The Big Book of SnowRunner Map Making

Revision 0.1 (2021-08-21)

If you own the Windows version of SnowRunner, then you can make and publish your own custom maps.

This guide documents every feature of the SnowRunner map editor, plus how to create custom objects for your map. You can read (or skim) sequentially through the guide to learn everything there is to know, or you can use it as a reference when you want to learn more about a particular feature.

What I put in this guide that I can't find elsewhere:

- Information about **all** editable map properties.
- Full details for complex editing tasks.
- Bugs to avoid.
- How to create and use reference maps.
- How to create custom materials, distribution brushes, roads, dynamic models, and even plants.
- Easy reference material for trucks, trailers, addons, and their permitted combinations.
- Tips for using Blender to create custom objects for SnowRunner.
- And more!

Table of Contents

Quick Start

Before getting into the nitty gritty of map editing, you'll want to create and try your first simple map as quickly as possible.

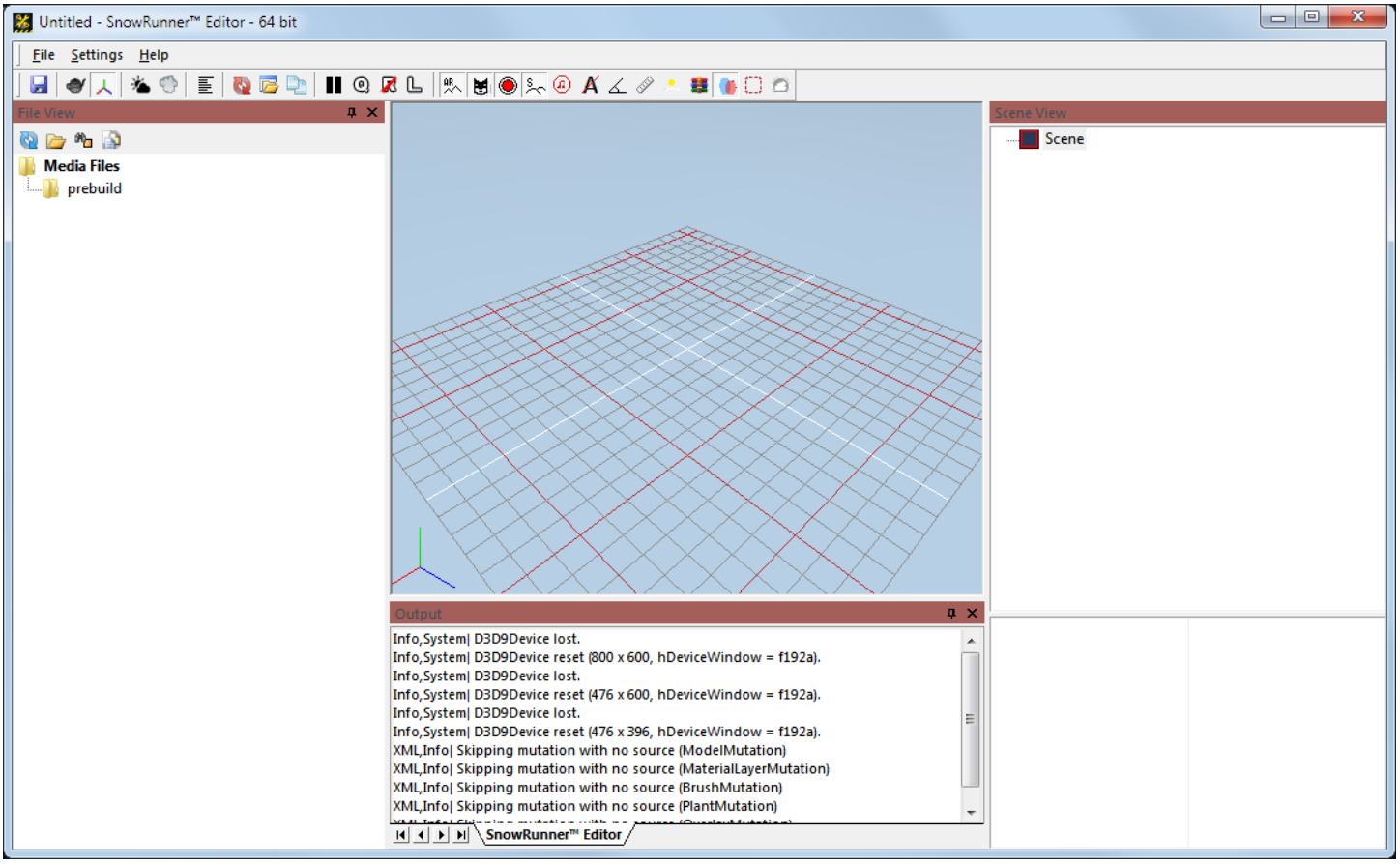
Run the Map Editor

Installing the SnowRunner game also automatically installs the editor, but it doesn't install a convenient shortcut icon. You can start the editor from a path that looks something like this, depending on whether you have the Epic or Steam version of the game:

```
C:\Program Files\Epic Games\SnowRunner\en_us\Sources\BinEditor\SnowRunnerEditor.exe
C:\Program Files (x86)\Steam\SteamApps\common\SnowRunner\Sources\BinEditor\SnowRunnerEditor.exe
```

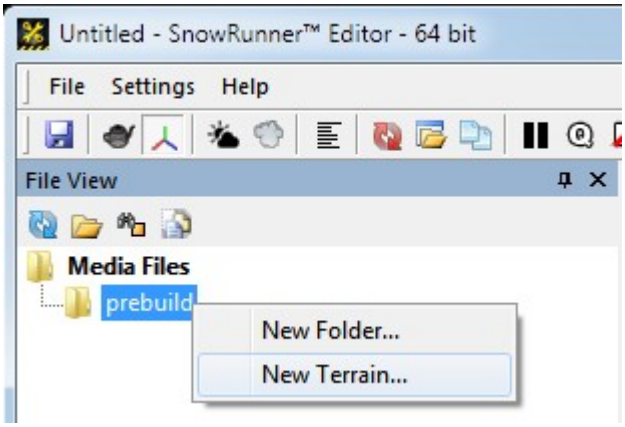
Saber claims that you can't run the SnowRunner Editor and the SnowRunner game at the same time, but I find that they run just fine simultaneously. But if you have any trouble, exit the game before starting the editor.

The initial map editor window looks like this:



Create a Fresh Map

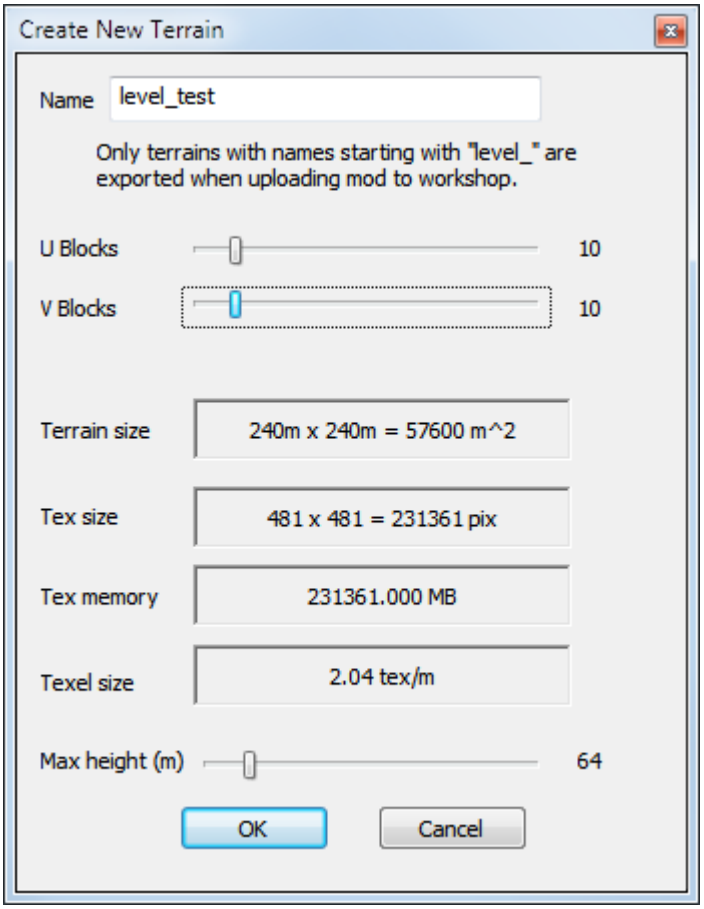
To create a new map, **right**-click on prebuild under Media Files, then select New Terrain...



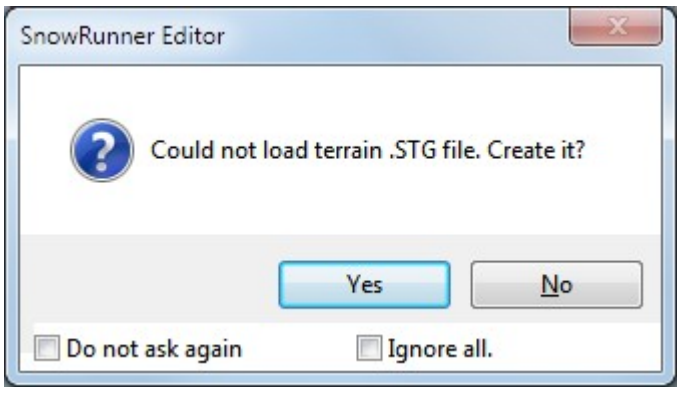
A dialog box pops up asking what size to make the map. The map name **must** begin with level_, so in this example I'll call it level_test. The default is a rather small map 64 meters on a side, so you might want to crank up the sliders for U Blocks and V Blocks to something like 10 and 10, which equates to 240 meters on a side. The default max height of 64 meters is fine.

Don't go wild with either of these values. A map with large U and V dimensions will be very slow to work with, and a large max height causes stair-stepping of the terrain.

When you're ready, click OK.

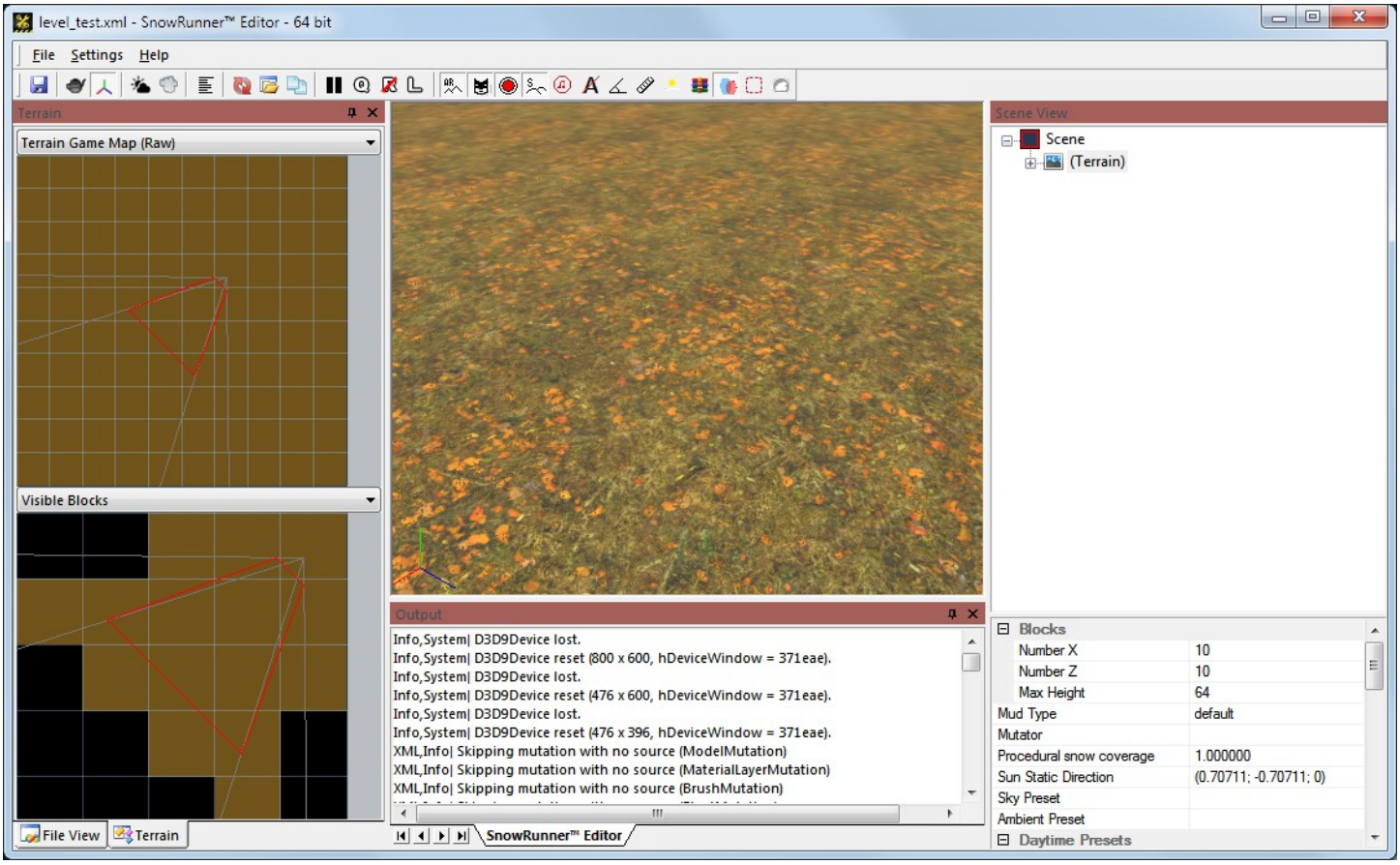


Because this is a new map, it does not yet have a terrain .STG file. Click Yes to create one.



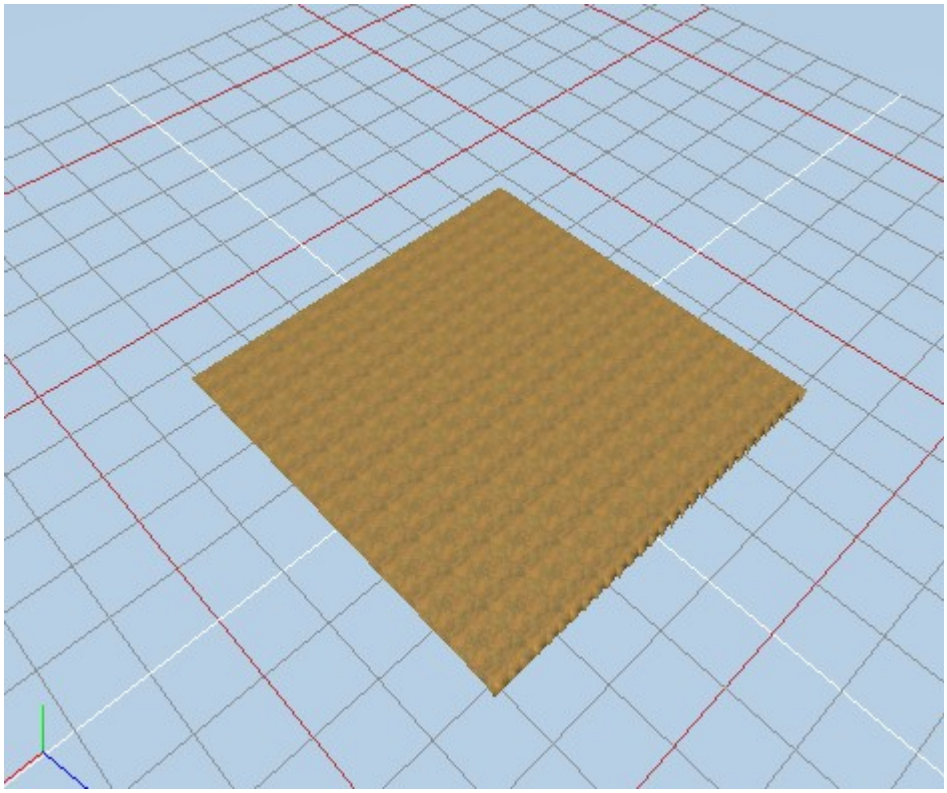
On the right side of the window, the Scene updates to include (Terrain).

Bug: Your first time running the editor, the camera is pointed in completely the wrong direction to see your terrain. To point the camera in the correct direction, double-click in the Terrain Game Map panel on the left. The main (center) panel now shows you a close-up look at the ground.



Move the Camera

Click in the main panel, then rotate the mouse scroll wheel up and down to zoom in and out. When you zoom out far enough, you can see the edges of the map.



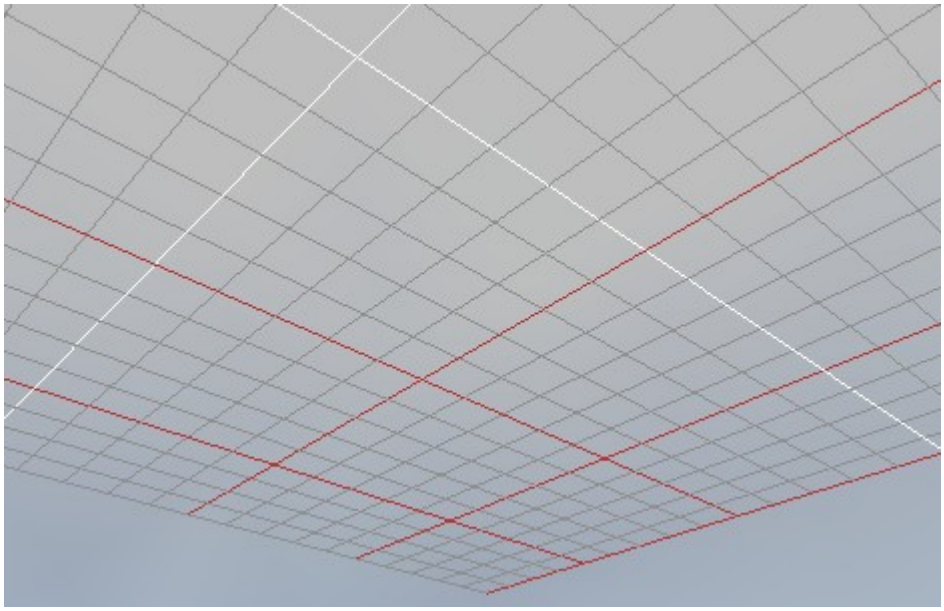
When you zoom in, the camera moves toward where the mouse is pointing. So an easy way to move the camera around the map is to zoom out, point the mouse at the area of interest, and zoom back in.

If you lose track of your terrain, double-click in the Terrain Game Map (Raw) panel on the left to restore the camera to a reasonable location.

You can point the camera in different directions by holding down the left mouse button in the main panel and moving the mouse. The camera pivots around the point that you were pointing at when you clicked the button, moving left, right, up, and down.

It is possible to move the camera below the ground. You're not expected to look at the terrain from underneath, so the terrain is only drawn from the top side. So when the camera is underground, all you see is the reference grid above. Rotate the camera back above the ground to restore your vision.

When the camera is pointed up, the background is shaded cloud gray. When the camera is pointed down as is usual, the background is shaded a pale blue.

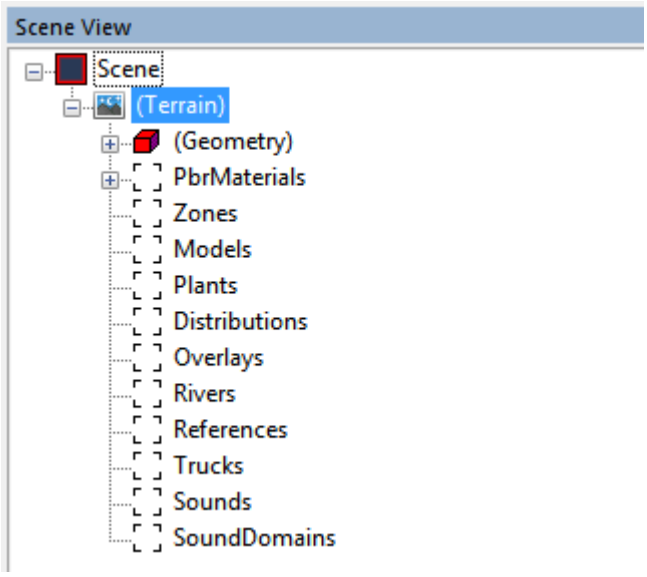


Bug: While moving the mouse to rotate the camera, the mouse pointer also moves. If it moves outside of the main panel, camera rotation will stop.

