Complex Scripts

Now that we have the basic syntax down, let's use filtering to solve a non-trivial problem. Whenever a large migrant waves arrive at a fortress, or even a small one in a lot of cases, what happens is that you pick the best of the bunch and set them to work immediately, and get the rest of the wave doing generic tasks that are plentiful and easy to designate - hauling things, getting rid of loose stone, building walls and other constructions, smoothing and engraving the fortress living spaces, and so forth. Integrating a new wave into your fortress is a slow but steady process, involving picking off individual dwarves for tasks you need done on a longer term one by one. Some will become jewelers, some will start operating furnaces, some will build parts for your pump stack, some will be enlisted in the miltiary - and some will remain plain old haulers.

In **"Using Roles"** we discussed combining groups, sorts, and roles to find ideal dwarves for a particular task within a group, but we were hamstrung by the limitations of a group: there's no easy way to group "gainfully employed" dwarves committed to their tasks apart from "part-time" ones that aren't. Your options were to make do with grouping them into migrant waves, which doesn't usually end with you selecting the best available dwarf in the *fortress* for the task, or not grouping them at all, which requires you browse through dwarves that are already working on something and risks switching the tasks of a dwarf you'd designated for a new role earlier (a problem discussed in **"Assigning Nicknames"**). We're going to neatly resolve this little problem with a script we're going to write.

Figure 24: In my fort I just started funneling my dwarves into tower fortress building duty.

Whenever you write a script the first question you've got to ask is, "what are the characteristics of a dwarf that I want" (if you're writing an *inclusive* script), or "what are the characteristics of a dwarf that I *don't* want" (if you're writing an *exclusive* one). In this case we're writing an inclusive filter meant to root out migrant dwarves that are available for full-time assignment to useful tasks. If we think about what this implies, we can come up with a number of "characteristics" for such dwarves (in order of complexity):

1. They have all hauling labors enabled. This is one of the most apparent indicators of a "working-class" dwarf, but it's nowhere near exclusive.

2. They don't have a nickname. Assuming you follow the advice given in **"Assigning Nicknames"**, nicked dwarves have already been dedicated to something. We're not actually going to consider this one, because dedicated nicked dwarves are already excluded by the condition above.

3. They have the masonry and/or stone detailing labors enabled, but are never above "adequate" skill in the former (since actually building stone blocks is assigned to dedicated masons), and never above "competent" in the latter (I consider "skilled" the breaking point for when an engraver is actually worth his salt, and should be taken out of the working-class pool).

4. They are never enlisted in the military (at least, not permanently). At least in *my* fortress, once a crossbowdwarf, always a crossbowdwarf slash hunter.

5. They never have certain "key" labors that are handled by dedicated dwarves enabled (mining, carpentry, woodcutting, stonecrafting, cooking, brewing, any of the metalsmithing tasks, either of the jewelry tasks, and clothesmaking). The precise composition of this list may vary somewhat for you, but most players learn to dedicate certain tasks to certain dwarves to maximize results very quickly.

6. If they are skilled in certain useless or niche tasks (animal caretaking, small animal dissection, fish dissection, bee keeping, wax working, soap making) they have those tasks enabled. No one creates extra work for themselves by actually turning these *off*, at least not until you want this dwarf doing something for you.

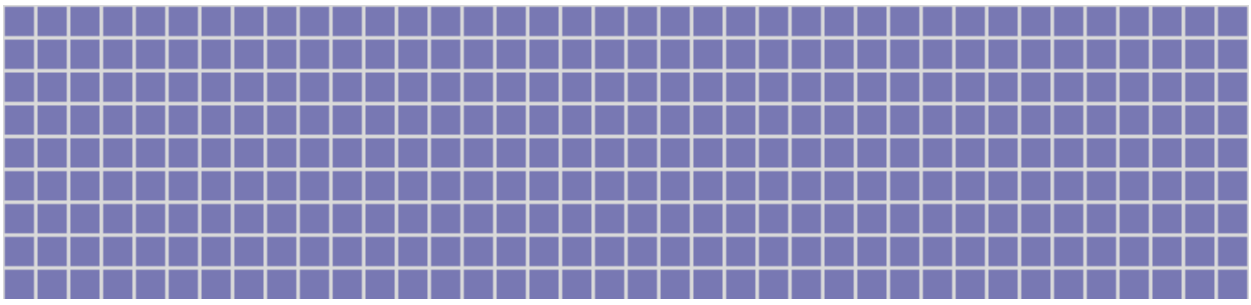
Now this is a pretty extensive list of characteristics common to our "working class" dwarves, and most of them require we do some thinking through as they're not immediately obvious. Hopefully this set of conditions both accurate and precise enough to work. To write such a complex script, let's break it down into individual steps.

All Hauling Labors are Enabled

This one's pretty annoying; we have to form an AND string out of `labor_enabled() == true` calls. However, a smarter solution that requires less work, both by the program and by us, would be to call `total_assigned_labors()` twice, once asking for labors with hauling included, once without, and subtract to see if we get the number we want. The behavior of this command is actually non-trivial: not only regular hauling but the two medical chores are folded in as well, and this isn't said anywhere in the documentation (well, now it is). That means that our magic number is 11:

```
d.total_assigned_labors(true) - d.total_assigned_labors(false) == 11
```

And if you plug that into your script editor and hit "Test Script" this is what you'll see:



This script alone reduced my fortress from 70 to 38 candidates! In fact, the capacity to filter your dwarves by whether or not they have hauling enabled is actually extremely useful on its own, and there's no other way to do this kind of thing in the program. So let's keep it! Give the script a name and hit "Save". We can actually also write a quick Hauling *Disabled* script, too, now:

```
d.total_assigned_labors(true) - d.total_assigned_labors(false) <= 10
```

If Masonry or Stone Detailing are Enabled, they are at a low Skill Level

Masonry is labor number 13 on the list, and stone detailing is number 12. We write separate tests, one asking if the dwarf has masonry enabled but is below skill level 3 in it, and one asking if they have stone detailing enabled but are below skill level 4 in it. We then link these with an OR. This is what the complete condition test looks like:

```
(((d.labor_enabled(13) == true &&  d.labor_rating(13) < 3) ||
d.labor_enabled(13) == false)||
((d.labor_enabled(12) == true &&  d.labor_rating(12) < 4) ||
d.labor_enabled(12) == false))
```

Then we create a new "Available for Work" script, and within it link the two statements we've written so far with an AND. So far, so good: on to the next condition.

They are Never Enlisted in the Military

This one's elementary, just test if the dwarf is in an active military squad and slap a `!` on it:

```
!d.active_military()
```

This isn't that useful as a standalone filter because there's a group for it. Next!

They Never have Certain Key Labors Enabled

This one involves a lot of table lookup:

```
!d.labor_enabled(47) &&       !d.labor_enabled(38) &&       !d.labor_enabled(00) &&
!d.labor_enabled(48) &&       !d.labor_enabled(45) &&       !d.labor_enabled(53) &&
!d.labor_enabled(29) &&       !d.labor_enabled(50) &&       !d.labor_enabled(46) &&
!d.labor_enabled(11) &&       !d.labor_enabled(51) &&
!d.labor_enabled(33) &&       !d.labor_enabled(49) &&
```

If you drop the negations and use OR statements, you pretty much get a list of key dwarves in your fortress: I call it "Key Dwarves". It's very useful for getting rid of new migrants who happen to be novices in skills that are dedicated to, say, my legendary woodcutter, so that they don't take up room at my carpenter's shop while my *real* bedmaker sits around doing nothing in particular.

Niche and Useless Labors Still On

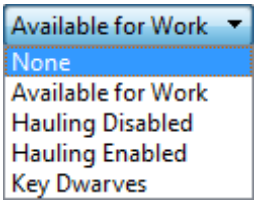
This is pretty much an inversion of the above: there we had key labors off, here we have useless ones on, but since it's conditional (and we don't have if statements available to us) it's a little more complex. It's important to note that dabbling is considered skill zero - *completely* unskilled is considered skill level -1. So it's time for some more table lookup:

```
((d.labor_rating(16) > 0 && d.labor_enabled(16)) || (d.labor_rating(16) <= 0)) &&
((d.labor_rating(43) > 0 && d.labor_enabled(43)) || (d.labor_rating(43) <= 0)) &&
((d.labor_rating(26) > 0 && d.labor_enabled(26)) || (d.labor_rating(26) <= 0)) &&
((d.labor_rating(72) > 0 && d.labor_enabled(72)) || (d.labor_rating(72) <= 0)) &&
((d.labor_rating(71) > 0 && d.labor_enabled(71)) || (d.labor_rating(71) <= 0))
```

Those are all our conditions; behold, the completed script!