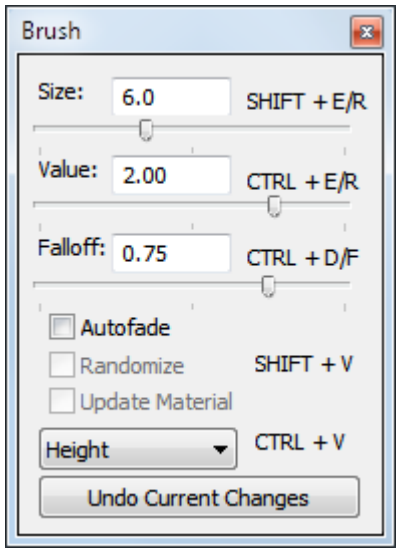
Make Hills and Depressions

The initial terrain is perfectly flat, which isn't very interesting. You can raise the terrain in some places and lower it in other places to make a more interesting terrain to drive around in.

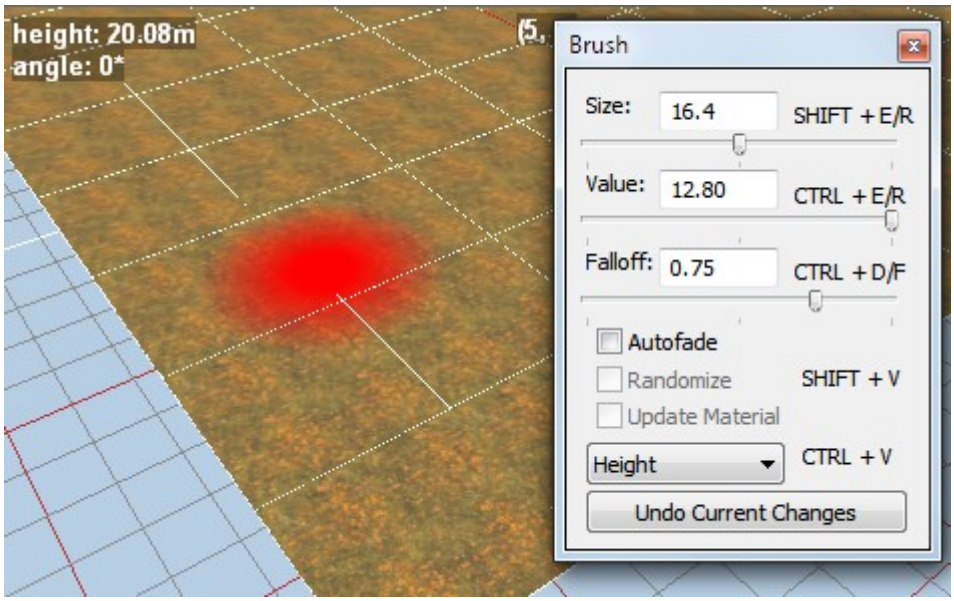
To edit the terrain height, first expand the hierarchy under Scene→(Terrain) to show the many editable aspects of your map, including its (Geometry). There is additional hierarchy within (Geometry), but you don't need to look at that yet.



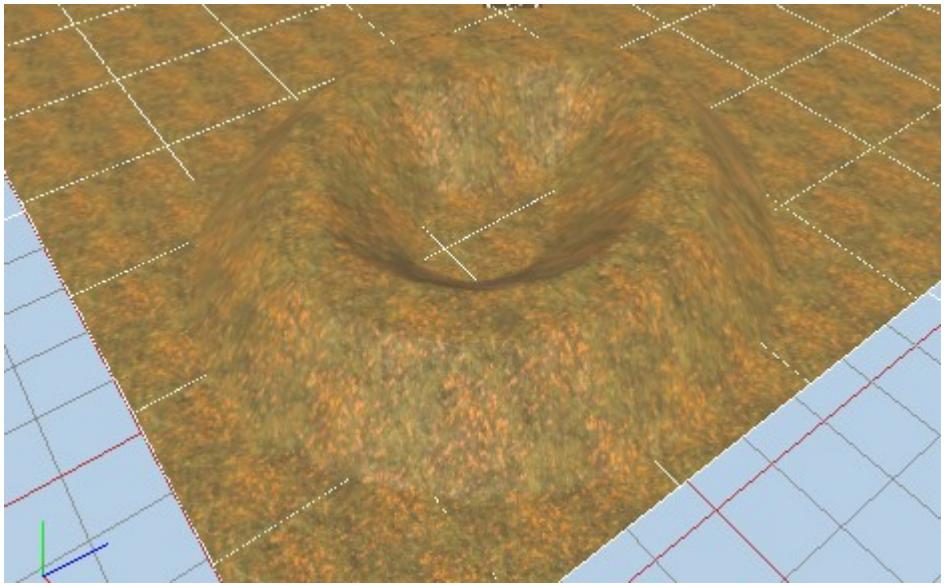
Left click on the (Geometry) element in the Scene View. This pops up a Brush dialog box. It's called a "brush" because you will be "painting" your terrain height changes onto the map. You can move the Brush dialog box to the side if you don't want it to obscure the main panel.



By default, your paintbrush shows up as a red dot on the terrain. Below I've increased the size and value of the brush to make it stand out more.



When the brush is active, you can zoom the camera in and out with the scroll wheel and rotate the camera with the left mouse button as usual. In addition, you can now adjust the terrain height with the right mouse button. With the mouse over the terrain in the main panel, hold down the right mouse button while moving the mouse. The terrain will increase in height by the amount of the brush Value if the value is posive, and it will decrease in height if the value is negative.



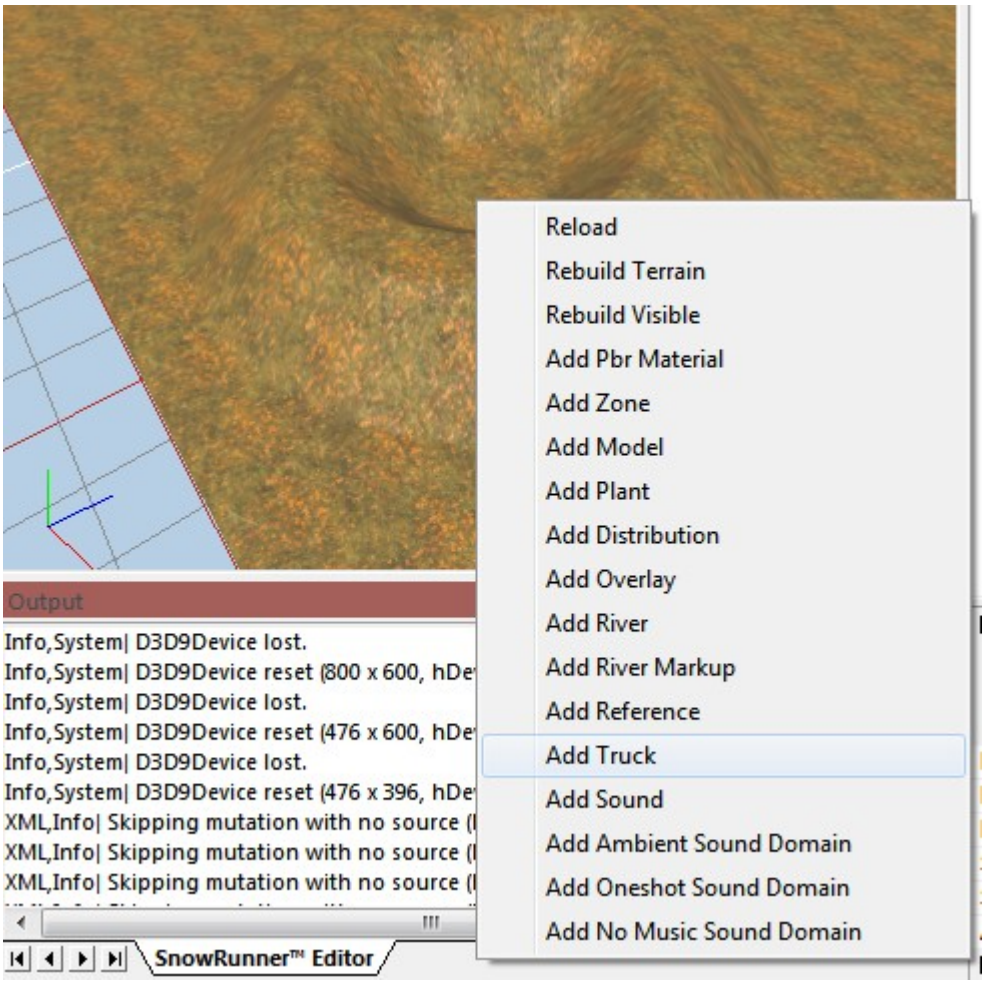
To make larger changes to the terrain, you can adjust the size and value of the brush, or you can make multiple passes with right click and drag.

When you are done editing the terrain height, left click in the main panel to exit painting mode. Note that left click and drag moves the camera while remaining in painting mode, while left click with no mouse movement leaves painting mode.

Place a Truck

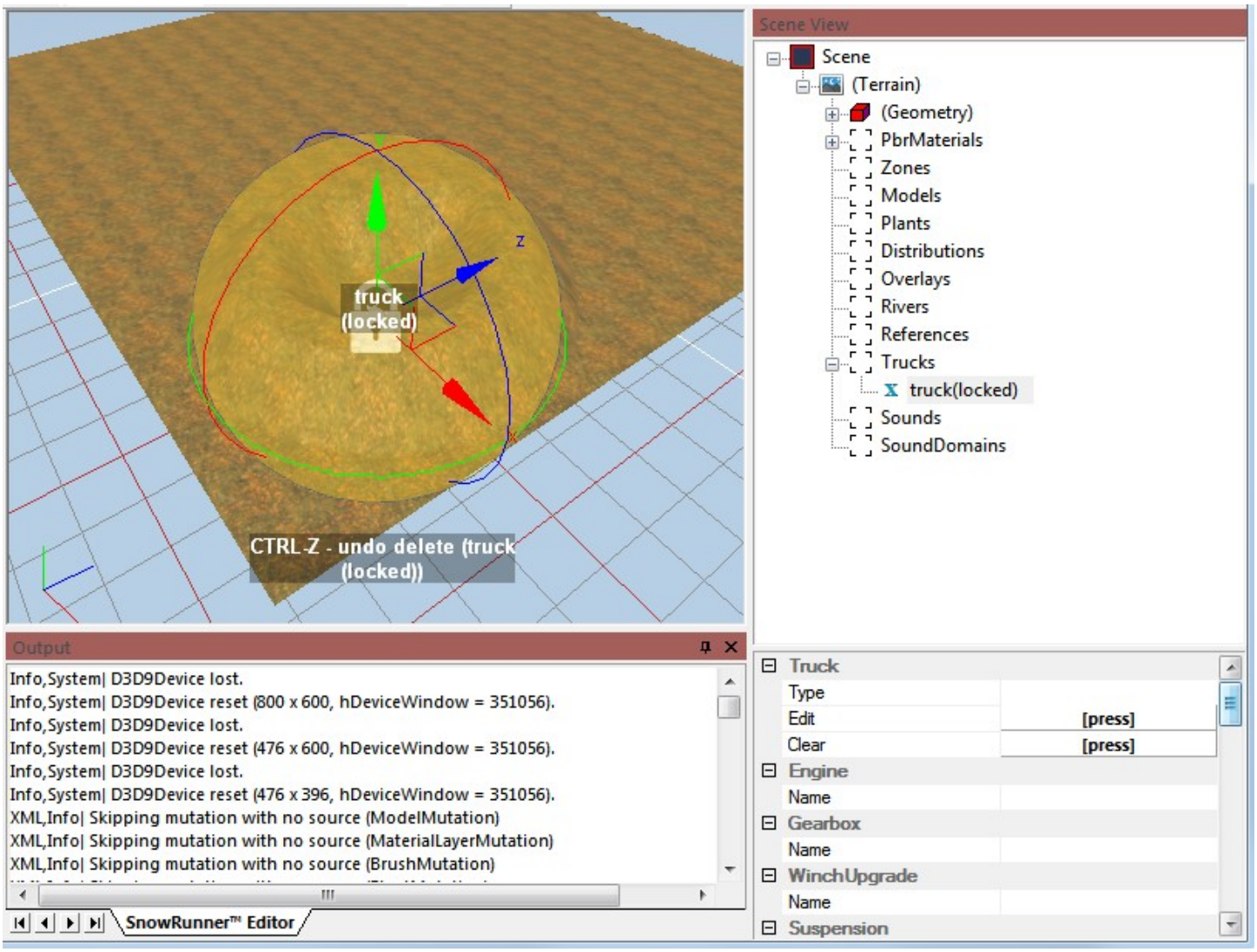
Now that you have some interesting terrain, you'll need a truck to drive around in it.

If necessary, left click to exit painting mode. Then point the mouse over the terrain in the main panel, right click, and select Add Truck from the context menu.

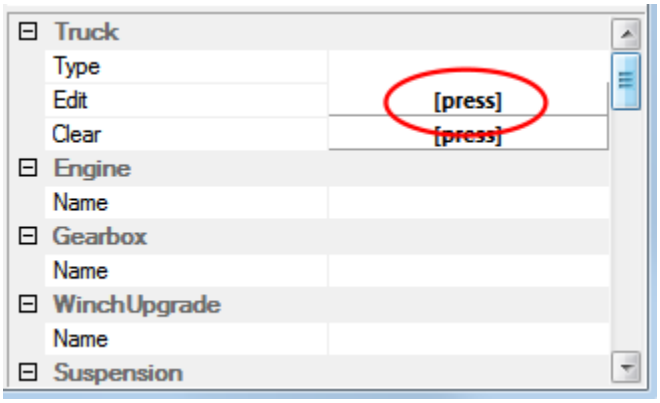


Bug: Unlike in most other programs, the context menu will not appear until you release the right mouse button.

A new truck appears in both the main panel and in the Scene View panel. Below the Scene View panel is an unlabeled panel that shows various parameters for the truck. I call this panel the property panel.



The truck that you've added doesn't yet have a type. To tell the editor which kind of truck it is, click in the property panel next to Truck → Edit where it says [press].



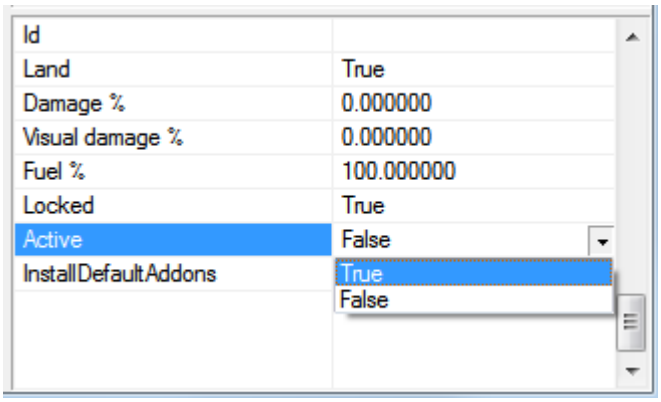
Bug: The first time you click here, the editor may lock up for 5 or more seconds as it creates icons for all possible trucks. It will be faster for subsequent uses.

The editor pops up a dialog listing all of the available kinds of trucks, as well as other things (such as trailers) that the game puts in the truck category.

For testing purposes, I like the ank_mk38 because it is a very capable truck in its base form, and because I don't have to scroll to find it. Click to select the truck that you like and click OK.

If you have experience with the MudRunner Editor, you may remember that changed feature properties did not commit until you changed the selection (e.g. by clicking elsewhere). However, the SnowRunner Editor mostly fixes this bug so that property changes take effect immediately. The few exceptions where the commit is delayed are described where appropriate.

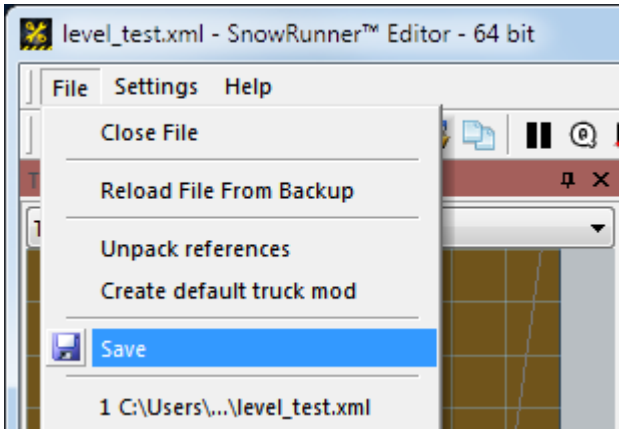
There is one more step to make this truck drivable. If necessary, re-select the truck by clicking it in the Scene View panel. Then scroll to the bottom of the property panel. Where it says Active False, click on False, and change the dropdown value to True.



The truck is now active.

Save and Pack

To save your work so far, select File → Save from the top menu bar or press the keyboard shortcut: Ctrl-S.



Now pack your map for testing. Click the stack of books in the toolbar. The toolbar is just under the top menu bar, and the stack of books is the fourth icon from the right end.

A dialog box warns that packing the level takes time. Click Yes to continue and wait for it to complete. For a map this simple, it will only take a few seconds.

Packing your map saves it automatically, so the previous save step wasn't strictly necessary.

Make a Backup Copy of your Campaign

In the course of working with the Editor, you will create a lot of new games for testing maps. SnowRunner gives you four save slots, so hopefully you have room to create these new games without disturbing the save files for your campaign, but there is always the possibility that you make a mistake.

> **Bug:** Even worse, on a couple of occasions I've left SnowRunner idling on the main menu in the background only to find out that it had spontaneously created a new game in a random save slot!

To avoid losing your campaign progress, I recommend making a copy of your data.

If you bought SnowRunner from the EPIC Store, your save files are in %USERPROFILE%\Documents\My Games\SnowRunner\base\storage.

If you bought SnowRunner from Steam, your save files are in a path something like C:\Program Files (x86)\Steam\userdata. That directory has a randomly numbered subdirectory. This subdirectory contains directories

for each of your Steam games which are also unhelpfully labeled with random strings of digits. If you've played SnowRunner recently, then its folder should be one of the most recently modified. Within that directory, your save data is stored in the remote directory.

Make a backup copy of all files in this directory. If you ever need to restore some files from backup, you can either restore them all, overwriting all four save slots, or you can pick and choose. But you definitely want to archive all of the files now so that you don't discover later that you missed an important one.

Test Your Map

Time to take your map for a spin. Exit the SnowRunner Editor and start the SnowRunner game.

> **Tip:** You have to wait through the epilepsy warning every time, but you can skip the rest of the ads by pressing the Escape key as soon as the epilepsy warning starts to fade out.

From the game's main menu, select New Game and then Custom Scenarios.

The game will list all available mods. Most likely this is only the one you created, e.g. mod_level_test. Select your map and a save slot for your custom game. The save slot must be separate from your save for the main game because progress for each mod is saved separately.

Bug: The game will not reliably replace an existing save with a new game. It asks for confirmation as if it is replacing the existing save, but then it loads the old game instead of the new game.

As far as I can tell, this bug always occurs when a new game is the first game loaded after SnowRunner is started. Exit the (faulty) game and try to create a new game again, it works.

I often leave SnowRunner running in the background as I repeatedly make edits to a map and test it. That means I don't have to wait for the game to start up, **and** it will work properly when I create a new game.

The game now starts in the truck that you chose and in the terrain that you drew.

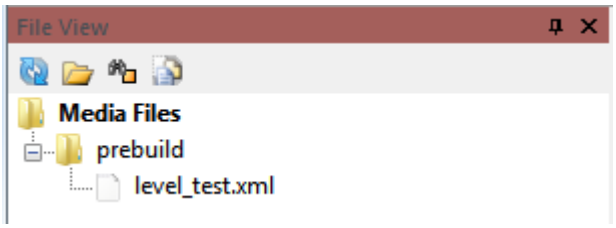
The game controls are all as usual, with one exception. There is now a permanent menu of Tools in the upper right of the screen. The Tools menu provides useful actions for testing such as spawning new trucks, allowing truck modification as if you are in a garage, etc. See the Dev Tools section on page 222 for more information about this menu.

Bug: The default grass and leaves texture on the terrain looks unnaturally large compared to the size of your truck. That can be fixed by manually adjusting the scale of the PbrMaterials. Recommended scale values are listed in section TBD.

Re-editing a Map

If you quit and restart the SnowRunner Editor, it will not automatically return to the map you were previously editing.

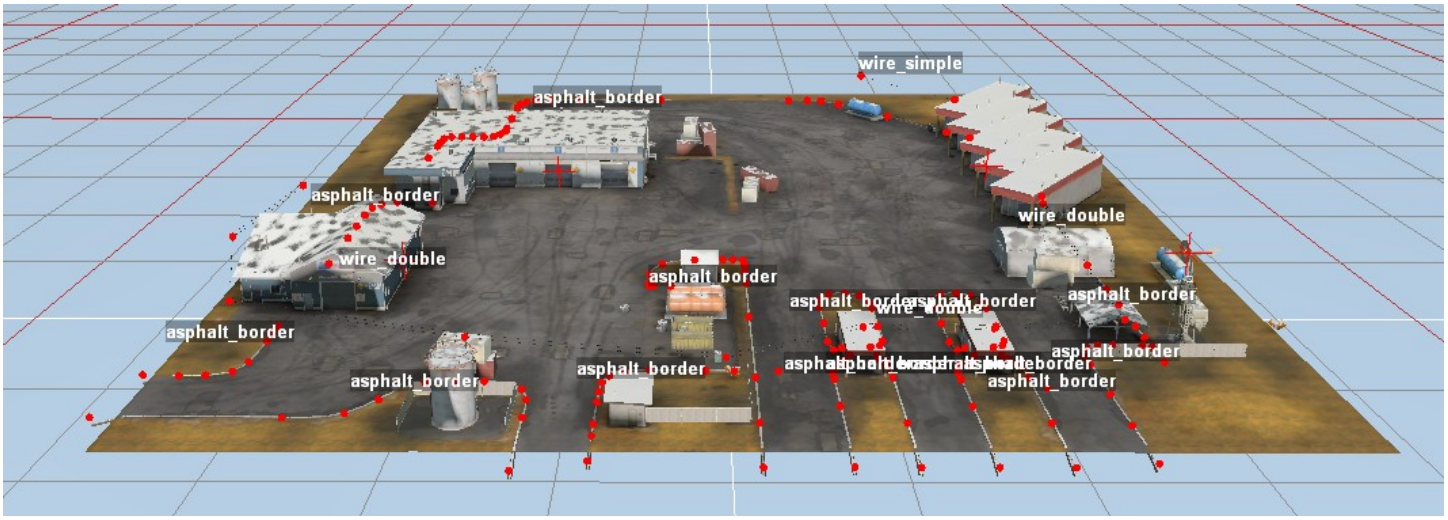
The maps that you have created are listed in the File View panel under Media Files → prebuild. To edit a map, double click it.



Example Maps

Saber has not published any of their maps in an editable format. However, their maps use a number of shared map sections that Saber has kindly made available for modders to use. File → Unpack references unpacks these map sections from the installation directory and puts them in the Media/prebuild directory where they can be read into the SnowRunner Editor.

Bug: Unpack references takes a very long time (over a minute on my system) with no indication of progress. Be patient, or watch the files as they appear in the Media/prebuild directory.



How to Use This Book

Sources of Information

This book is designed so that you can read it sraight through from beginning to end. Each section builds on the information from previous sections to form a complete picture without too many instances of “I’ll tell you later”. However, you can also just start experimenting with the editor while reading only the sections of this book that strike your interest. To accommodate this, I’ve done my best to label all sections so that you can easily find the section you need in the table of contents.

I should point out that this is not your only source of information regarding SnowRunner map making. There are various other guides on the web in either text or video form, and Saber have themselves published a reasonably complete guide which you can find installed at a path similar to this:

```
C:\Program Files (x86)\Steam\SteamApps\common\SnowRunner\Sources\BinEditor\Guides\
SnowRunner_Editor_Guide.pdf
```

Advantages of the Big Book of SnowRunner Map Making:

- Faster “quick start” to editing and testing.
- Includes topics not included in the Saber guide.
- Covers all topics in exhaustive detail.

Advantages of the Saber guide:

- More likely to stay up to date as the game and editor are updated.
- Not as exhausting to read.

By the way, if you think this book is useful, please give it a “thumbs up” wherever you found it. Those thumbs up will help more people find and enjoy this book. More educated modders means better mods!

Is This Book Up To Date?

There are a lot of websites that will “helpfully” copy this book for their readers, but won’t keep it up to date. The official release location is on GitHub: https://fred-rum.github.io/bbsmm/BBSMM.pdf

The title page of this book has the date on which I last edited it. However, Saber doesn’t notify me when they change something, so it’s quite possible for me to edit the book after a change and yet not include it.

I’ll make announcements in the Steam forums and Focus forums when I make major updates. If you notice anything missing or wrong, let me know in the comments to these announcements.

Editor and Game Bugs

No mod editor will ever be designed to be as consumer friendly as the game itself for a number of reasons:

- Only a fraction of game buyers will even try mod editing.
- People who edit mods tend to be more tech savvy and better able to work around any bugs that exist.
- The universe of possibilities is much larger when editing mods than when playing a game. That makes it harder to test every corner of the design and iron out the bugs.

In most cases, a mod editor is tested only to the extent that it was used to make the game levels themselves. If a feature wasn't used in the game, then it probably wasn't tested in the editor.

If it sounds like I'm making excuses for Saber, I totally am. I have complete respect for Pavel's original Editor and for all of the quirks that Saber has smoothed out of it. But Saber has also added a pile of new features which come with their own bugs.

For any insufficiently tested feature, there can be a bug in the Editor (so the Editor can't edit that feature properly) or a bug in the game (so that the game doesn't work correctly with that feature). I've chosen to highlight either type of bug as follows:

> **Bug:** This is an example bug warning. I'll explain when the bug occurs and how to recognize it, how to avoid it (if possible), and how to fix any problems it causes (if possibl).

I generally put each bug warning immediately after the related text that describes how to use a feature. So if you see a bug warning, read it!

And if Saber wants to use these bug reports while developing the next version of the Editor, I wouldn't mind that, either. :)

Design Tips

I'll admit right now that I am not the best map designer. I'm an engineer at heart, and my artistic eye is limited. Therefore, I'm hesitant to give a lot of design tips in this book. Where I do give a tip, it is often highlighted as follows:

> **Tip:** This is an example design tip. You don't have to do it this way, but this is probably a good way to do it.

Some of these tips I've directly borrowed and expanded from Saber's guide. Other tips come from my experience with making maps and/or wrestling with the Editor for long periods of time. I try to stay away from

totally obvious tips. You already know what a natural world looks like, so I'll leave it up to you to decide whether rocks should go in trees.

If You Made Maps in MudRunner

If you made maps in MudRunner, then the editor should be mostly familiar to you. I recommend reading through the Quick Start and Directories, Files, and Archives sections since those highlight some important differences from MudRunner.

In particular if you already read my Big Book of MudRunner Map Making, you'll find a lot of familiar material in this book. On the other hand, a lot of things have changed from MudRunner, and a lot of things are new. If you have the patience for it, I recommend at least a skim through the entire book to avoid frustrations later.

The MudRunner Editor included the ability to import maps from SpinTires. However, the SnowRunner Editor doesn't include any ability to import maps from SpinTires or MudRunner. You may be able to import selected features of previous maps by manually copying bitmaps and XML sections, although you'll need to hand edit the names of any assets you used such as models, overlays, and distributions. I haven't tried it, so I can't estimate your odds of success.

Directories, Files, and Archives

It is useful to understand the directory structure used by the SnowRunner Editor since you will need to occasionally interact with it.

The primary directories of interest are in `%HOMEPATH%\Documents\My Games\SnowRunner\Media`

Prebuild Directory

The `prebuild` directory contains the input files required to build your maps. For each map, there is a corresponding XML file and a directory. The directory starts with a number of default files that will be updated as you edit your map. More files may also be added, depending on which editor features you use.

In the below example listing, I have organized the files and directories by type:

```
prebuild/
     level_test.xml
     level_test/
            _sounds.xml
            _zones.xml
            _height.dds
            _height_source.png
            _base_tints.tga
            _merge_mud_wetness.tga
            _quickmud.tga
            _snow.tga
            _tint.tga
            _water.tga
            _water_flow.tga
            _water_mud.tga
            mtrl_base_blends.tga
            mtrl_layout.tga
            mud
            subgroups/
            ui/
                  textures/
```

You generally won't ever need to manually edit these files outside the SnowRunner Editor. However, I briefly explain their format here in case you want to do something that is easier to do outside the Editor than inside it.

The XML files are a standard ASCII/UTF format which can be easily read and edited by humans. These files start out as mostly empty stubs, with XML tags ready for new content to be added. The `<map_name>.xml` file contains most of the non-bitmapped data that describes your level, but some information is split off into the `_sounds.xml` and `_zones.xml` files.

The _height.dds file is in an old Microsoft DirectDraw bitmap format. Not many graphics editors can read and edit this format anymore. Interestingly, _height_source.png is **also** in DDS format, despite its suffix. In general, when SnowRunner reads a file containing a bitmap, it ignores the file suffix and determines the format from the file contents. This can be handy sometimes, but it can also be deceptive.

The *.tga files are also bitmaps in an old format created by Truevision. Fortunately in this case, many modern graphics editors (e.g. the GIMP image editor) can read and write this format.

As far as I can tell, the mud file is a collection of images in the DDS format. Because SnowRunner's mud model is highly detailed, the mud file holds data only for terrain blocks where mud is present. So it starts out very small, but it can expand to very large if your map contains a lot of mud.

The subgroups, ui, and textures directories start out empty.

If you delete the prebuild directory or its contents, then you will lose all of your work. Unless you follow the recommendation in the next section...

Create an Archive

Because your mods are saved in your user area and not the game area, your mods are safe even if you uninstall SnowRunner. However, there is always the possibility of losing or corrupting your files either through user error or an editor bug. For this reason, keeping archives of your work is important.

The easiest archival method is to occasionally copy the necessary files and directories from the prebuild directory to another directory. Even better, copy it to a different drive in case you have a drive failure.

Bug: Some files necessary to rebuilding your map are stored in the levels directory. See the next section for details.

Levels Directory

The levels directory contains the files required by the game executable. If you think of the files in the prebuild directory as the "source" files, the files in the levels directory are the "compiled" files.

The levels directory contains a directory corresponding to each of your maps. If you have packed a map, then there is also a ZIP archive that contains your packed mod, ready for uploading to mod.io.

```
levels/
     level_test/
            data.stg
            ...
     level_test.zip
            level_test.pak
            level_test_playstation_4.pak
            level_test_xbox_one.pak
```

The data.stg file contains the majority of the map data, stored in an opaque binary format. The remaining files in the <map_name> directory are more or less ready to be manually edited, but there isn't any reason to do so.

The *.pak files are themselves each an archive of files that can be opened with a supported archive manager such as 7-Zip. These files are essentially just a repackaging of the same files as are in the <map_name> directory.

If you delete the levels directory or its contents, then the SnowRunner Editor will offer to recreate the data as you re-open each map for editing. The ZIP archive will be re-created when you next pack your map.

Bug: Some files necessary to rebuilding your map are stored in the levels directory instead of the prebuild directory. So when you archive the prebuild directory, also make a copy of these files from the levels directory if they are present: content_settings.json, level_desc.json, objective_settings.json, and zone_settings.json (i.e., all *.json files).

Mods Directory

The Mods directory contains the same *.pak files as above for each map, as if they had been downloaded as a mod. These are the files that are read by the SnowRunner game.

If you delete the Mods directory or its contents, the *.pak files will be re-created when you next pack your map.

AppData Directory

SnowRunner keeps track of past activity in a separate area: %appdata%\SnowRunner 2 Editor

E.g.

```
backup/
     level_test/
            ...
CustomColor.xml
LastStates.xml
Log.txt
ViewTerrain.xml
```

The backup directory keeps for each map a number of backups of the XML files only, not the bitmap files. These can be restored through the editor as described in the Menu Bar section on page 29. Or the editor may offer to restore a backup automatically if it detects a problem.

The CustomColor.xml file records any custom colors that were set in the color picker, e.g. in the brush dialog for (Geometry)→(Colorization).

The LastStates.xml file is used to detect when the editor crashes (because the file isn't updated correctly before closing).

The Log.txt file keeps the text from the Output panel for all sessions. (So this file eventually gets quite large.)

The ViewTerrain.xml file keeps for each map the most recent camera position within the editor. If the editor camera gets hopelessly lost, quit the editor, delete this file, and restart.

Config Directory

SnowRunner keeps track of the editor configuration in %HOMEPATH%\Documents\My Games\SnowRunner\base\config\editor.cfg

Currently, however, the only saved information is whether Show Snow By Up Vector is enabled.

SnowRunner Installation Files

The SnowRunner installation directory is mostly inscrutible, but it does contain a few *.pak archive files with useful information in them. This book won't tell you how to modify these files, since any such edits can't be published as a packaged mod for others to enjoy. However, it can be useful to extract the contents of a few of these archives for browsing or searching.

The installed *.pak archives are in a path that looks something like this, depending on whether you have the Epic or Steam version of the game:

```
C:\Program Files\Epic Games\SnowRunner\en_us\preload\paks\client
C:\Program Files (x86)\Steam\SteamApps\common\SnowRunner\preload\paks\client
```

The specific archives that I found interesting are as follows.

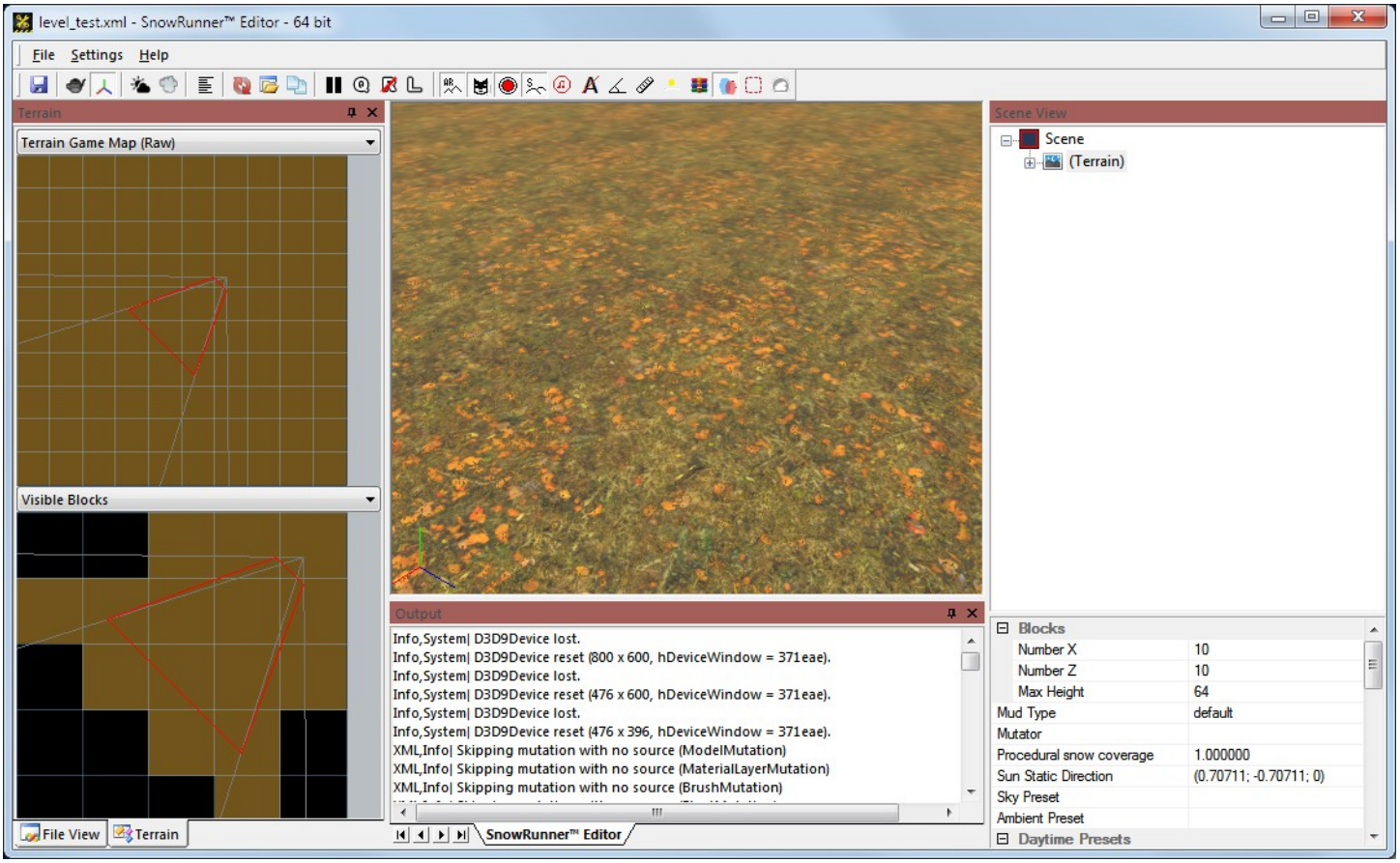
- initial.pak: contains all of the class definitions used in the game as well as localization strings
- shared_sounds.pak: contains all of the ambient sounds used in the game.

The *.pak archives can be extracted with a full-featured archive manager such as 7-Zip.

Various sections of this book will mention when information of interest can be found in these archives.

The Map Editor Window

After creating a new map, the map editor opens. It contains a number of sections that I'll go over briefly here.



Menu Bar

The menu bar at the top of the window has File, Settings, and Help menus.

File Menu

Possible actions in the File menu are as follows.

Close File closes the map and returns to the File View panel. If there are unsaved changes, the Editor will ask if you want to save first.

> **Bug:** Edits in the property panel are not considered to be unsaved changes until the edits are committed by changing the selection, e.g. by left-clicking elsewhere.

Reload File From Backup opens a dialog that allows you to restore XML files from a selected backup. Note that bitmap files are not backed up and cannot be restored from backup. The location of the backups is described in the AppData Directory on page 27. I do not know how often backups are recorded.

Saber used a large number of map reference sections that they have kindly made available for modders to use. The Example Maps section on page 21 describes how to use Unpack references to get these references.

Create default truck mod is used for creating truck mods. Truck mods are not covered in this book.

Save (shortcut: Ctrl-S) saves the map's the named properties of all features from memory to disk. This differs from bitmaps painted with a brush, which are updated on disk as soon as a change is committed.

Exit exits the Editor.

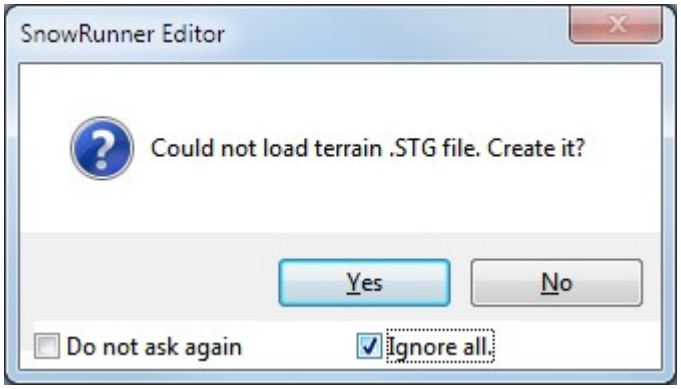
The File menu also includes a list of recented edited maps. Selecting closes the current map and opens the selected map.

Settings Menu

Possible actions in the Settings menu are as follows.

Ignore Warnings disables all warning dialog boxes. The editor prominently warns you when warnings are being ignored.

The same can be accomplished by selecting Ignore All in a dialog warning box.



The Ignore Warnings setting applies across all maps, but it **does not** persist across Editor sessions.

Show Snow By Up Vector shows the snow cover effect for models and plants in snowy areas. This setting applies across all maps and **does** persist across Editor sessions.

Help Menu

The only option in the Help menu is Guides. This opens a Windows Explorer window for the folder containing the official guide PDFs that were installed with the game.

Toolbar

Under the menu bar is a toolbar with a row of buttons for various features. You can hover over each button to get a description of what it does.

Many of the buttons are toggles. Click once to enable, click again to disable. The Editor always starts with the same default mix of enabled and disabled buttons regardless of what was selected in the previous session.

Save (shortcut: Ctrl-S) saves the map's the named properties of all features from memory to disk. This differs from bitmaps painted with a brush, which are updated on disk as soon as a change is committed.

Wireframe (shortcut: Ctrl-W) is a toggle button. When enabled, it redraws everything using only colored outlines for all triangles, instead of filled shapes.

Grid (shortcut: Ctrl-G) is a toggle button. When enabled, it draws a reference grid below the terrain at height 0. The size of each square depends on the zoom level, so it can't be used as a measurement tool, but it is sometimes helpful to keep yourself oriented.

Night is a toggle button. When enabled, it changes the lighting to night mode. This is useful for testing how everything looks in the dark, especially lights.

Fog has is a toggle button with no known effect.

Statistics (shortcut: Ctrl-T) is a toggle button. When enabled, it puts a block of text in the upper left detailing the camera position and orientation and statistics regarding what is being drawn.

Reload Resources reloads most data used by the map (as opposed to data **about** the map). E.g. this includes shapes, texture bitmaps, etc.

Open File (shortcut: Alt-Shift-O) pops up a dialog box listing all files in your Media directory. When you select a file from this list, the Editor attempts to open the file in some manner:

- If you select a map or mesh XML file, it opens in the SnowRunner Editor. These can also be opened via the File View (page 36)
- If you select another XML file, it opens in whatever editor Windows lists as your default (usually some sort of text editor).