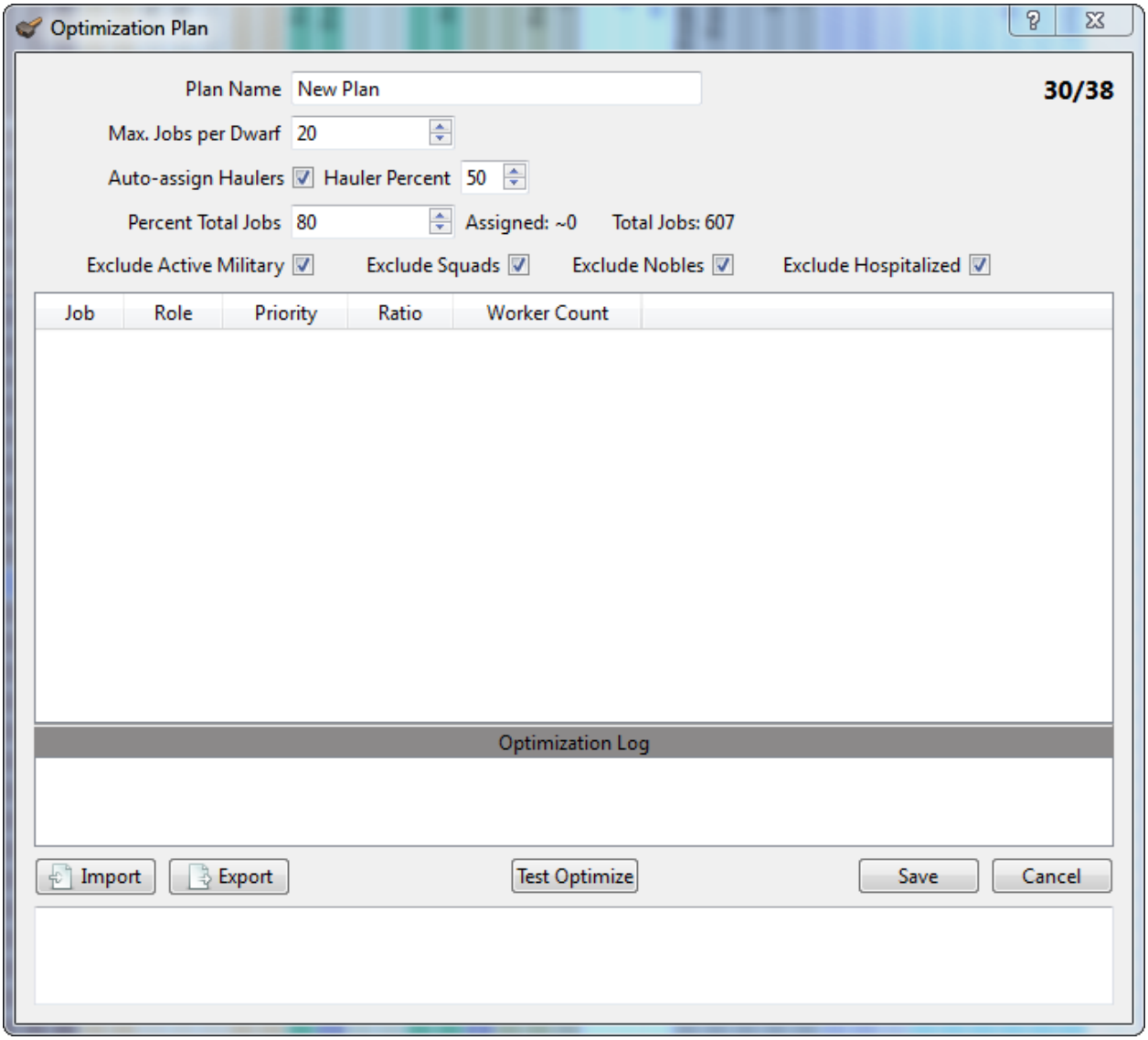
```
(d.total_assigned_labors(true) - d.total_assigned_labors(false) == 11) &&
// masonry and stone detailing test
(((d.labor_enabled(13) == true &&  d.labor_rating(13) < 3)
|| d.labor_enabled(13) == false) ||
((d.labor_enabled(12) == true &&  d.labor_rating(12) < 4)
|| d.labor_enabled(12) == false)) &&
!active_military() &&
(!d.labor_enabled(47) && !d.labor_enabled(48) && !d.labor_enabled(29) &&
!d.labor_enabled(11) && !d.labor_enabled(33) && !d.labor_enabled(38) &&
!d.labor_enabled(45) && !d.labor_enabled(50) && !d.labor_enabled(51) &&
!d.labor_enabled(49) && !d.labor_enabled(00) && !d.labor_enabled(53) &&
!d.labor_enabled(46)) &&
((d.labor_rating(16) > 0 && d.labor_enabled(16)) || (d.labor_rating(16) <= 0)) &&
((d.labor_rating(43) > 0 && d.labor_enabled(43)) || (d.labor_rating(43) <= 0)) &&
((d.labor_rating(26) > 0 && d.labor_enabled(26)) || (d.labor_rating(26) <= 0)) &&
((d.labor_rating(72) > 0 && d.labor_enabled(72)) || (d.labor_rating(72) <= 0)) &&
((d.labor_rating(71) > 0 && d.labor_enabled(71)) || (d.labor_rating(71) <= 0))
```