- If you select a bitmap/texture file, the Editor displays an error dialog requesting that you specify a texture editor path in the Editor settings. However, although the MudRunner Editor had such a setting, the Snow Runner editor does not, so the error message now just looks silly.
- Finally, if you open some other file type, the Editor displays an error dialog to say that it isn't supported.

Duplicate selected (shortcut: Ctrl-D) makes a copy of the currently selected feature.

Pause is a toggle button. When enabled, animation of models is paused. (E.g. wind_mill_01 is an animated model.)

Quick mode is a toggle button. When enabled, rebuilding the terrain causes the Editor to rebuild large features as usual, but it hides grass, shadows, and other minor features rather than rebuilding them. This can speed up the rebuild quite a bit, especially on large maps.

No references is a toggle button. When enabled, rebuilding the terrain causes the Editor to hide references instead of rebuilding them. References are described in TBD.

Local transform is a toggle button. It is described on page 59.

Enable autorebuild terrain is a toggle button. When enabled, the terrain is rebuilt whenever you commit a change to a (Geometry) feature. Major changes are updated in real time as you edit, but shadows, 3-D grass, and other minor features are updated only after the edits are committed (and only if this button is enabled).

Show all sound domains is TBD.

Hide Strings is a toggle button. When enabled, it hides the labels that normally display near each object.

Show HeightMap Angles is a toggle button. When enabled, it recolors the terrain to show the angle of the terrain at each point. Further information is in the Height Information section on page 117.

Use Ruler is a toggle button. When enabled, a line segment or complex curve can be overlaid on the terrain, and the Editor reports its length. More detail is in TBD.

Terrain brightness pops up a dialog box where the brighness of the terrain can be adjusted. Other map features are not effected. This is particularly useful on snowy maps with a lot of white terrain.

Pack terrain converts the map to a format that can be played in the game and uploaded to mod.io. More detail is in TBD.

Collision Notification is a toggle button. When enabled, dynamic models are highlighted if they collide with the terrain, other models, or individual plants. More detail is on page 76.

Zone settings opens the Zone Settings Editor. More detail begins on page 149. **Caution:** while the Zone Settings Editor is open, the main SnowRunner Editor window is completely unresponsive. Close the Zone Settings Editor to return to the main SnowRunner Editor.

Region settings opens the region settings editor. More detail is in TBD.

Main Panel

Most of your time will be spent in the main panel. This panel is unlabeled, but you can find it centered between the Terrain panel on the left and the Scene View panel on the right. It shows a “camera eye” view of your map.

Navigate in the Main Panel

The main panel shows the 3-D map from the perspective of a virtual camera. To view different parts of the map, the camera can be moved in all three dimensions, it can be panned left and right, and it can be tilted up and down. To prevent disorientation, the camera is never tilted left or right.

A small icon in the lower left of the main view shows the coordinate axes as they appear from the current camera angle. The red line shows the X axis and always points east. The green line shows the Y axis and points up. The blue line shows the Z axis and points north. The SnowRunner coordinate system is described in more detail in the TBDError: Reference source not found section on page Error: Reference source not found.

You can move the camera around as follows:

- left click and drag – Move the camera around whatever terrain you clicked on and rotate the camera to continue pointing at that spot. This is my favorite method for rotating the camera. If the initial click is not on visible terrain, the anchor is attached to the 0-elevation reference plane instead.
- mouse wheel up or down – Push the camera closer to the terrain under the mouse or pull it further away. (Caveat: the most recent click or drag must have been in the main panel.)
- ctrl + left click and drag – Keep the camera pointed the same way while moving it relative to whatever terrain you clicked on. This effectively drags the terrain around under the mouse, and it’s my favorite method for moving the camera across the map.
- shift + left click and drag – Keep the camera fixed in space and rotate it to point in new directions.
- alt + left click and drag – If a feature is selected, move the camera around that feature and rotate the camera to keep that feature in the same spot in the view (or off view). If no feature is selected, the alt modifier is ignored.

It is very easy to get completely lost in the main panel so that no terrain is visible. If the camera is too far away from it, the display engine won't draw it. If that happens, double click in the Terrain panel to recenter your camera over the terrain.

If the camera is below the terrain, the display engine also won't draw it. In that case, double clicking in the Terrain panel will move the camera close to the terrain, but still pointing up at it, so the terrain is not drawn. In that case, left click in the main panel and drag downwards to tilt the camera downward while moving the camera itself upward until the terrain comes into view.

Extremely rarely you may end up with the camera pointed straight up or down, and left click and drag won't rotate it back to a proper angle. The editor tries to prevent this by keeping the camera just off of vertical, but its algorithm can occasionally fail. Shift left click and drag may work when other methods fail. Otherwise, you'll need to quit the editor and edit the file that stores the camera position:

```
%appdata%/SpinTiresEditor/ViewTerrain.xml
```

Find your level in the file and delete its entry. The next time you open the level, the camera will reset to its default position and angle. (This may be underground, but that's more easily fixable.)

Select Targets in the Main Panel

You can select one or more targets in the scene as follows:

- left click – Select the terrain block, model, plant, or spline point under the mouse. Other types of objects cannot be selected in the main panel. This notably includes trucks and zones.

Bug: A model, plant, or spline point cannot be clicked on in the main panel if it is off the edge of the map.

Bug: If you forget the purpose of a zone that you see in the main panel, it takes a lot of searching in the Scene View panel to even find out its ID, and then more searching in the Zone Settings Editor to find out what it's for.

- ctrl + left click – Add the terrain block, model, or plant to the current selection or remove the terrain block from the selection. It is not possible to remove a model or plant from the selection in this manner. It is also not possible to select different types of targets at once. E.g. a model and a plant cannot be in a selection together. Ctrl-clicking a different type of target resets the selection as if it were clicked without the control key.

- double left click – Select the terrain block, model, plant, or spline point under the mouse and move the camera (without rotation) to center the selected item in the scene.

- right click and drag – Select all models within the dragged rectangle. (This only works for models.)

Other Actions in the Main Panel

A right click without a drag brings up a context menu. The context menu is described on page 47.

A middle click (e.g. depressing the moue wheel) displays a copy of the coordinate axes at the mouse position for as long as the middle button is held down. It performs no other action.

Output Panel

The Output panel (below the main panel) shows logging information and error messages from the editor and the embedded display engine. You will refer to it occasionally when things go wrong.

Note that the Output panel will often contains lines saying “D3D9Device lost” and “D3D9 Device reset”. These messages look ominous, but they seem to be part of normal operation, and you can ignore them.

At the bottom of the Output panel are four arrows pointing left and right. I do not know what these do.

The angled tab at the bottom of the Output panel indicates that it is showing the main log from the SnowRunner Editor. On occasion, another tool within the Editor may create a separate tab in the Output panel in order to display its log. You can then click on the angled tabs to switch between logs.

Terrain Panel

The Terrain panel on the left includes two subpanels looking into the scene from a top-down view.

By default, the upper panel shows a view of the entire map. A truncated triangle shows the field of view of the main panel’s camera. If the map is not square, it is stretched into a square for display in the terrain panel.

The terrain panel only shows the colors that are painted on the terrain itself. So, for example, it will show a road overlay, but not a house model. If the house model has shadows, the terrain panel will show one type of shadow but not the other.

A dropdown menu above the upper panel changes it to show many other map textures used by the display engine. I don’t find these to be very useful. Once you’re done looking around, flip it back to the default Terrain Game Map (Raw) at the end of the dropdown list.

The lower panel is similar to the top one, but by default it zooms in on the terrain blocks that are visible to the camera. (I’ll describe terrain blocks later in the guide.) A truncated triangle again shows the field of view of the main panel’s camera. Only those terrain blocks within this view (or very near it) are drawn in the lower panel.

Another dropdown menu allows the lower panel to be switched to show only the Selected Blocks. I find the Visible Blocks default to be the more useful view for the lower panel.

Changes made from the dropdown menus in the Terrain panel do not persist across Editor sessions.

Below the two map panels is a block of text that gives various information about the selected terrain blocks. The information isn't particularly useful, though, so I never pay any attention to it.

Navigate in the Terrain Panel

Double click a terrain block in either part of the terrain panel to fly the camera in the main panel to that block. (This also selects the block.)

It is tempting to rotate the scroll wheel up and down within the terrain panel to zoom in and out, and it sort of works. But because the mouse is positioned outside the main panel, the target point for the zoom is off the edge of the view, so zooming in isn't very effective. Move the mouse back to the main panel before zooming.

Select Targets in the Terrain Panel

Left click in either part of the terrain panel to select a terrain block. Hold the Control key while you left click to add or remove a terrain block from the current selection.

File View Tab

At the bottom of the Terrain panel are two tabs which you can use to switch between the Terrain panel and the File View panel.

The File View panel lists selected files and directories in the Media directory.

Double-click any map's XML file in the prebuild folder to open that map. Only one map can be open at a time, so if you have a map open already, the editor will prompt you to save it before opening a new one.

Double-click a folder to expand or contract its level of hierarchy.

Right-click the prebuild folder or any sub-folder for a context menu that allows you to create a new directory (New Folder...) or a new map (New Terrain...) in that folder.

Remember that you can open a recently edited map by selecting it from the File menu, but you can open any map at all by double clicking its XML file under the prebuild directory for its mod.

The File View tab has its own toolbar. You can hover over each button to get a description of what it does.

The Update button refreshes the file list.

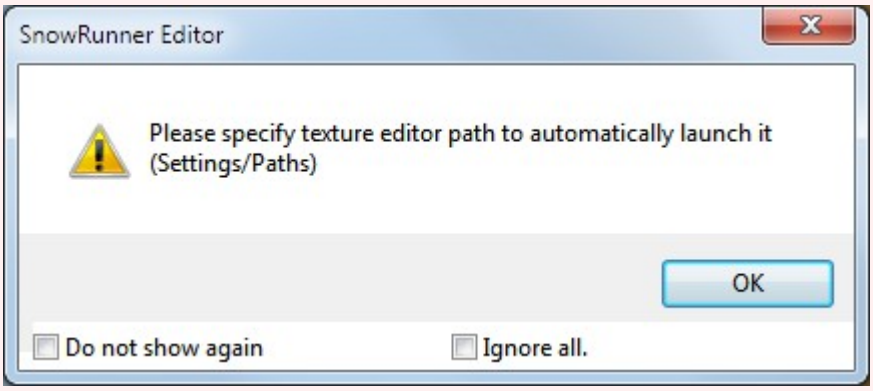
The Show in Explorer button opens the selected folder in Windows Explorer.

The Find References button finds all references within the Media directory to the selected XML file. For example, it can find all maps that use a reference map, or it can find all references to a custom asset. The Find References button remains depressed while the search is ongoing, and it releases when the search is complete. The search results are displayed in the Output panel in a new tab called Find References Results. Be aware that the X button in the corner of this tab closes the entire Output panel, not just the tab.

The Show All Files button is a toggle. When enabled, it shows all files and folders in the Media directory. In the default disabled state, it shows only the top-level XML file for each map, and folders are shown only if they contain at least one top-level XML file.

When all files are shown, double-click an XML file to open it in your default XML editor (e.g. Notepad++).

Bug: When all files are shown, if you double-click a bitmap file, the Editor complains that no texture editor has been configured. This is a holdover from the MudRunner Editor; there is no way to specify the path to a texture editor in the SnowRunner Editor.



Controls for Hierarchical Lists

The File View tab, the Scene View panel, and the property panel all display hierarchical lists that share a number of controls in common.

A few common controls are performed by the mouse:

- Left click item selects that item.
- Left click on a + to the left of a folder (in the File View tab) or hierarchical container (in the Scene View panel or property panel) expands that container, showing its contents.
- Left click on a - to the left of a folder or hierarchical container contracts that container, hiding its contents.

- Double click on a folder (in the File View tab) or hierarchical container (in the property panel) expands or contracts that container. Double click has no effect in the Scene View panel.

- Right click performs special actions specific to the panel in question.

Certain actions can also be performed using the keyboard:

- The up and down arrows move the selection up and down the list.

- The right arrow does different things in different contexts:
  - On an unexpanded folder or container, it expands the container.
  - On an expanded folder or container, it moves to the first item in the container.
  - On a non-container item in the property panel, it moves the selection down to the next item.
  - On a non-container item in the File View tab or Scene View panel, it does nothing.

- The left arrow does different things in different contexts:
  - On an expanded folder or container, it hides the items in the container.
  - On an unexpanded folder or container, it moves to the container of the current item (up one level in the hierarchy).
  - On a non-container item in the property panel, it moves the selection up to the previous item.
  - On a non-container item in the File View tab or Scene View panel, it does nothing.

- The Home key moves the selection to the top of the list.

- The End key moves the selection to the bottom of the list.

- The Page Down key moves the selection down by approximately one page.

- The Page Up key moves the selection up by approximately one page.

In all cases, the panel automatically scrolls to keep the selected item in view.

Scene View Panel

The Scene View panel on the right contains a categorized list of all the features on the map.

Select Targets in the Scene View Panel

Left click on a feature or category selects it.

Ctrl + left click on a feature in the same category as previous selections adds that feature to the selection. Features in different categories cannot be selected at the same time.

Ctrl + left click on a selected feature removes it from the selection.

Shift + left click on a feature in the same category as a previous selection adds all features from the initial selection to the clicked feature to the selection.

Bug: Shift click does not correctly highlight the selected items, although the proper items are secretly selected. If you're not sure, shift click the same item again, and all items should highlight correctly.

Bug: Shift click only works if the current selection was selected with a simple left click. So shift clicking on different items to adjust the selection range won't work after the first one.

Right click on a feature or category brings up a context menu. The context menu is described on page 47.

Double click has no function the Scene View panel.

Other controls are summarized in the Controls for Hierarchical Lists section above.

Introduction to Features

This book groups features into three types:

- A zone, model, plant, reference, truck, sound, or sounddomain is a "simple object". It has a single location and orientation (even if it has no visual presence).
- An overlay or river is a "spline object". It is a curve that links a number of points, each with a position but not an orientation.
- Everything else is just a feature, not an object. This includes geometry features, PbrMaterials, and distributions.

Simple objects and spline objects are together referred to simply as objects.

Each feature may have one or more named properties and/or one or more associated bitmaps.

A **named property** has a name and a value, and it is displayed in the properties panel at the bottom of the Scene View. Named properties are recorded in your map's XML file in the prebuild directory. Unless otherwise stated, references to "properties" in this guide refer to named properties. Named properties can be edited in the property panel, described below.

A **bitmap** is painted onto the terrain using a brush. Each pixel in the bitmap corresponds to a local region of the terrain and is displayed in the corresponding section of the main panel (which itself has many pixels on the screen). Each bitmap is recorded in a file in your map's directory within the prebuild directory. The name of the bitmap file is implicit for some features, and it is a named property for other features. Using brushes is described in more detail in TBD.

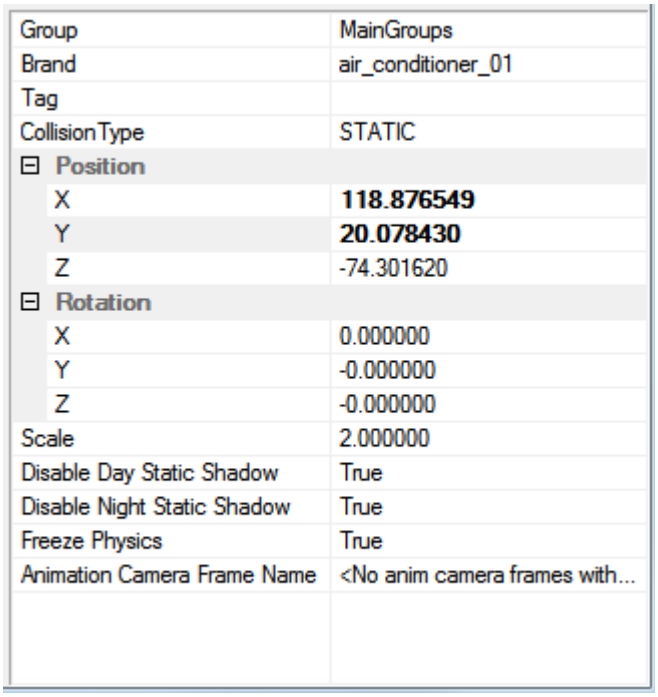
Property Panel

At the bottom of the Scene View panel is an unlabeled sub-panel that I call the property panel. It shows the named properties of the item selected in the main Scene View panel and allows those properties to be edited where applicable.

Most named properties can be edited directly in the properties panel, either by directly typing a new value or by some other means of selecting a new value. The editing conventions are different depending on the property type. Editing methods for each property type are described in the sections below.

When a property is edited, the changed values are displayed with bold text in the properties panel. The text remains bold until you change the selection or until you change a value back to its previous value.

If the changed values result in a visual change to the feature, it is immediately redrawn with its new appearance.



A simple object's position, orientation, and scale properties or a spline object's point positions can also be modified by manipulating the object's interface in the main panel. The new values are updated in the property panel when you release the mouse button, but unlike manually entered values, the new values are not displayed in bold.

When selecting named properties, the Tab key switches between the left column to the right column, and the enter key returns to the left column. If the selection is at the top of a set of related properties (e.g. Position or Rotation in the figure above), the Enter key expands or collapses the set.

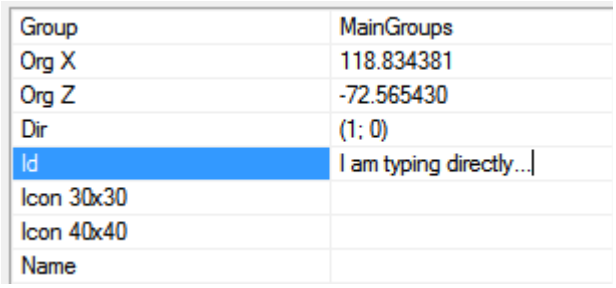
Bug: For an editable text value, pressing the Tab key confirms the change (and updates the main panel if appropriate), but it does not return to the left column. Instead, neither column is selected. Rather than use the Tab key, I recommend that you use the Enter key to confirm the change and return to the left column.

A few named properties are for display only or otherwise cannot be edited once the feature is created (e.g. the map size).

Other controls for navigating through the list of properties are summarized in the Controls for Hierarchical Lists section on page 37.

Edit a Property by Typing Directly

For some properties, you can directly type a new value. Click in either the left column (name) or right column (value) of the desired property in the property panel and begin typing. If you clicked in the left column, your new text entirely replaces the old value. If you clicked in the right column, your new text is inserted where you clicked, and you can generally edit the old or new text. Pressing tab or enter or left clicking somewhere else completes the data entry.

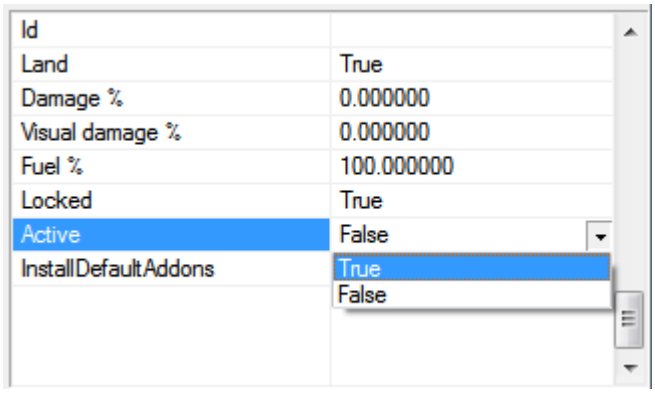


Many named properties will look like they have a value that you can type directly, but when you click the property some other interface appears. These other interfaces are described below.

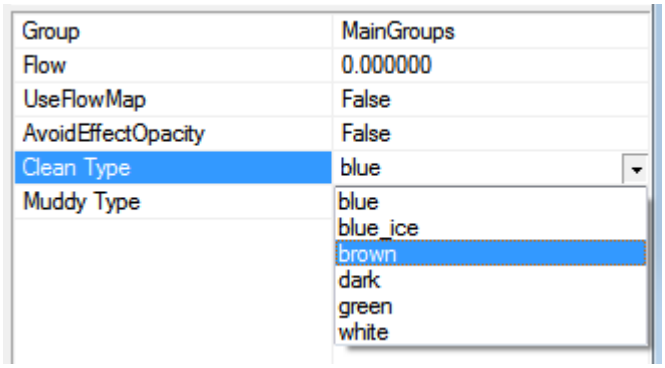
Edit a Property Using a Dropdown Menu

For a property with a dropdown menu, a small downward-pointing arrow appears when you select the property. Click the arrow to get a menu of values that you can choose among. You can also click at the right end of the property where the arrow will appear to select the property and invoke the dropdown menu with one click.

Properties that can be either True or False are the most common use of the dropdown menu.



When the property is selected, you can also type the first letter of the desired value to immediately select that value. If there are multiple values that start with the same letter, typing the letter cycles among those values. Picking a dropdown value by typing is not sensitive to case, so e.g. you can press t for True and f for False.



Finally, you can double-click the property to cycle among the values. This is particularly handy for switching between True and False. You have to select the property before you double click in the right column. You can double-click the left column without selecting the property first.

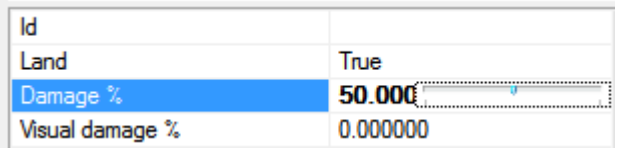
(Now aren't you glad you're reading this book? That's the kind of secret knowledge that could shave **seconds** from your map editing time.)

Edit a Property Using a Slider

For a property with a defined range of numerical values, clicking in the right column of the property causes a slider to appear to the right of the property value. The slider can be manipulated using the mouse or keyboard:

- Drag the slider indicator to the right or left to quickly increase or decrease the value.
- Click on the slider to the right of the indicator or press the Page Down key to increase the value by a large step.
- Click on the slider to the left of the indicator or press the Page Up key to decrease the value by a large step.
- Press the right arrow key to increase the value by a small step.
- Press the left arrow key to increase the value by a small step.
- Press the End key to set the value to the maximum value.
- Press the Home key to set the value to the minimum value.

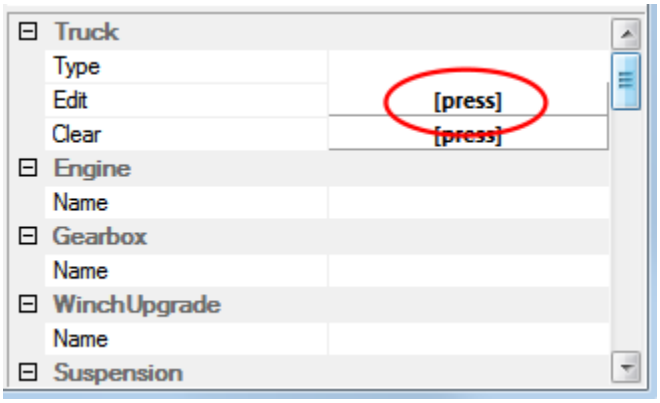
There is no way to type a value directly into a slider property.



Edit a Property That Selects an Asset

For properties that chose among things that can be represented visually, the editor uses a Select Asset window.

For many objects, the Select Asset window pops up when the object is created in order to choose its type. For other objects, an asset can be selected at any time. For these, the editor displays the name of the current asset as an uneditable value, but it adds a second pseudo-property that allows you to edit the asset property. The name of the pseudo-property is Edit, and its value displays as [press]. Left click the [press] text to open the Select Asset window. (The text acts like a button, even though it doesn't look like one.)



Bug: The first time you open a Select Asset window for a particular asset time, the editor may lock up for 5 or more seconds as it creates icons for all possible assets of that type. It will usually be speedy for subsequent asset selections of that type. Occasionally, it will be slow again as it checks for any changed assets, although never as slow as that first time you used it.

Bug: In most programs, when you click in the scrollbar above or below the current scroll position, the window scrolls up or down by one page. However, in the Select Asset window, it instead scrolls by a fixed percentage of the total number of entries. This makes it hard for your eye to predict where to look for assets that have newly scrolled into view. And for assets such as truck add-ons, the fixed scroll percentage scrolls so far that it actually skips over many of the available assets. When you click the up or down arrow buttons at the ends of the scrollbar, it scrolls by a smaller fixed percentage. This has a similar bug, but at least it scolls by a small enough percentage that it shouldn't skip over any assets (unless you've added enough custom assets to greatly expand the list).

To select an asset, left click it in the Select Asset window and then click OK or press enter. Or double-click an asset to choose it and immediately confirm your choice.

Click Cancel or press the escape key to cancel the asset selection. If the Select Asset window was opened during the creation of a new object, the object creation is canceled. If you click OK or press enter when no asset is selected, it is the same as clicking Cancel.

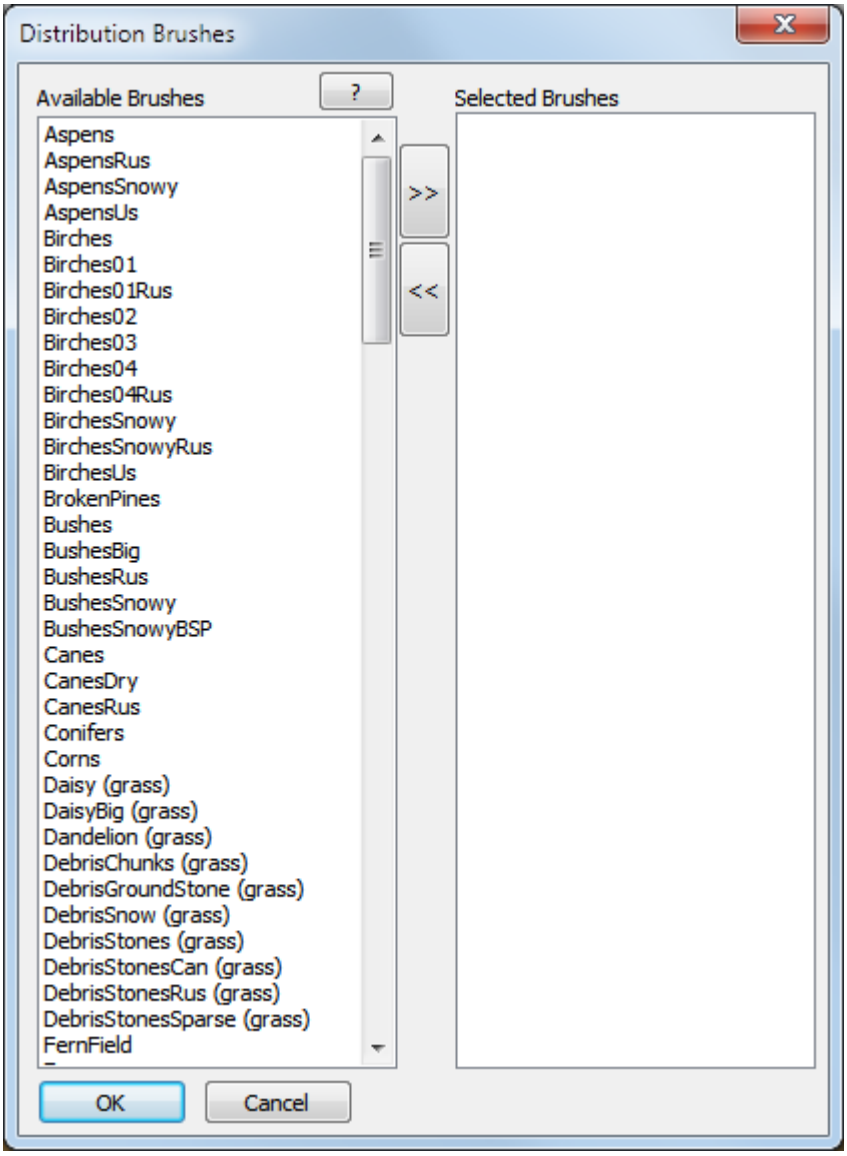
For properties where an asset is not required, you can click [press] next to the Clear pseudo-property in order to clear the selected asset.

Bug: The Clear property is shown for many properties where an asset is required. For these properties, clicking [press] next to Clear does nothing.

You can filter the list of assets by typing in the blank panel near the top of the dialog box. Only assets that include your text somewhere in the name are shown in the dialog box.

Edit a Property that Selects Brushes

The appearance of some features depends on which brushes are attached to the feature. Zero, one, or many brushes can be selected using the Select Brush window. The editor displays the names of the current brushes as the value of a List property which can be modified using the following Edit pseudo-property. The value for the pseudo-property displays as [press]. Left click the [press] text to open the brush selection window.



Move a brush from the left Available Brushes column to the right Selected Brushes column by clicking its name and then the >> button. Move it back by selecting it from the right column and clicking <<. Or quickly change a brush from one side to the other by double clicking it in either column.

When you are done, click OK or press enter to confirm your selection of brushes. Or click Cancel or press the escape key to cancel your changes.

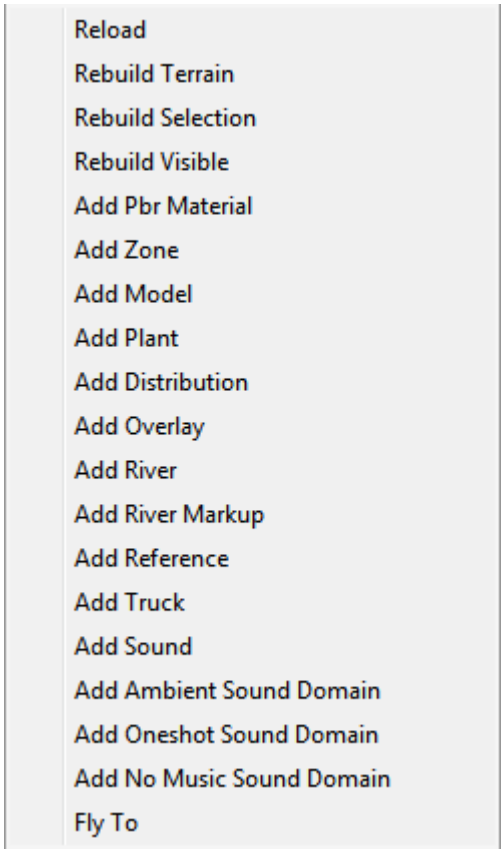
The Context Menu

The SnowRunner Editor has a context menu that can be invoked by right clicking in a couple of different places. The context menu always includes a number of common entries, and more entries are added for the context of the chosen feature.

In most cases, if you right click anywhere in the main panel, the context menu appears for the currently selected feature. However, if a feature with a bitmap property is selected, right click in the main panel operates the paint brush.

If you right click on a feature in the Scene View panel, the context menu appears for that feature. If you close the context menu without selecting an item from the menu, the feature is selected as if you left-clicked on it.

The rest of this section describes the common context menu entries, plus a few entries that are used by a number of features.



Bug: The context menu only works in the main panel when something is selected at the (Terrain) level or below. Right click in the main panel is ignored if the top-level Scene is selected or if nothing in the Scene View panel is selected (so the property panel is blank). This most commonly occurs after the map is saved, after the terrain is rebuilt, or after a feature is deleted. My habit is that if right click doesn't work in the main panel, I just left click and then right click again.

Reload

Reload discards all unsaved changes and reloads the map from disk. The editor asks for confirmation before reloading.

Rebuild Terrain

Rebuild Terrain redraws everything in the map. For a large, complex map, this can take a very long time.

If the editor has taken shortcuts in its iterative drawing (such as not redrawing shadows), Rebuild Terrain will correct the view. It also occasionally reshapes some features such as terrain height and rivers that need a wider context to correctly join together.

Each of the Rebuild actions first commits any property edits in progress, and it clears any current feature selection.

Rebuild Selection

Rebuild Selection redraws everything in the selected terrain blocks. This takes some shortcuts, but not as many as the editor takes in its iterative drawing. Because selecting multiple terrain blocks is a pain, I find this option to be not very useful.

Rebuild Visible

Rebuild Visible redraws all visible terrain blocks. This takes some shortcuts, but not as many as the editor takes in its iterative drawing. This is occasionally useful for updating the view of certain features, although I can never remember what. I'll sometimes try it, and then only choose the slow Rebuild Terrain option if necessary.

Add <Feature>

Add <Feature> adds the named feature to the map and selects it. The feature's position is initialized to the terrain in the center of the main panel. If there is no terrain visible at that spot, the feature is added at a corresponding point off the map.

If a map feature is selected, the feature's category will often add a context menu item such as "Trucks - Add Truck". This is the same as the generic "Add Truck" menu item, but is a little closer to the mouse when the context menu is invoked.

Fly To

Fly To moves the camera (without rotation) to a position in which all selected terrain blocks are visible. This option is only available when a feature is not selected.

<Feature> - Fly To

<Feature> - Fly To moves the camera (without rotation) to point at the selected feature. If multiple features are selected, the menu item is Selected - Fly To, and it moves the camera such that all selected features are visible. If the selected feature is not an object, <Feature> - Fly To moves the camera to the selected terrain block like Fly To above.

You can also double click a model or plant in the main panel to select and fly to it.

<Feature> - Delete

<Feature> - Delete deletes the chosen feature and clears the current selection. The secret keyboard shortcut is the Delete key.

> **Bug:** After a feature is deleted, nothing is selected. If you then press the Delete key to delete another feature, a confirmation dialog pops up anyway. If you click Yes in the confirmation, the Editor crashes.

If multiple features are selected, the menu item is Selected – Delete All, and it deletes all of the selected features.

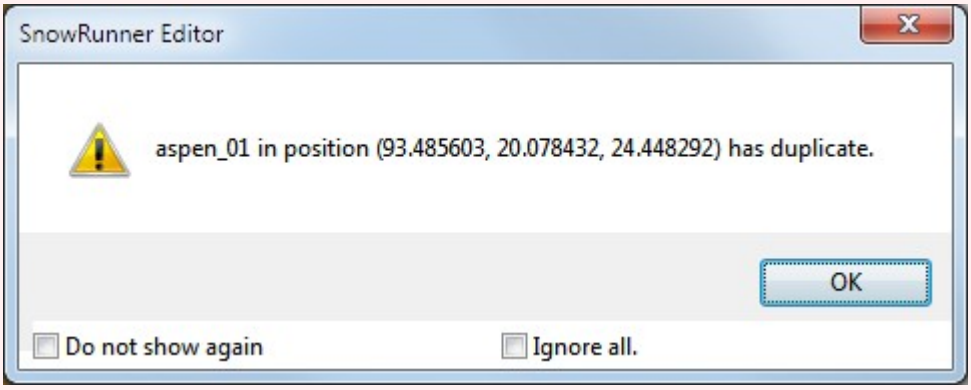
There is a secret undo function for delete. Pressing Ctrl-Z restores the deleted feature. Ctrl-Y redoes what was undone, so in this case it re-deletes the feature. There is only one level of undo.

> **Bug:** Ctrl-Z restores only the position and rotation and a few other properties of the deleted feature. Most properties of the feature are lost. E.g. model properties are restored perfectly, but all useful truck properties are lost.

<Object> - Duplicate

<Object> - Duplicate makes a copy of an object and selects the copy. This copies all of object's properties to the new object and selects it. The object can also be duplicated with a shortcut key, ctrl-D.

Bug: Duplicating a model or plant displays a warning that the object is duplicated. This dialog may be repeated even when you delete the duplicate object. The checkbox for Do not show again only works if the next warning dialog has the exact same text, so it won't supress the warning when duplicating a different object.



You might want to cultivate a habit of pressing Ctrl-D to duplicate and then Enter to dismiss the dialog for models and plants. Note, however, that the warning dialog does not appear for other object types.

If multiple features are selected, the menu item is Selected – Duplicate All, and it deletes all of the selected features.

Bug: Duplicating multiple features at once may cause an extra one to spawn every time the duplicate warning dialog pops up with no obvious way to make it stop.

<Category> - Add <Feature>

<Category> - Add <Feature> is the same as the generic Add <Feature> menu item, but is a little closer to the mouse when the context menu is invoked on another feature of the same type.

If the selection is within a group, there is also a menu item for <group name> - Add <Feature>, and it adds the new feature directly to the group.

<Category> - Add Group

<Category> - Add Group creates a new hierarchical container under the feature category. Groups allow some amount of organization in maps which may contain a very large number of features.

Groups can only be created directly within a category. They cannot span categories, and they cannot be nested.

A new group is assigned a random numeric ID. This can be changed to any desired name.

Features within a group are recorded in a separate XML file which is saved in the map's subgroups directory. This XML file can be copied into another map's subgroups directory to effectively copy all of the grouped features to that map. Their locations from the original map may not be appropriate for the new map, but it is easy to select all of the grouped features in the new map and move them together.

<Category> - Collapse All

<Category> - Collapse All contracts the category and all groups within the category so that their contents are hidden from the list.

If the selection is within a group, there is also a menu item for <group name> - Collapse All, and it contracts only that group.

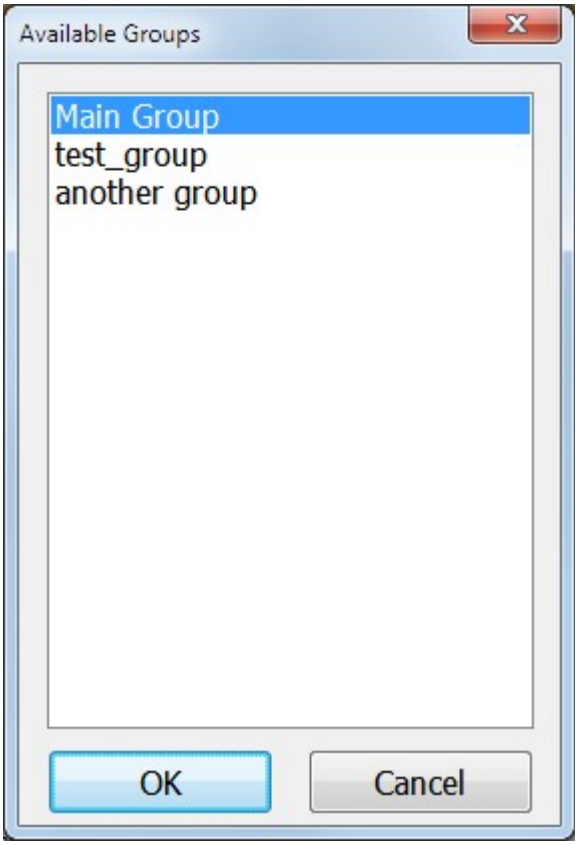
<Category> - Expand All

<Category> - Expand All expands the category and all groups within the category so that their contents are shown in the list.

If the selection is within a group, there is also a menu item for <group name> - Expand All, and it expands only that group.

<Category> - Move to other group

<Category> - Move to other group or <Group> - Move to other group pops up a dialog to allow the selected feature(s) to be moved to another group or back into the main category list (called Main Group in the dialog).

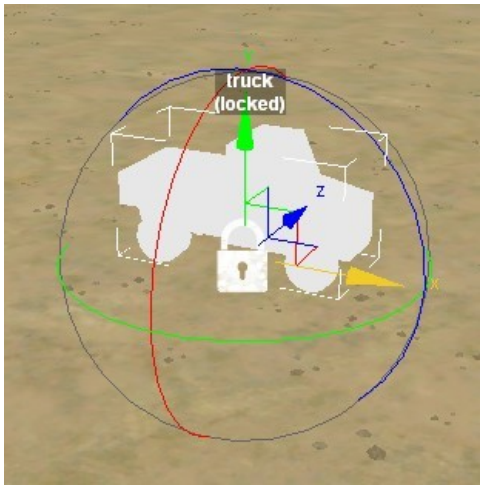


Trucks and Trailers

The minimum requirement to publish a map is that it has a starting truck, so we'll begin with that.

Right click anywhere in the main view or on any feature in the Scene View and select Add Truck. (If it doesn't work, review the context menu and its caveats and bugs starting on page 47.) This same option is used to add a trailer. Page 59 describes how to specify what kind of truck or trailer you want.

The new truck comes with some fancy interface elements in the main panel. If these are all jumbled together, zoom in until you can distinguish the individual elements.



The text above the truck indicates its type, which for now is just a generic "truck". The text also indicates that the truck is locked. The white padlock over the truck indicates the same thing. Locked trucks and all other truck details are discussed below.

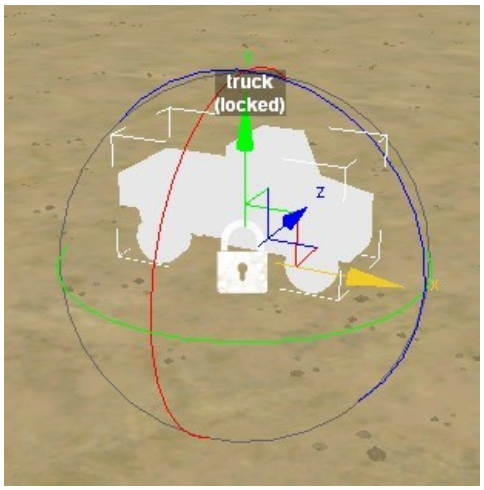
The colored interface elements allow you to move and rotate the truck. These manipulations are described in the next section.

When a truck is created, it is automatically selected for editing. Unfortunately, you cannot click on a truck in the main view to select it. You can only select it by left clicking it in the Scene View.