Over the course of composing this script we actually wrote three other useful scripts, which I think is a *pretty good* demonstration of their utility. Unfortunately Dwarf Therapist does not provide any facilities for exporting or importing filter scripts, which is a unfortunate, because filter scripts get very complicated and having to copy other people's scripts manually is very annoying, as even a single error or typo will likely invalidate the entire thing. If you like the scripts we wrote here, save yourself some trouble and simply copy paste them into your filters list!



11 Optimization Plans

Optimization plans are a (somewhat complicated) way to automate the labor assignment process that we've so far been doing manually. To access it, go to `Optimization > New Optimization Plan`, which should bring up a screen similar to the following one:



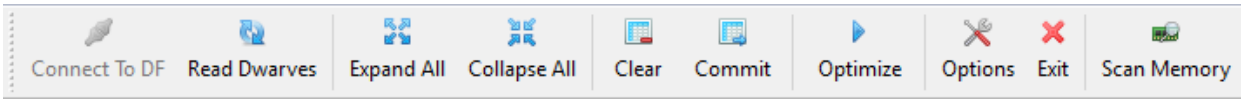
This isn't the easiest of screens to piece apart, so let's look at what each option is meant to do first. It isn't immediately apparent, but the labor optimizer works by assigning jobs in a ratio-wise manner. The two numbers in the top right corner tell you how many dwarves are being considered, and how many of those dwarves will have their labors specialized in the current assignment scheme. If you do not select any dwarves the optimizer will work with all available dwarves, but if you select a dwarf (by clicking on their name) or multiple dwarves (by dragging the mouse down dwarves in a column, or by clicking on the first dwarf and shift-clicking on the last one) it will constrain itself to those dwarves.[23] You can use filters to only display those dwarves that you want to be optimized - I recommend using the "Available for Work" filter we just wrote in the last section, **"Writing Complex Scripts"**.

Let's use the optimizer for a simple task - say, assigning (role-optimized) dedicated wood burner and furnace operator from betwixt a few dwarves. Open a new optimization plan again. Set "max jobs per dwarf" to 1. This obviously restricts the maximum number of jobs a dwarf can be assigned to just one job, which makes sense given what we're trying to do. Disable "auto-assign haulers" - this is a more advanced feature that we'll talk about in a moment. Now set "Percent Total Jobs" to 100. This number tells the optimizer that we want all of the jobs that these dwarves can be assigned to be assigned in our optimization scheme.

Roll your mouse over the empty white space and right click to bring up a menu of labors, and select first "Wood Burning" and second "Smelting" from the list. You can add any number of labors to the list in this manner, and you can easily delete any labors you've already added by right clicking and hitting "Remove Selected".

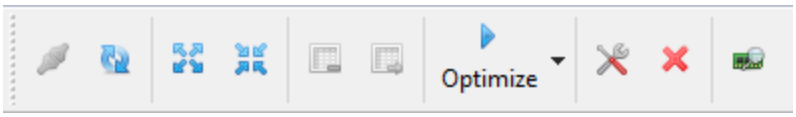
The first thing listed here is the job - the labor title that you've select. The next item is the role by which the optimizer is examining your dwarves' fitness: the one assigned to the job at hand by default. It's possible to use custom roles here. For the task at hand, however, leave this as is. Priority, meanwhile, changes the weight with which role fitness is considered, and so shifts overlaps in favor of the higher-priority labor. The rati changes the amount of dwarves that will be assigned this labor by the optimizer relative to the set. For any particular job the ratio is divided by the sum of the ratios, giving you the percentage of jobs that will be assigned under that labor, which when multiplied by the number of jobs to be assigned gives you the last column - the worker count.