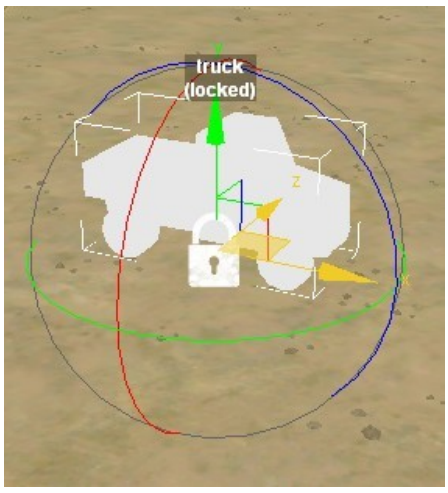
Move and Rotate a Truck

Move and Rotate in the Main Panel

When a truck is selected, the main panel draws on the object a set of colored arrows corresponding to the X/Y/Z coordinate axes. Left click and drag an arrow to move the object back and forth along that axis. By default, a truck is attached to the ground and cannot be raised into the air, so dragging the green Y arrow does nothing.

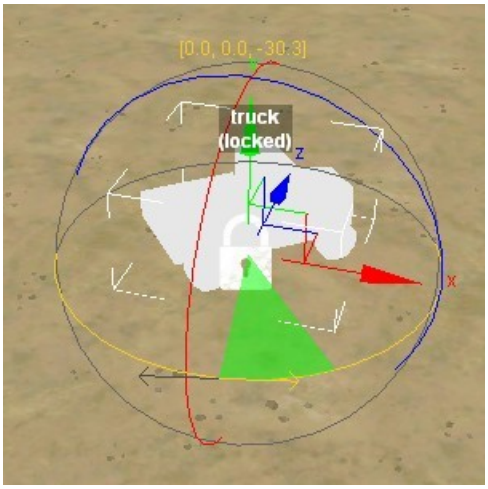


Connecting the bases of the arrows are three squares in 3-D space that designate the XZ plane (ground plane), XY plane, and YZ plane. Dragging one of these squares moves the object around in the 2-D space associated with the plane. The squares will often partially overlap on the screen, which can make it difficult to grab the right one. However, moving in the ground plane is the most generally useful, and the editor always prioritizes that one when there's an overlap. Because a truck is attached to the ground by default, dragging the XY plane or the YZ plane will only move the truck in the X or Z directions, respectively.



The object is also surrounded by a sphere composed of three colored circles. Left click and drag the green circle to rotate the object around its vertical axis (the Y axis). By default, a truck is attached to the ground and cannot tilt away from it, so the red and blue circles do nothing.

When dragging the mouse to rotate an object, your intuition is likely to drag the mouse in a circle, but that is not how the Editor works. When you start dragging, the editor draws an opposing pair of arrows tangent to the circle at the location where you first clicked. Drag the mouse linearly in the direction of one of the arrows to rotate the truck. The more you drag, the more it rotates (even past 360°), although rotation stops when the mouse leaves the main panel.



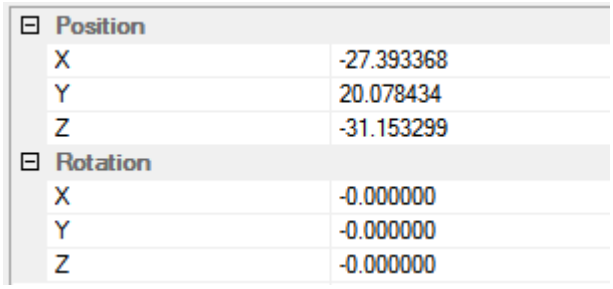
The object is surrounded by a virtual box that designates the boundaries of the truck in the X, Y, and Z axes. The corners of this box are shown with white lines. Because the box has a fixed orientation, its boundaries expand and contract to follow the corners of the object as it rotates.

Move a Truck in the Scene View

A truck's position and orientation can also be edited in the property panel at the bottom of the Scene View.

> Scene.(Terrain).Trucks.*type*: Position.X, Y, Z: numeric, in meters
>
> Specifies the initial location of the truck.
>
> Default: ground level in the center of the main panel.

> Scene.(Terrain).Trucks.*type*: Rotation.X, Y, Z: numeric, in degrees clockwise rotation
>
> Specifies the initial orientation of the truck.
>
> Default: 0, 0, 0; pointing east (+X).



The position of an object is listed in the Position group with three properties X, Y, and Z. The values can be edited directly and are measured in meters.

The orientation of an object is listed in the Rotation group. The X, Y, and Z properties specify the object's rotation around each axis, measured in degrees.

Bug: If you edit the position or rotation properties of a truck by hand, those changes will not be saved. The truck will revert to its previous position and orientation if you pack the map or reload the map. When hand editing a truck's position or rotation, also edit at least one other property, even if you change it right back to its original value. Or use the widgets in the main panel to move and rotate the truck. Changes made by these widgets will save correctly.

Only trucks have this bug. Other types of objects are safe.

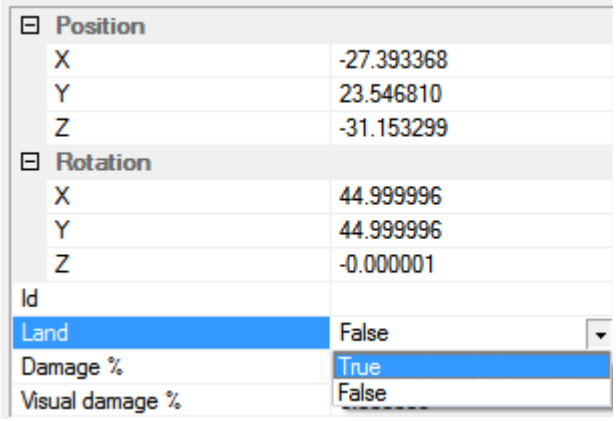
For any given orientation, there are many combinations of X, Y, and Z that achieve the same rotation. You might notice that when you click away from the object and then re-select it, the Editor has changed the Rotation property values. However, they still result in the same orientation.

The coordinate systems for position and rotation are described in TBD.

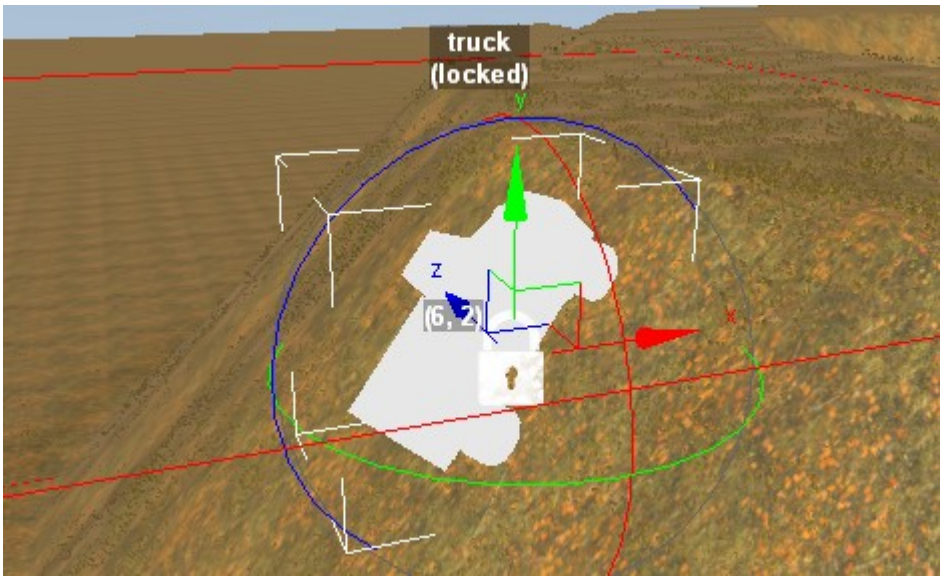
Land Property

> Scene.(Terrain).Trucks.*type*: Land: True/False
>
> Specifies whether the truck's position and orientation are tied to the ground position and orientation.
>
> Default: True.

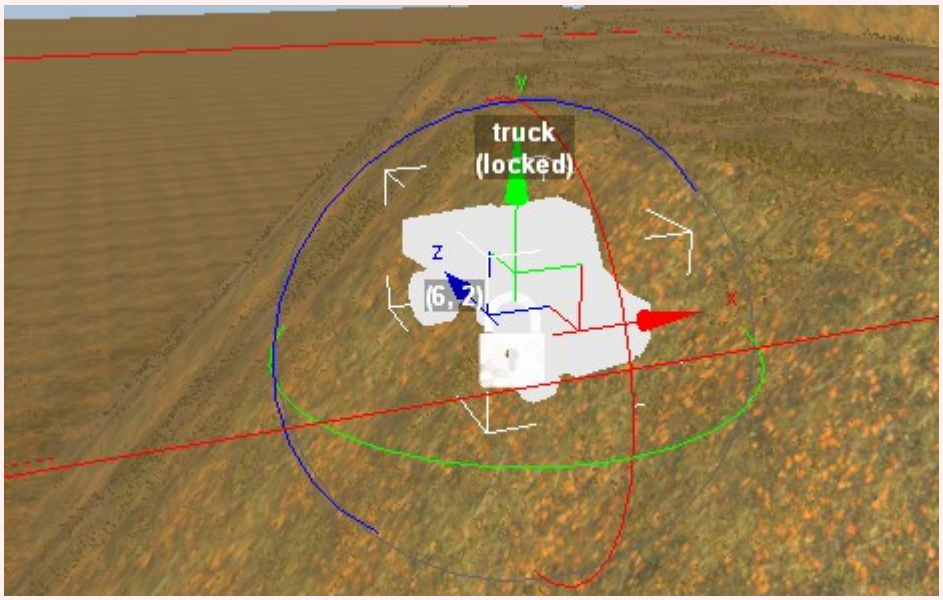
Remember that once you select the True/False value, you can double click it to invert its value. The full set of property panel controls is described on page 40.



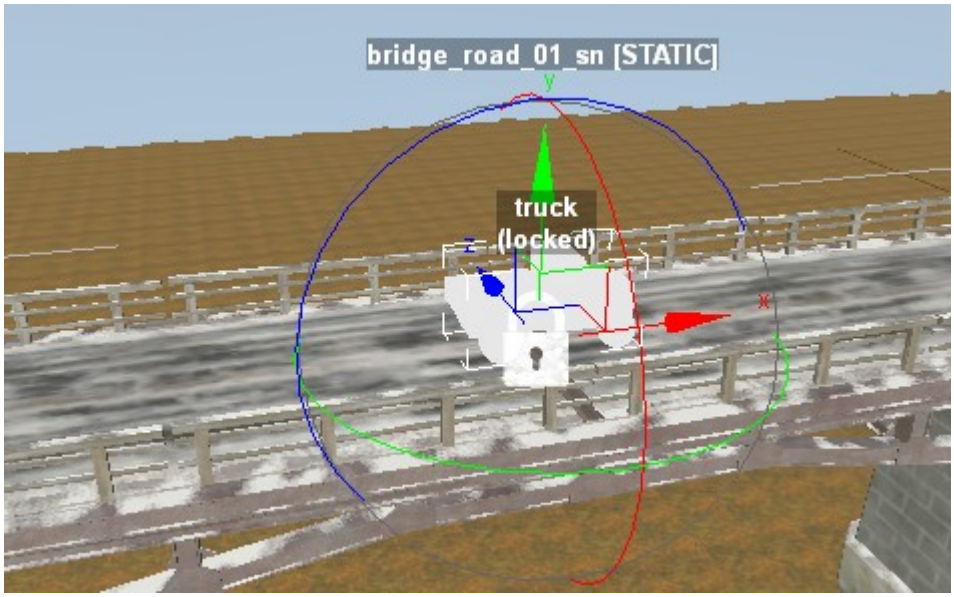
When the Land property is True, the truck is attached to the ground. The Editor tilts the truck so that its wheels make maximum contact with the ground.



Bug: The Editor shows the truck pitched correctly up and down, it does not show it tilted correctly side to side. However, its initial position will be tilted correctly in the game.

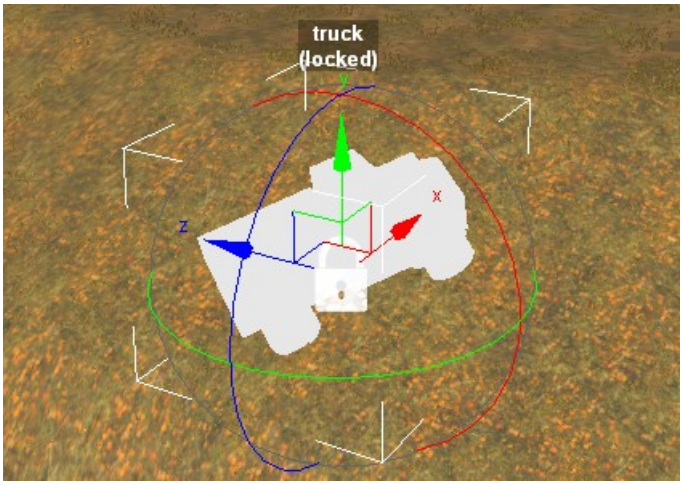


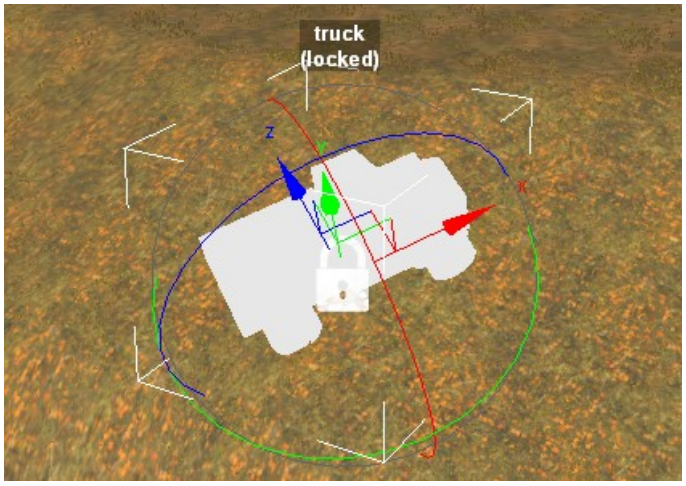
When the Land property is False, the truck can be moved freely. It can be raised into the air, buried in the ground, and tilted to any orientation. In most cases, this will look unnatural and result in strange physics. But it can also be useful, e.g. to place a truck on a bridge.



Local Transform

When moving and rotating a truck, it can be helpful to manipulate it using its local axes rather than the global axes. The Local Transform option in the toolbar can be toggled by clicking it or pressing ctrl-L. Using local axes generally makes setting the pitch and tilt of a truck much more intuitive.





Select Truck Components

Choose the Vehicle Type

> Scene.(Terrain).Trucks.*type*: Truck.Type: asset selection
>
> Specifies the type of truck or trailer.
>
> Default: blank; vehicle does not appear in the game.

The primary property for any truck is its type. When a truck is selected, this property is the top one in the property panel. To select the truck type, left click [press] for the Edit pseudo-property in the Truck group. This brings up a list of trucks from which you can choose your truck type.

Bug: The first time you click here, the editor may lock up for 5 or more seconds as it creates icons for all possible trucks. It will be faster for subsequent uses.

Bug: The Truck group includes a Clear pseudo-property, but clicking [press] next to Clear does nothing.

In the Select Asset dialog window, you can type in the white box near the top to filter the list of trucks as desired. More information about this dialog is on page 44.

Once a truck type has been selected, it appears in the properties panel, and the ghost view of the truck in the main view changes to reflect the selected truck type.

Bug: Although the truck silhouette updates immediately, the name of the truck type isn't displayed correctly in the main panel until you deselect the truck.

Note that the truck list includes trucks from all possible DLC even if you do not own that DLC. If a player tries to play a custom map with one of these trucks without owning the corresponding DLC, the truck will not appear. Keep this in mind when selecting trucks for your map and when advertising your map to others. Trucks are the only features that require DLC ownership. All other features from DLC can be used freely in any map.

You can choose a different truck type by clicking next to Edit again and choosing a new truck. Choosing a new truck type discards the previous selection, but keeps all other properties including trailers and addons.

The Editor uses lowercase truck names that differ slightly from the names used in the game, but it's generally obvious how the names in the Editor correspond with the names in the game.

In the truck selection dialog you can choose to select a trailer instead of a truck. If you select a trailer this way (instead of the usual trailer selection below), the trailer is parked by itself as if it is a truck. Cargo can be loaded on this trailer as described in the Choose Add-ons section below. Trailers that were introduced in DLC can be freely used even without DLC ownership.

By the way, because wheels are a separate asset that are attached to the truck/trailer by the game, they are not shown in the images in the asset selection dialog. They also aren't shown on the silhouette of a truck in the main panel, although they are shown for trailers.

Identify Compatible Truck Parts in the Virtual Garage

Since each truck generally has many available parts, few of which are shared with other trucks, there are a huge number of part names to keep track of. Since each truck can only install a subset of parts, and only in certain combinations, the best way to find compatible truck parts is in the game itself.

The various truck parts and trailers have lowercase names in the Editor that differ from the names used in the normal game. However, when playing in your own custom map or in any proving grounds map, the Garage option opens a virtual garage that allows you to easily discover the lowercase names of compatible parts.

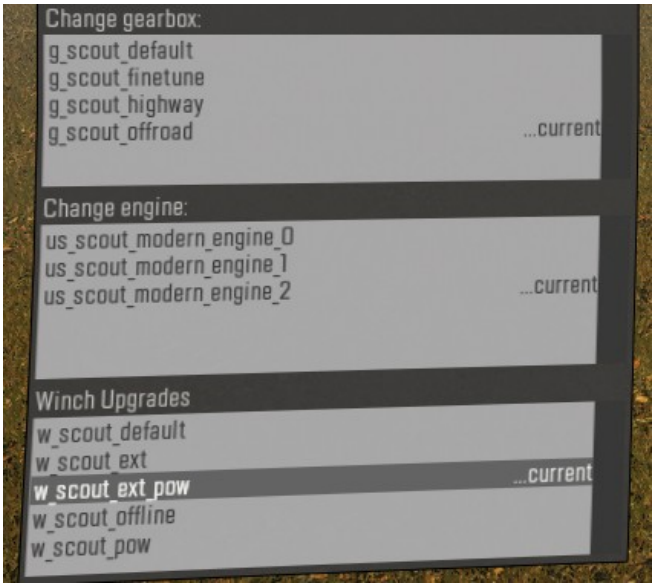
When selecting a truck part or attached trailer, the Editor does not check that it is compatible with the truck or the other parts. If it is not compatible, it will simply be ignored by the game.

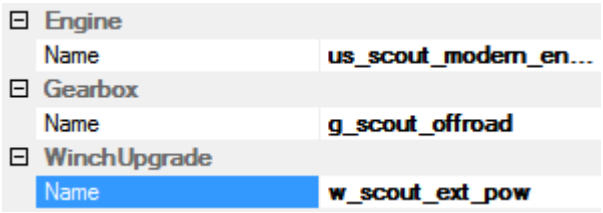
For more information on the dev tools and the virtual garage, see the Dev Tools: Virtual Garage section on page 223.

Choose an Engine, Gearbox, Winch, and Suspension

> Scene.(Terrain).Trucks.*type*: Engine.Name: string
> Scene.(Terrain).Trucks.*type*: Gearbox.Name: string
> Scene.(Terrain).Trucks.*type*: WinchUpgrade.Name: string
> Scene.(Terrain).Trucks.*type*: Suspension.Name: string
>
> Specifies the engine, gearbox, winch, and suspension add-on types.
>
> If Name is blank or invalid, the truck uses a default component.

Enter the name of a compatible truck part into the corresponding property value for that part to be installed in the truck.





Choose Wheels

> Scene.(Terrain).Trucks.*type*: Wheels.Type: string
>
> Specifies the wheel type.
>
> If Type is blank or invalid, the truck uses its default wheels.

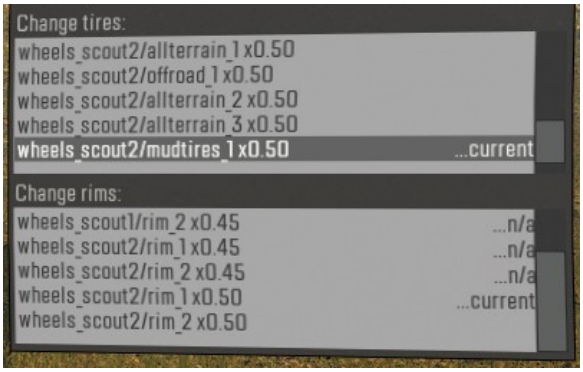
> Scene.(Terrain).Trucks.*type*: Rim.Type: string
>
> Specifies the rim type.
>
> If Type is blank or invalid, the truck uses the default rim that is compatible with the wheels.

> Scene.(Terrain).Trucks.*type*: Tire.Type: string
>
> Specifies the tire type.
>
> If Type is blank or invalid, the truck uses the default tire that is compatible with the wheels.

> Scene.(Terrain).Trucks.*type*: Scale.Type: numeric, in meters
>
> Specifies the wheel diameter.
>
> If Type is 0 or invalid, the truck uses the nearest size that is compatible with the rim and tire types.

Values for Type, Rim, Tire, and Scale can be determined from the tire and rim names shown in the virtual garage.

- The Type value is the text prior to the / in the name of the tire and rim. (It will always be the same for both.)
- The Rim value is the text after the / and before the x in the name of the rim.
- The Tire value is the text after the / and before the x in the name of the tire.
- The Scale value is the number after the x in the name of the tire and rim. (It is always the same for both.)



Choose a Color

> Scene.(Terrain).Trucks.*type*: Customization.PresetId: integer
>
> Specifies the color scheme for the truck.
>
> If PresetId is –1 or invalid, the truck uses its default color scheme.

You can test color schemes in the Customization section of the virtual garage. The PresetId value is the number after the word Preset.

Note that you can deselect all choices in the virtual garage to return to the default color scheme, which is different than all presets. The default PresetId property value of –1 chooses this color scheme.

The standard color schemes are numbered 0 – 31. Many of these are a single solid color, but some are multi-toned. Experiment in the virtual garage ro find a pleasing scheme.

Special skins and vinyl wraps are numbered 32 and higher. Note that they are typically available only with DLC installed, and only for certain trucks.

Choose a Trailer

> Scene.(Terrain).Trucks.*type*: Trailer.Type: asset selection
>
> Specifies the type of trailer to attach to the truck.
>
> Default: blank; trailer is not attached in the game.

Selecting a trailer is performed much like selecting a truck, this time using the pseudo-properties in the Trailer section.

Bug: The game **cannot initialize a truck with an attached trailer**, so this property is useless.

If you select a trailer that is not compatible with the truck type, the editor simply records the choice without any error notification. Make sure to find a compatible trailer in the virtual garage.

The trailer is not drawn in the main view. You will have to use your intuition about its length to ensure that it doesn't overlap any other objects.

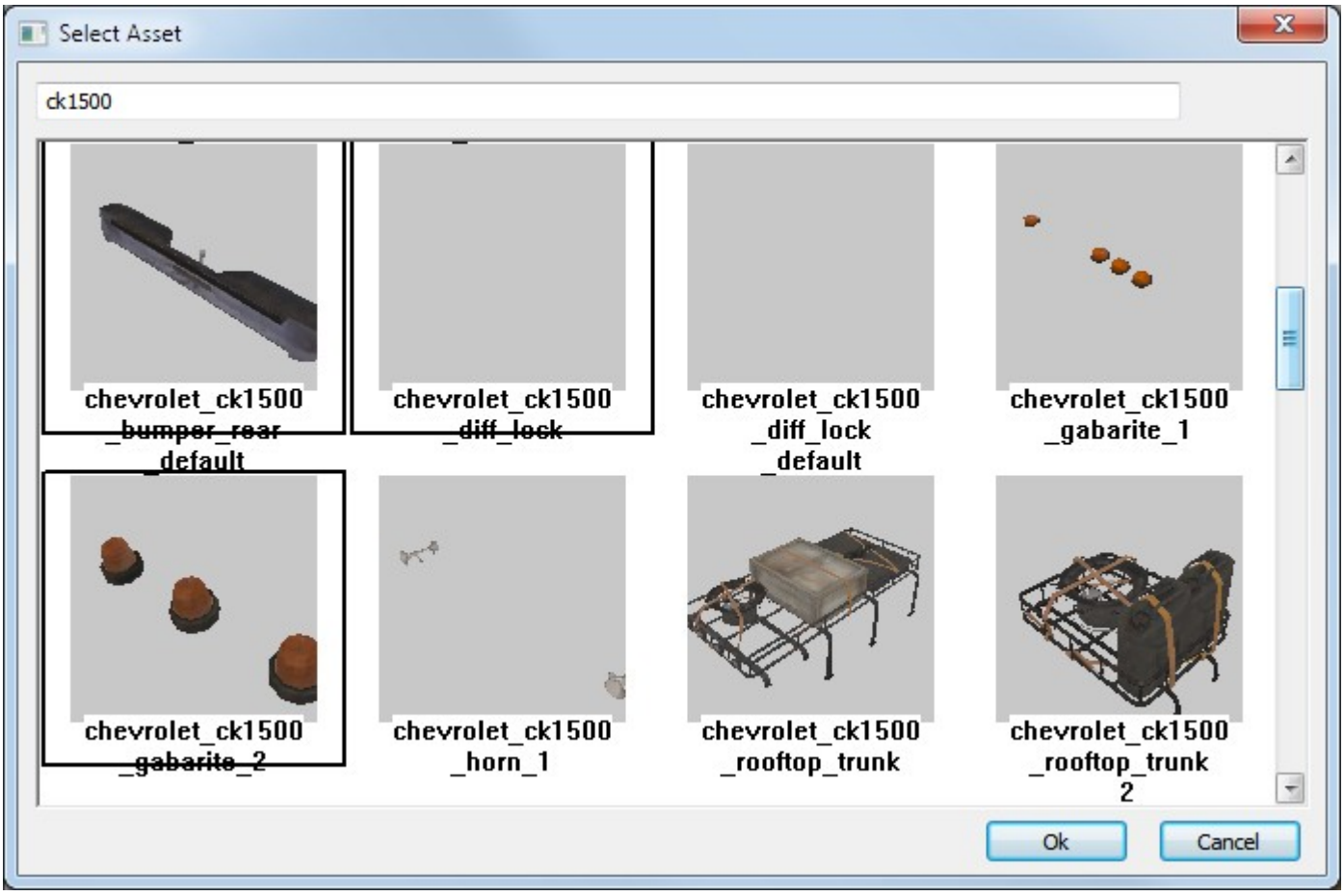
Choose Add-ons

> Scene.(Terrain).Trucks.*type*: Addons.Type: multiple asset selection
>
> Specifies the add-ons to attach to the vehicle.
>
> Default: blank; only default add-ons are installed.

Selecting add-ons is performed using the pseudo-properties in the Addons section. Unlike for trucks and trailers, however, this dialog box allows you to select multiple items before clicking OK. Clicking an add-on in the dialog once selects it. Clicking it again deselects it. Selected add-ons are indicated with a black border.

Bug: The first time you click here, the editor may lock up for 30 or more seconds as it creates icons for all of the many add-ons. It will usually be faster for subsequent uses. It may be occasionally slow when it checks for any changes to its cached assets, although never as slow as that first time.

You can filter the list of add-ons by typing in the blank panel near the top of the dialog box. Only add-ons that include your text somewhere in the name are shown in the dialog box.



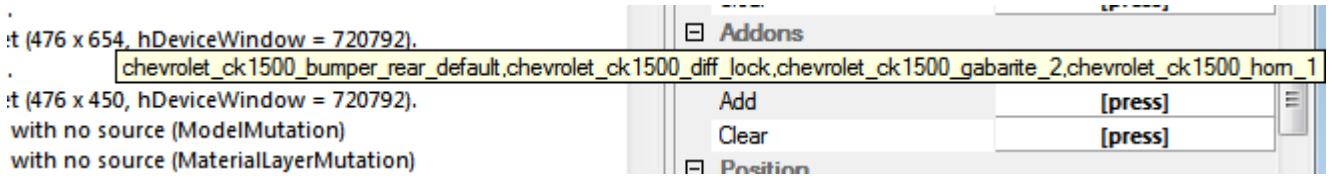
You may have noticed that the pseudo-property name for add-ons is Add instead of Edit. Once some add-ons are selected, click [press] next to Add again to add more items to the list.

Unfortunately, there is no way to selectively remove add-ons from the list. Instead, click [press] next to Clear, then re-add the desired add-ons.

The Editor does not check the compatibility of add-ons. The game quietly ignores any incompatible add-ons and does not attach them to the truck. Use the virtual garage to find the names of add-ons that are compatible with the truck and can be used together. The order of compatible add-ons doesn’t matter, as long as the necessary prerequisites are somewhere in the list.

Add-ons are not drawn on the truck in the main panel.

The list of addons may be longer than will fit in the properties panel. Hover your mouse over the list of addons to get the full list in hover text.



Default Add-ons

> Scene.(Terrain).Trucks.*type*: InstallDefaultAddons: True/False
>
> Specifies whether to install non-required default add-ons.
>
> Default: True; all default add-ons are installed where there is no conflicting part.

Most default add-ons are part of a required group. It is possible to uninstall required add-ons in the virtual garage (e.g. to leave a truck without any exhaust pipes), but the game always installs (or re-installs) the necessary defaults when the map is loaded. E.g. if no exhaust is installed on the truck, then the default is installed regardless of the InstallDefaultAddons property.

However, a few trucks have add-ons that are default but not required. If InstallDefaultAddons is False, then these non-required default add-ons are not installed. E.g. the default spare wheel is not installed on the KHAN 39 Marshall.

Cargo

Cargo can be loaded onto the truck as an "add-on". All cargo add-ons can be found in the addon dialog by filtering for the word “cargo”. Note that in a few cases (particularly for logs), the cargo is shaped differently for a particular truck and so has a unique name. Be sure to pick the right add-on for your truck. Unfortunately, the dev tools cannot help test the compatibility of cargo.

Warning: Unlike other add-ons where order doesn’t matter, the appropriate truck bed for the cargo must appear in the add-on list before the cargo. This means that typically you will add the truck bed first, confirm and close the dialog box, and then re-open the dialog box to add the cargo.

To load multiple copies of the same cargo on a truck, add one copy, exit the asset dialog, then add another copy. Each copy is added to the list of add-ons, and all the cargo that fits will be loaded onto the truck. In the case of cargo of different types, cargo earlier in the list is loaded onto the truck closer to the front.

Truck ID

> Scene.(Terrain).Trucks.*type*: Id: string
>
> Specifies a unique identifier for this vehicle for use by objectives.
>
> Default: blank; objectives cannot refer to this vehicle.

The Id property is optional. If filled, it should provide a unique identifier for this exact vehicle. I.e another vehicle should not have the same Id value. This ID allows the vehicle to be linked to an objective (page 180).

For safety, I recommend that the truck ID be limited to a combination of lowercase a – z, uppercase A – Z, 0 – 9, and _. It should not include other characters, including spaces.

Starting Damage and Fuel

> Scene.(Terrain).Trucks.*type*: Damage %: numeric slider from 0 – 100
>
> Specifies the initial damage to the truck's components as a percentage.
>
> Default: 0.

Each component is damaged by the same percentage, including wheels.

Any repair add-ons or repair trailer always start out full of parts and are unaffected by the Damage %. The Damage % property has no effect on a trailer since it has no components that can take damage.

> Scene.(Terrain).Trucks.*type*: Visual damage %: numeric slider from 0 – 100
>
> Specifies the initial damage to the vehicle's body panels as a percentage.
>
> Default: 0.

Visual damage is only visual; it has on effect on the operation of the truck. Visual damage is evenly distributed across the truck body. Even 5% visual damage can look rather significant (shown below), so don't overdo it. The Visual damage % property also applies to a trailer.

> Scene.(Terrain).Trucks.*type*: Fuel %: numeric slider from 0 – 100
>
> Specifies the initial fuel in the vehicle's internal fuel tanks as a percentage.
>
> Default: 100.

Any refueling add-ons always start out with full fuel and are unaffected by the Fuel %. If a trailer is placed on its own, the Fuel % property specifies the initial level of any refueling tank on the trailer.

Locked and Unlocked Vehicles

> Scene.(Terrain).Trucks.*type*: Locked: True/False
>
> Specifies whether a truck or trailer is locked at the start of the map.
>
> Default: True.

When the Locked property is True, the Editor adds a (locked) suffix to its name and draws a padlock icon on its silhouette in the main panel. A vehicle starts locked by default.

A locked vehicle has three attributes of interest:

- The player cannot drive a locked truck.
- When the player discovers a locked vehicle, she gains an experience bonus.
- When the player discovers a locked vehicle, it becomes available for sale in the garage or trailer store (if it wasn't already available).

See the Discovery of Vehicles section on page 222 for more information about the discovery process.

Reserved Vehicles

A truck or trailer can be reserved for use in a vehicle delivery goal (page 209), or a truck can be reserved for a truck repair goal (page 212). If so, its locked/unlocked status is ignored. An objective may reserve a vehicle permanently, or the vehicle may be awarded to the player upon completion of an objective.

A player cannot "discover" a reserved truck, nor can she enter or othewise drive it. A player can repair or refuel a reserved truck and can tow it with a winch from another truck. If the player attempts to change to a reserved truck via the functions menu, a helpful error message appears that depends on the objective that is reserving it.

A player cannot "discover" a reserved trailer, but she can otherwise attach it and use it a regular trailer (e.g. pack and unpack goods). The only thing a player can't do with a reserved trailer is sell it.

The Active Truck

> Scene.(Terrain).Trucks.*type*: Active: True/False
>
> Specifies whether the player starts the map in the truck.
>
> Default: True.

The Active property determines which truck the player starts in. If you set the Active property to True for a truck, the Editor automatically sets all other trucks on the map to False so that there is only one active truck. When the Active property is True, the Editor adds an (active) suffix to its name and draws a giant arrow pointing down at it. The Locked property is ignored when the Active property is True.

Bug: When an active truck is selected, the Editor draws a 32-meter gray disk around it in the main panel. This is a holdover from the MudRunner Editor, and it has no useful meaning in SnowRunner.

Although the Editor attempts to allow only one truck to be active, it is possible to end up with multiple active trucks, e.g. by duplicating an active truck. In this case, the game displays an error message, but allows the map to load; it uses only the last truck marked active in the list of trucks in the Scene View, and it treats the other trucks as inactive.

If no truck is marked as active, then the player will spawn in free space with no ability to move the camera or enter a truck. If the Tools menu is enabled in the game, you can use it to jump into an existing truck. See the Dev Tools section on page 222 for details.

If the active “truck” is actually a trailer, the player spawns in the trailer. From here, the player can move the camera or enter a different truck, but the trailer cannot be driven. (No engine or drivetrain.)

If the active truck is reserved by an objective, the player still starts in that truck. I’m not going to try to document the behavior in this case. Don’t do it.

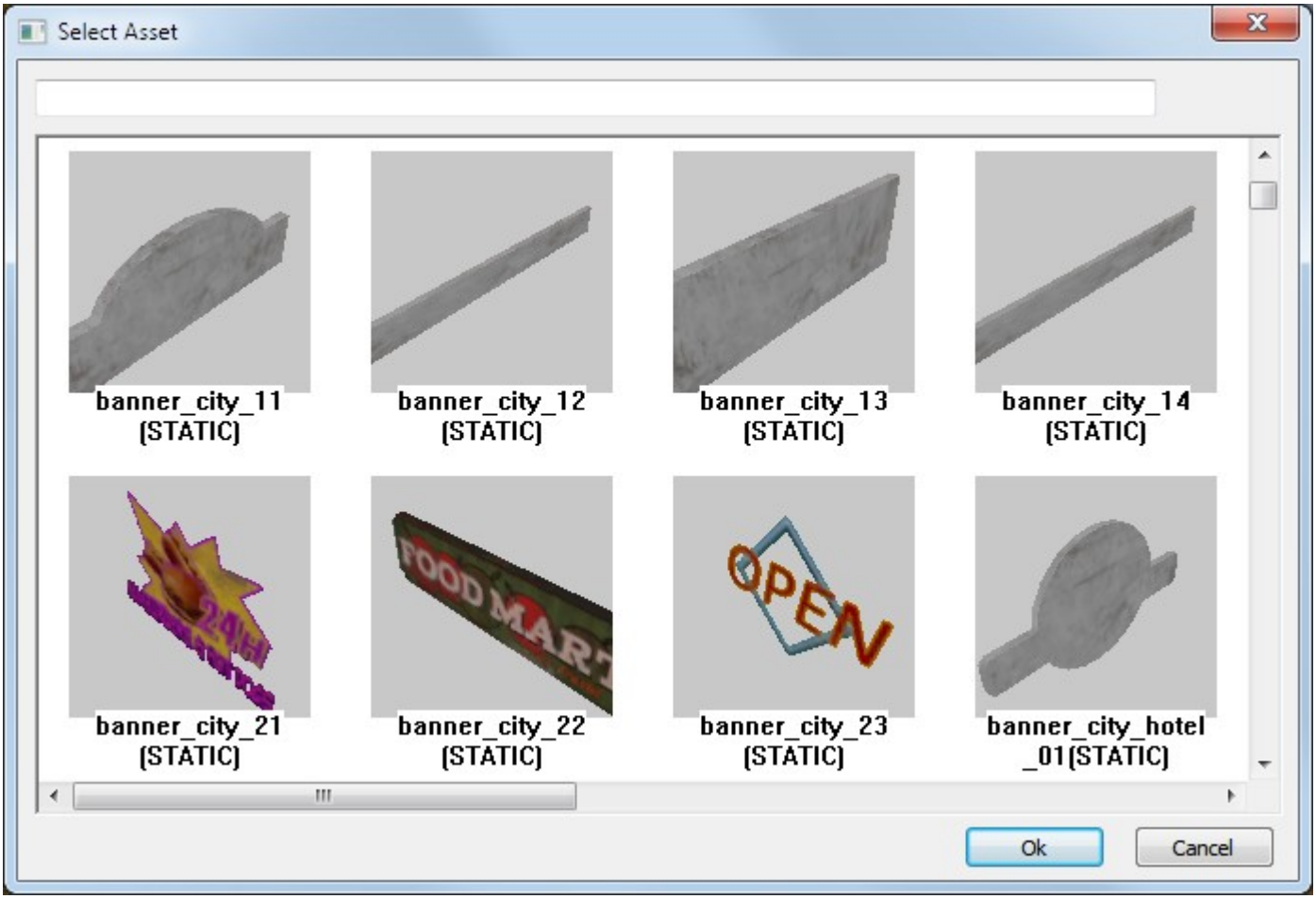
Hand Edit the XML

The properties for all of your trucks (and other data) is saved in your map’s XML file in the prebuild directory. Editing the XML by hand is useful for tasks that are difficult or impossible to do in the SnowRunner Editor. For example, you can rearrange the trucks so that the active truck is first. Or you can use a text editor’s find and replace function to change every Locked property to False for testing purposes.

More detail about hand-editing the XML file appears in Appendix B on page 261.

Models

To add a model, choose Add Model from the context menu. A window appears where you can select what type of model you wish to create.



Bug: The first time you click here, the editor may lock up for 30 or more seconds as it creates icons for all of the many add-ons. It will usually be faster for subsequent uses. It may be occasionally slow when it checks for any changes to its cached assets, although never as slow as that first time.

The model type is displayed in the Brand property, but it cannot be edited.

> Scene.(Terrain).Models.*type*: Brand: read-only string
>
> Indicates the model type.

Conveniently, you do not need to own any DLC in order to use models that were released as part of DLC. Nor does a player need to own any DLC as long as her game is up to date. This is unlike trucks, which do require you to own the necessary DLC (page 59).

Bug: DO NOT USE the Replace action in the context menu. It doesn't work right; it can't reliably be canceled; it will flicker between old and replacement models; it will silently corrupt your models until you quit the Editor and restart. Instead, create a new model of the desired type, then delete the old one.

If you accidentally select Replace, cancel it immediately. If cancel doesn't work, select a new model (not the original one), save your work if necessary, then entirely quit the editor. When you reload your map, the XML may have the wrong model, but at least it is a consistent, error-free model. Also verify that the model types are correct for any models you had touched recently and the most recent models that you added (listed last in the Scene View).

Unlike a truck, you can left click a model in the main panel to select it.

Broken Model

If the Editor does the wrong thing when packing a map, then the game may not be able to render a model correctly. When this happens, the game renders a default "broken model" instead, a white exclamation point on a red pedestal.

If you see this, entirely quit the **Editor**, reload, and repack your map. In my experience, this should correct the problem.

Move, Rotate, or Scale a Model

> Scene.(Terrain).Models.type: Position.X, Y, Z: numeric, in meters
>
> Specifies the initial location of the model.
>
> Default: ground level in the center of the main panel.

> Scene.(Terrain).Models.type: Rotation.X, Y, Z: numeric, in degrees clockwise rotation
>
> Specifies the initial orientation of the model.
>
> Default: 0, 0, 0; facing east (although what counts as the "front" of a model varies among models).