We don't have to actually change anything here: everything seems to be in order, with one job going out for wood burning and one for furnace operating. Let's name our plan "Test Plan" and save it. This allows us to edit or remove our new optimization plan from its associated taskbar item, and you can test your plan in-view using the "Test Optimize" button. Properly applying your designation, however, requires a new item that appears on your main menu bar - the optimization button.

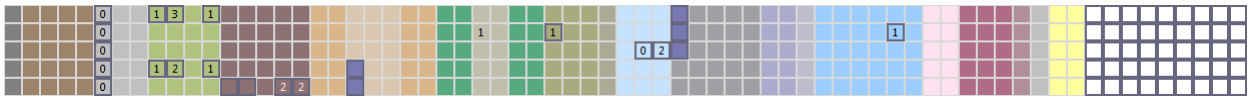


This button will not appear when you don't have any optimization plans saved. When you have multiple optimization plans, a button appears besides it that lets you drop down a menu to select which script is the active one - this also applies the script in question, saving you a mouse click:

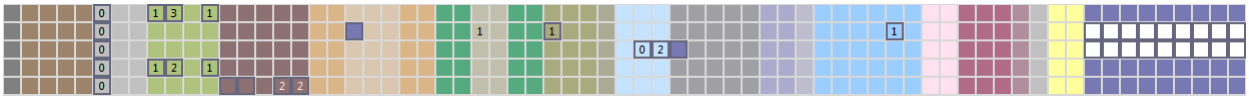
[23] You can also select all with the `Ctrl + A` hotkey.



Now selecting any number (or, as earlier, selecting none and therefore all) of dwarves and hitting Optimize will cleave them in two, between the wood burners and the smelters. If the number of dwarves selected is odd you will get an uneven number of job divisions - in this case three furnace operators but only two wood burners.

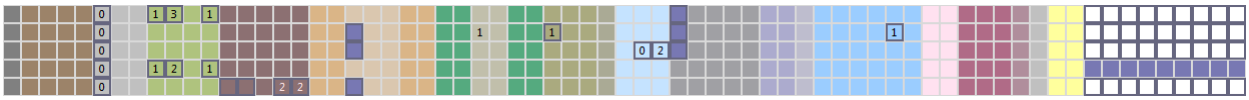


Now let's add a fold of complexity to our plan by including hauling as well. To edit an optimization plan, go to "Optimizer" on the taskbar and select the plan from the "Edit Optimization Plan" menu. This time let's turn auto-assign haulers on and set it to 50 percent: likewise, set the total jobs percentage to 50 as well. Now hit "Test Optimize" at the bottom of the window to see what effect this plan has on our dwarves (I deliberately avoided doing this earlier to showcase predefined usage and the changes to the menu bar). As you might have guessed this cleaves our dwarves in two: a quarter each are assigned dedicated wood burning or furnace operating, and the remaining half are made haulers, with their other labors disabled. A note that needs to be made: optimization plans will obviously only work properly if roles are tuned to the default skill rank setting we discussed in **"Roles"**.



From our little group of five dwarves this results in one wood burner, one furnace operator, and three haulers. Once again because five is indivisable by two the division is uneven, with more haulers being assigned than workers. This is the reason that the numbers that the dialogue gives us are approximate - if the numbers don't fold there will inevitably be some fudging on the corners.

Let's dig a little deeper. Set the maximum number of jobs a dwarf can have to 2 this time, and test the script again to see what effect this has:



Unless you're particularly insightful, it's probably not immediately apparent what's going on here - why is only one dwarf hauling anything? So far we've generated three optimization scenarios; now let's use them to get a better understanding of how the labor optimizer works.

The percent total jobs parameter is a cap on how many jobs can be assigned, out of how many total possible jobs there are. In the first example this number was set to 100, so the labor optimizer could give every dwarf a job to do. Since the max jobs per dwarf paramter was set to 1, every dwarf could only get one job total. Since we have two labors of equal priority under consideration, this means that half of the dwarves will recieve the wood burning task, and nothing else, and half would recieve the furnace operating task, and nothing else.

In the second example, we reduced the cap to fifty percent, and kept the number of jobs per dwarf at 1. This means that *fifty* percent of dwarves will be assigned *one* job. Indeed, ignoring some fudging that the program had to do, again because five is not divisible by two, half of half of the dwarves recieved the wood burning labor and half of half of them recieved the furnace operating labor.

How did the program behave in the last example? This time the percent labor cap (let's just call it the labor cap from now on) was still set to 50, but the maximum number of allowed jobs (let's just call it the job cap from now on) was set to 2. We still have two labors, but now we also have two "spaces" for them to be assigned to. In the resulting configuration *fifty* percent of dwarves are assigned wood burning, but then *another* fifty percent are *independently* assigned furnace operating. Logically, because the labor optimizer has been allowed to assign jobs twice instead of only once, dwarves that are fitter than the group average at both wood burning *and* furnace operating will be assigned both jobs. The labor cap no longer regulates how many jobs the two labors together can be assigned, but how many times they can appear individually. Under the single-job scheme, two labors had to contend against one another under the fifty percent job cap, and so each labor could only be turned on 25 percent of the time; under the two-jobs scheme this is no longer the case, and so each labor could be turned on a full 50 percent of the time.

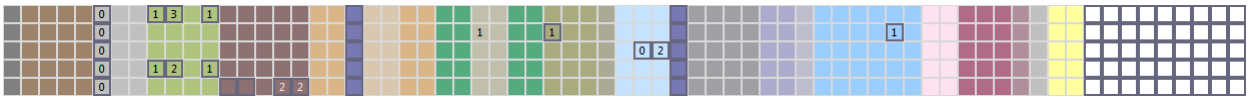
In another form, the number of labors that will be turned on will thus be given by the following equation (note that you have to divide the labor cap by a hundred because it is in percentage form):

$$jobs\ cap \times number\ of\ dwarves \times \frac{labor\ cap}{100} = number\ of\ labors\ to\ be\ assigned$$

And the number of assignments for each individual labor, which is dependent on the ratio, is as follows:

$$number\ of\ labors\ to\ be\ assigned \times \frac{that\ labor's\ ratio}{sum\ of\ labor\ ratios}$$

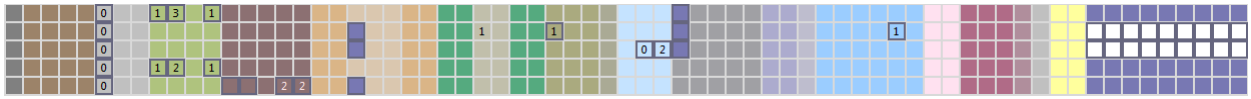
Let's take this to its logical conclusion: what would happen if we kept jobs cap at 2, but raised the labor cap to 100 percent? Logically that would mean that every instance of that labor will be assigned - and a quick test confirms that, indeed, that is the case:



What about the funkiness with the assignment of hauling in example three? If you were following along with the examples I was giving, and didn't check the tooltip text on the Hauler Percent box in the dialogue, you might have (wrongly) concluded based on the parameter's behavior in examples one and two that the optimizer assigns hauling to a certain percentage of the total dwarves under consideration. That is not the case: rather, that percentage is a bit of dwarf-wise post-processing optimization based on how many labors that dwarf already has assigned.

Basically, the optimizer assigns labors, then goes down the list and asks each dwarf how many labors they have assigned: one? Two? None? It then compares this to the "Hauler Percent" value (hauler threshold from now on), and if the dwarf has fewer than that percent of the maximum number of labors assigned, then they are assigned hauling tasks. In the example above every dwarf has both labors enabled, and so obviously no dwarf falls under the fifty percent threshold - and no dwarf is assigned hauling. In example two three of our dwarves don't have anything assigned post-processing, so they obviously get hauling booted onto them.

Example three actually demonstrates an optimization behavior that's important to know: that the hauling threshold is strict. Dwarves one and five both have one labor assigned, which is half of two, the number of labors they *could* have assigned. In percent form one over two is fifty percent: on the border of our threshold, but not below it. To get the behavior we want, increase the threshold to 100 percent:

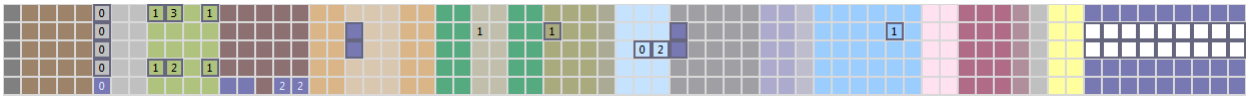


The number required is 100 percent, and not 51 percent, because of the program's behavior.

There's just one more thing to be said: you can exclude various groupings from optimization right from the dialogue by unchecking their associated boxes:

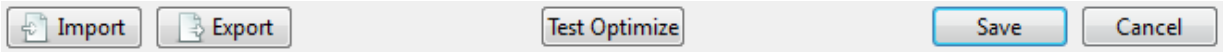
Excluding hospitalized dwarves is important if you want the jobs done now, but isn't necessary if no one (or no one freely available to work *and* under consideration) is injured, and may be counterproductive if on the flip side a significant number of your dwarves have sustained recoverable wounds. Excluding nobles is a poorly optimized solution because while assigned nobles work as normal, royal nobles will not have much of a work ethic regardless of what tasks you assign to them - you usually won't be considering nobles regardless, so not excluding them is a safe bet. There is some overlap between "active military" and "squads": dwarves are considered to be on active duty when they are on call, correspondingly appearing by their military position in-game - but to get that far they obviously have to be assigned to a squad first. Excluding squads is important when your military is fully professional, but of dubious use when you have a attack-that-thing-now dwarven militia that you never disbanded.

As it turns out, dwarf number five in our examples above is actually a noble - I just haven't been excluding him in the optimization process. Doing so results in exactly the sort of behavior you would expect - he's ignored - and has the nice side effect of tidying up our "five divided by two" problem.



11.1 Exporting and Importing Optimization Plans

Like most of the configurable utilities packaged into Dwarf Therapist, optimization plans can be exported and made available for others to use, or imported and made available for use in your program.



This feature is only available from the options bar on the optimization plan editing menu, and it doesn't appear in the Optimization taskbar item. This will be fixed in the next version.

12 Addendum: Scanning Memory

If you click on the "Scan Memory" icon on the main toolbar, you will be brought to the Dwarf Therapist memory scanning utility:

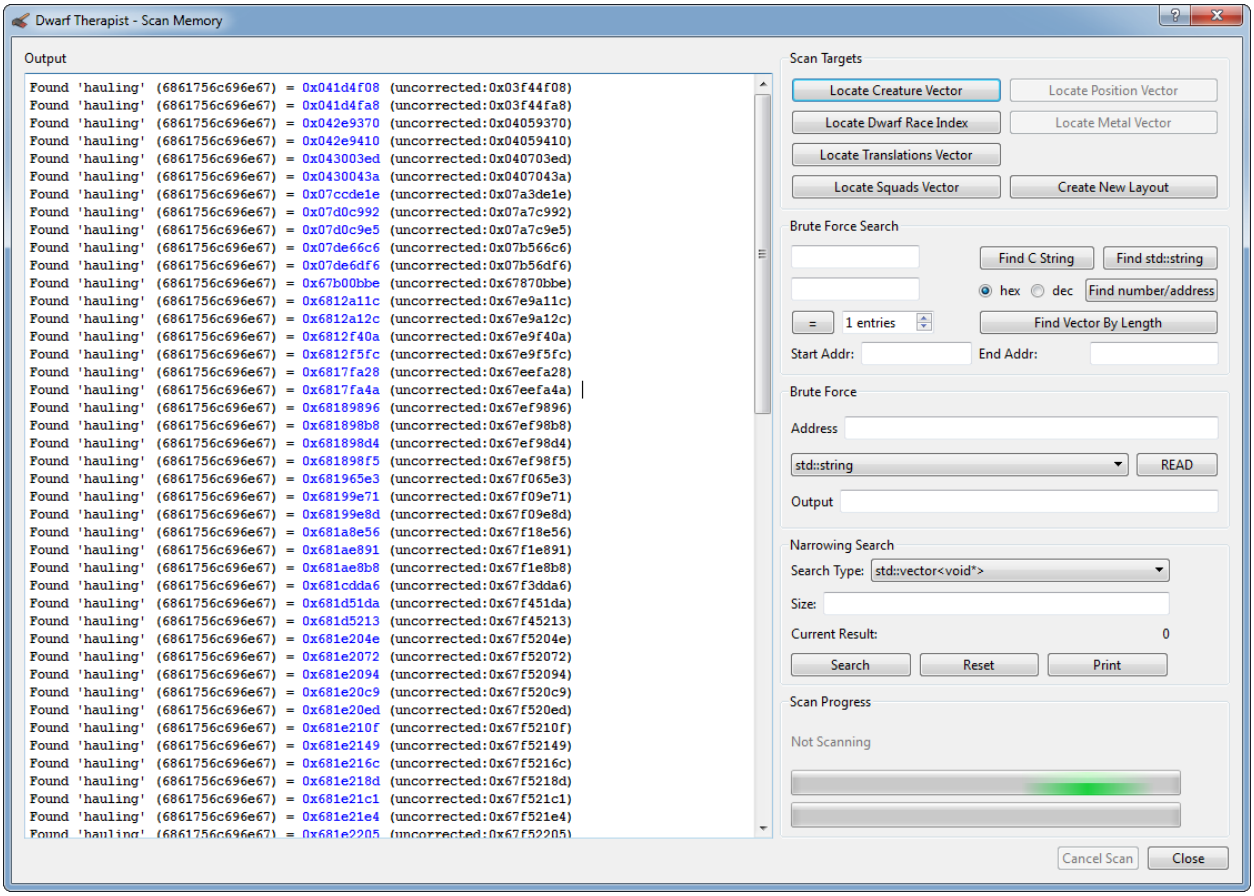
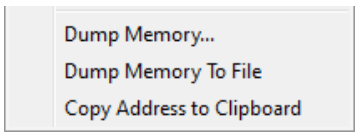


Figure 25: The result of a brute force string search for "hauling".

This will not be covered in this guide. Dwarf Fortress memory mapping and modification is not for the faint of heart, and is used for two things: finding the RAM values for things necessary to get complex utilities and mods working (like, say, Dwarf Therapist), and cheating the game by changing them. The first is way out of the scope of this guide, and for the second there are much more elegant tools available: as labor management has been given a GUI by the community, so to has memory access and modification. I'm also not going to pretend I fully understand how the tool even *works*.



Additionally, if you right click on a dwarf's name you will see that there are some memory options available for individual dwarves as well. You can "Dump Memory" to open the memory matrix associated with that dwarf tabulated in a text box by the program. "Dump Memory to File" will also read the dwarf's memory values into a chart, but will save it to a `TXT` file in the Dwarf Therapist log subfolder. Finally, you can "Copy Address to Clipboard" to copy that dwarf's memory location to your clipboard. This function may be useful if you're interested in modding your dwarves in-game, but once again it's not something the average user would need.

Part V
Appendix

13 Hotkeys

Although the main functions that you'll be performing with the program have hotkeys, overall hotkey support is still quite limited.

Hotkey	Action	Refer To
Commands		
`Ctrl + C`	Connect to Dwarf Fortress	**"Connecting to Dwarf Fortress"**
`Ctrl + R`	Read Dwarves	**"Connecting to Dwarf Fortress"**
`Ctrl + T`	Commit	**"Managing Your Dwarves"**
`Ctrl + E`	Clear	**"Managing Your Dwarves"**
`Ctrl + S`	Snapshot	**"Menu Bar"**
`Ctrl + P`	Options	**"Options"**
Views (when the View is active)		
`Scroll`	Scroll Vertically	**"Labors View"**
`Alt + Scroll`	Scroll Horizontally	**"Labors View"**

Groups (when Groups are active)
Hit the key that corresponds with the first letter of the group's name to bring it up.
If there are multiple groups that begin with that letter, you can cycle between them using that key.

14 Modifying Game Data

`game_data.ini` is a .ini file packaged with Dwarf Therapist that describes standard settings for a number of things, divided between some RAM values, skill levels, position flags, profession IDs, attributes and attribute strings, military preferences, skill names, traits, dwarven job IDs, thoughts, and a very, very long list of role modifiers; if you're interested in the profession and job sort key, the master ID list is here. You can change these values (and therefore the program's behavior) yourself by modifying the document in any TXT editor to change the program's behavior. Be warned: messing with it can cause all sorts of strangeness and even crash the program on startup, and should you update Dwarf Therapist the changes you've made to this document will be lost. Since the default settings are fine for the most part (the happiness bonuses/penalties for certain thoughts are complete guesswork, but that's because they are unknown to anyone), you should only modify stuff here if you *really* know what you are doing.