Move or rotate a model in much the same way as for trucks (page 54). Unlike trucks, however, models cannot be tied to the terrain by way of a Land property. Instead, select Do Land from the context menu to move the model up or down to an appropriate height.

Do Land works best when the terrain is completely flat under the model. Otherwise, the Editor moves the model so that all terrain points touch the bottom of the model's bounding box or are below it. This may allow the edges of the model to intersect terrain where the terrain is interpolated between its defined grid of points.

You're on your own to set the orientation of the model. Again, it works best if the ground is completely flat under the model so that the default model orientation looks natural.

If the terrain is not completely flat, be sure to careful check around the base of the model for gaps and either re-orient the model or adjust its height until it has a solid connection.

Scale

> Scene.(Terrain).Models.*type*: Scale: numeric
>
> Specifies the size of the model as a fraction/multiplier of the default size.
>
> Default: 1.0.

Each model has a default size, but you can adjust its size to make it larger or smaller. This adjustment factor is the Scale property.

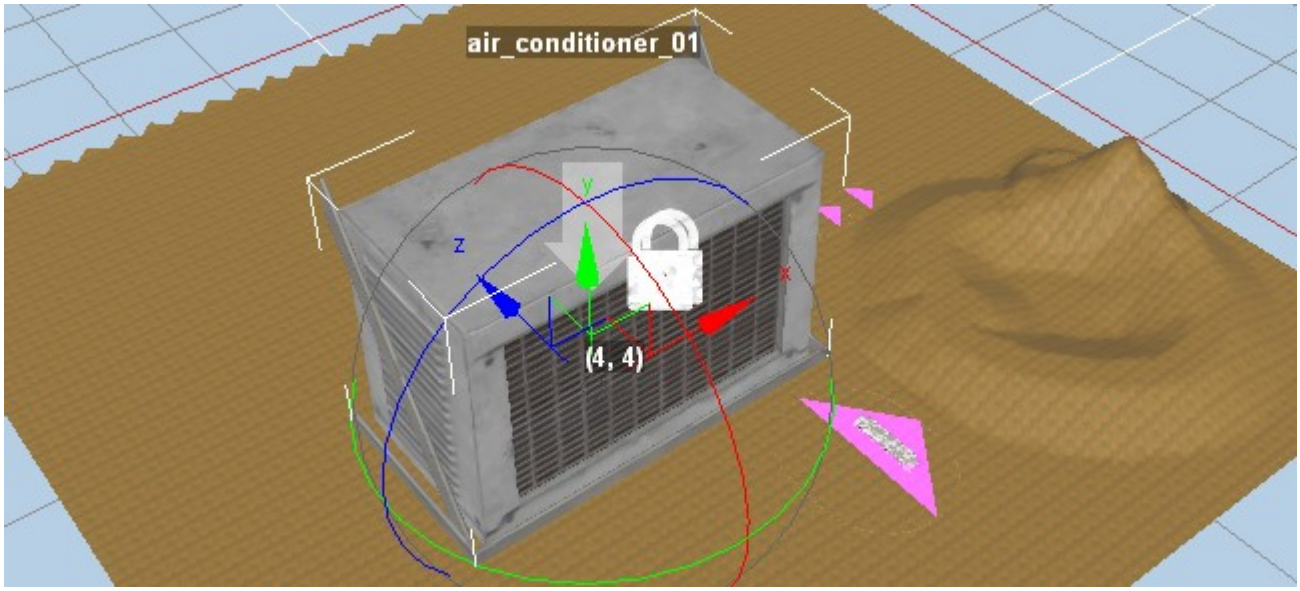
Tip: For many models, changing the size from the default can make it look unnatural. E.g. a larger building may look mostly fine, but its doors will look like they're made for giants. On the other hand, a bigger-than-usual windmill or a half-size bag of cement will look fine.

Tip: Many rock models are best used at larger scales. See the rock scale reference on page 251.

Tip: barrel_01 and barrel_02 should always use Scale = 2. The other barrels are fine with Scale = 1, and they should generally keep that scale in order have the industry standard barrel size. barrel_06 is the same as barrel_02 but with the correct default scale.

You can adjust the Scale value directly in the property panel. Alternatively, in the main panel you can left click in the diamond at the center of the model's axes and drag up or down to increase or decrease the scale, respectively.

The Editor limits the Scale to be no smaller than 0.01. There is no explicit upper limit, but if a model is much larger than anyone would consider reasonable, the Editor may crash. Be careful when typing values directly into the Scale field.



Model Physics

> Scene.(Terrain).Models.*type*: Collision Type: read-only string
>
> Indicates the default physics behavior of the model.

The Editor annotates each model with what type of physics it implements:

- [STATIC]: The model never moves when a truck hits it.
- [DYN]: The model can move and/or deform when a truck hits it.
- [DESTR}: The model can break into pieces when a truck hits it. This is a special case of a dynamic model.

A static model can be one that a truck runs over, runs into, or passes through (e.g. because it is only a visual effect). The physics of static models are very efficient for game performance.

A dynamic model is one that moves when a truck hits it (or comes very close to it). Since a dynamic model can move, it also has to model the effects of momentum and gravity and check for collisions with the terrain and other obstacles.

For efficiency, the game enables physics for a dynamic model only when a truck gets very close to it. After a short period, of activity, a dynamic model sinks into the ground and is removed from the game. It is reset to its initial position only when the player gets far enough away that she probably won't see it reset.

A destructable model is one that can break into multiple pieces when a truck hits it. Destruction may be trivial or may require considerable force. Once destroyed, each piece of the original model then acts like its own dynamic model.

If a static model is placed such that it intersects the ground or another static model, nothing special happens. Indeed, it is quite common to place a static model slightly depressed into the ground so that there is no visible gap below it.

On the other hand, if a dynamic model isn't placed well, it may react with the environment before the truck touches it. This can cause the model to fling itself into space, or even fling itself at the player's truck, causing unexpected damage.

It is generally best to place a dynamic model exactly on level terrain or just above bumpy terrain.

Mud and snow are a special case since a dynamic model will sink slowly into these surfaces when the player's truck gets near. You can consider slightly depressing the model's initial position into the mud or snow. E.g. for deep mud, barrel_06's natural buoyancy is about 40 cm deep, and for deep snow (without mud underneath), 40 cm is coincidentally the maximum depth that anything can sink. Be sure to test nudging the model with a truck in the game to see how it reacts.

Collision Notification

If Collision Notification is enabled in the Editor's toolbar, then the Editor highlights a dynamic model that is in a position to collide with another object or the terrain. The highlight changes the bounding box of a deselected model from its normal gray to violet.

Bug: Because the bounding box is only colored when a model is deselected, you can't see potential collisions while you are manipulating a model, only when you are done.

Bug: The bounding box is color is changed only for a model as it is deseleced. The color does not change for other models that it is now or was previously colliding with.

Bug: The Saber guide indicates that the bounding box should be red or violet depending on whether the dynamic model collides with the terrain or another model. However, in the current Editor, all collisions are highlighted with violet.

A collision notification doesn't necessraily mean that the model will react violently when approached. The forces may be small enough that it only moves gently away from the collision, or friction may even prevent it from moving at all.

Conversely, the lack of a collision notification doesn't mean that the model will stay put when approached. If it isn't supported properly, gravity could cause it to fall when its physics are enabled.

Freeze Physics

> Scene.(Terrain).Models.*type*: Freeze Physics: True/False
>
> Specifies whether to force the model to behave as a static model.
>
> Default: False.

You may prefer to use a dynamic model as if it is a static model, e.g. when it is clustered with other models in a static scene. This can improve performance (particularly if there are a lot of models) and also be less confusing than having some models move while other models stay put.

To treat a dynamic model as a static model, change its Freeze Physics property to True. A truck can still collide with the model, but the model won't move as a result. When a model has Freeze Physics enabled, its bounding box is black when deselected.

Freeze Physics has no effect on a static model other than to change its bounding box color. It does not stop animations.

Level of Detail

For performance reasons, most models have multiple levels of detail. When viewed up close, all details are visible. But from a medium or far distance, a simpler model is used. The models from Saber are generally done quite well with their levels of detail so that you don't see a difference as you approach or get further away. But you weill occasionally notice some small differences.

Procedural Snow Coverage

> Scene.(Terrain): Procedural Snow Coverage: numeric
>
> Specifies the opacity of snow texture on models and plants.
>
> Default: 1.0, but the value is ignored if the map isn't considered "snowy".

To add a snow-cover texture to models on your map, select (Terrain) in the Scene View, then edit the Procedural Snow Coverage property value. This snow coverage adds a white tint and snow-like sheen reflective glints to appropriate portions of a model, but it does not add any "thickness" to the snow.

The editor always applies procedural snow coverage to all unselected models. A selected model is rendered without snow coverage until deselected. The SnowRunner **game**, however, applies procedural snow coverage only for a "snowy" map (page TBD).

Bug: The default Procedural Snow Coverage is not appropriate for a non-snowy map. Since the Editor always applies procedural snow, even on a non-snowy map, all of your models will look white on their upper surfaces. You can change the Procedural Snow Coverage to 0.0 to make your models look more natural but even with a value of 0.0, the Editor adds a light dusting of snow. You can instead disable Settings → Show Snow By Up Vector, but that setting applies to all maps, so you have to remember to switch it when appropriate.

Snow is initially colored on the upper surfaces of each model. Increasing values for Procedural Snow Coverage increase the amount of snow and the angles at which it sticks:

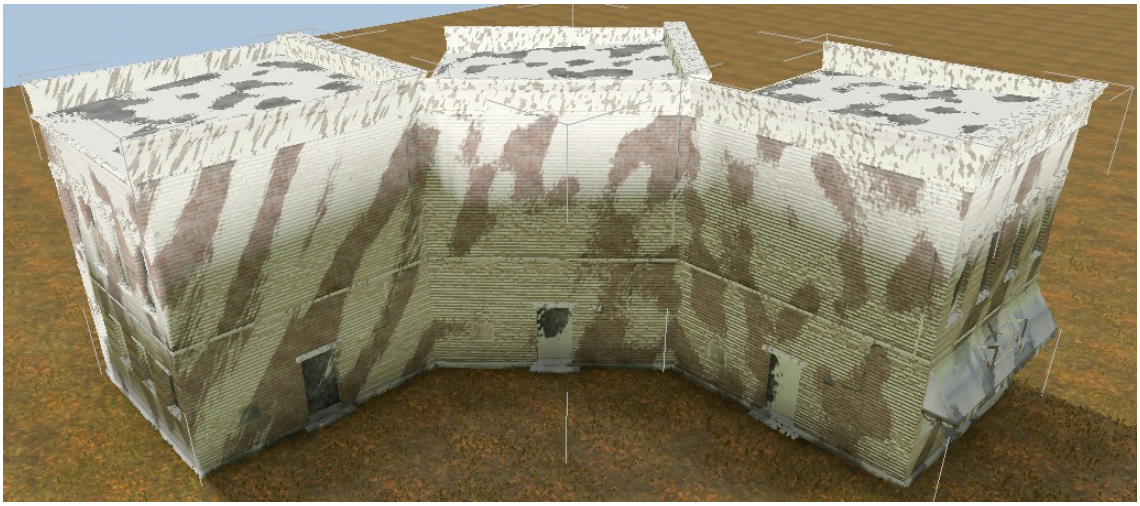
- For a surface that is perfectly horizontal and facing upward, values between 0.0 and 0.5 increase the opacity of the snow cover on that surface. At 0.5, the surface is entirely covered.
- Snow sticks to increasing angles from 0.0 until, at a value of about 1.95, it begins to stick to vertical surfaces.
- For a surface that is perfectly vertical (facing sideways), values between 1.95 and 2.5 increase the opacity of the snow cover on that surface.
- Snow sticks to increasing angles from 1.95 until, at a value of about 3.95, it begins to stick to horizontal surfaces facing **downward**.

- For a surface that is perfectly horizontal and facing downward, values between 3.95 and 4.5 increase the opacity of the snow cover on that surface. At 4.5, the surface is entirely covered, and greater values make no meaningful difference.

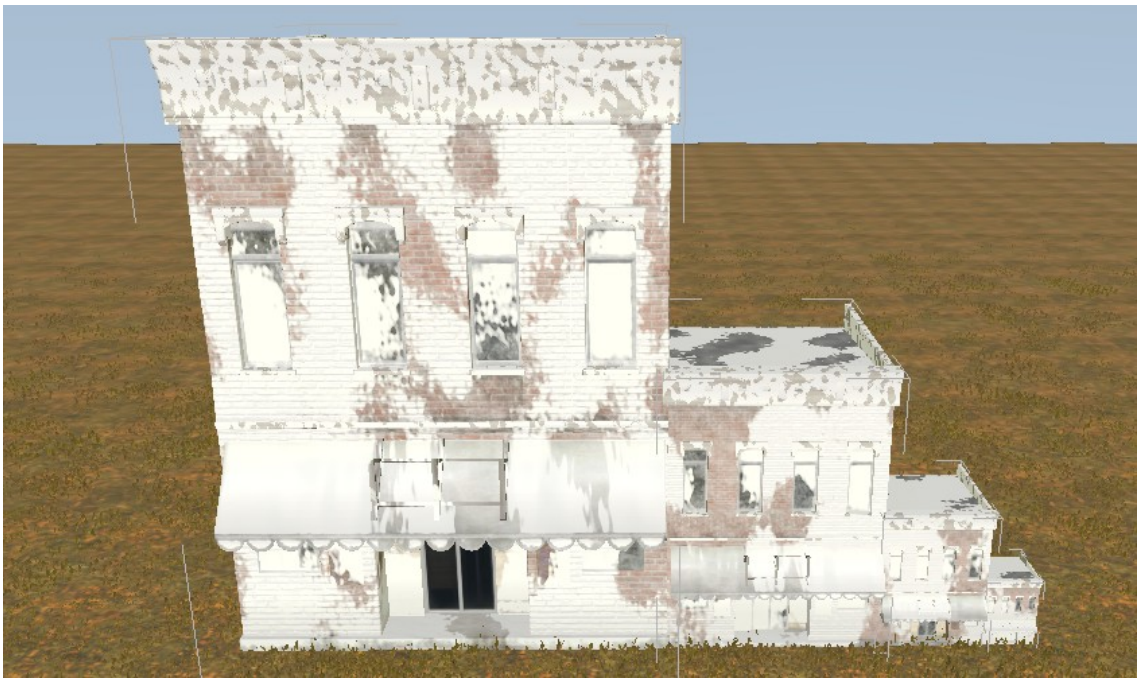
There is only a single Procedural Snow Coverage value for each map, so there is no way to specify more snow in parts of the map and less snow in other parts.

For dynamic models, snow coverage is applied based on the model's default orientation. Unlike static models, the snow on a dynamic model sticks to the same original surfaces even when it is tilted in the Editor. Likewise, snow remains on the same original surfaces regardless of how the dynamic model is punted around in the game.

Snow coverage is patchy on large models, even with a high Procedural Snow Coverage value. The snow patches are more "streaky" than "patchy" at some angles. But the patches are based on a volumetric heuristic, so the degree of snow coverage is consistent within a local region, regardless of how the horizontal rotation of the surface. (Vertical rotation changes the amount of snow cover as described above.)

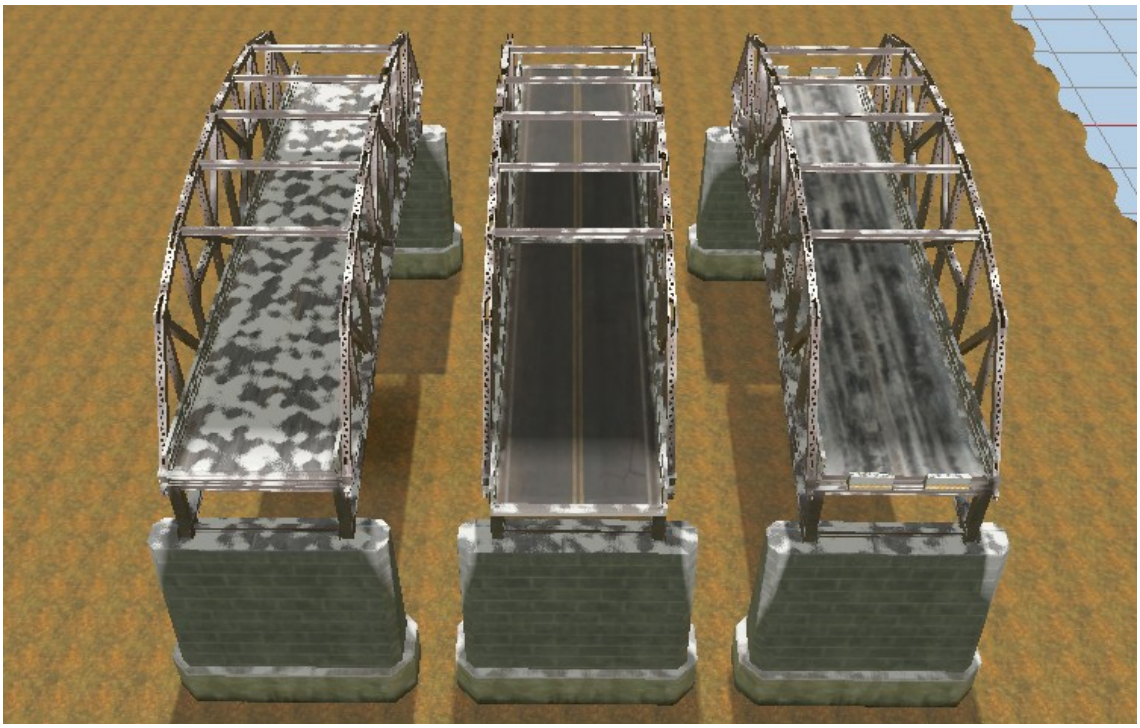


The size of snow patches depends on the properties of the various model elements. E.g. some parts of a model may have larger snow patches, while other parts have smaller patches. For a given model element, the patches are always the same size, regardless of how the model is scaled.



A model can also disable procedural snow coverage for some elements of the model. Depending on the model, this may apply only at high levels of detail. (The level of detail scales with the model, which explains why one building’s doors are clear of snow in the screenshot above, while the smaller buildings have procedural snow applied.)

The default procedural snow coverage works pretty well for most models even without any special model properties. However, some models need those properties to look good. For example, the screenshot below shows bridge_road_01 (which as the default patchy snow on the road), bridge_road_01_a (which fixes the road to be snow-free, the same as the default asphalt road overlay) and bridge_road_01_sn_objective (which also leaves off the procedural snow, but adds hand-painted snow to the road).



Shadows

Model shadows come in three flavors, described below.

Dynamic Shadows

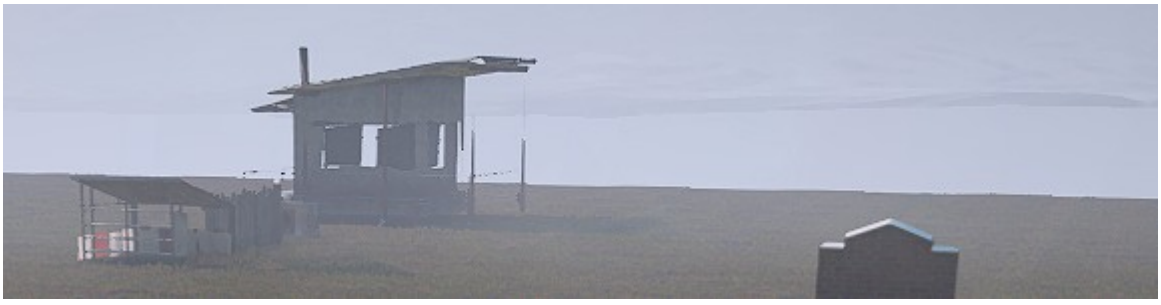
Dynamic shadows are the best looking shadows: they have high detail and are based on the sun's dynamic position. E.g. a low sun casts long shadows.

Dynamic shadows have a heavy performance cost, so the game only renders dynamic shadows within 50 meters of the player's truck. Dynamic shadows are not visible in the Editor.

There are no dynamic shadows at night. Although the back sides of model surfaces are less illuminated than the front sides (relative to the truck's headlights), the model does not actually block light from reaching other surfaces.

Static Shadows

Static shadows are similar to dynamic shadows, but are based on a fixed sun (or moon) position. They are computed for day and night when you rebuild terrain and are saved with the map. Static shadows can be seen in the Editor and are also used in the game for distances above 50 meters.



Note that the 50-meter cutoff is the distance to the shadow, regardless of the distance to the model. This occasionally results in visible artifacts such as a shadow that appears short in the distance resumes as a long shadow near the truck.

Static shadows are also visible in the game at night, but they're generally quite subtle and can only be noticed as a vaguely darker area behind a large object. Static shadows are not visible in Night mode in the Editor.

> Scene.(Terrain).Models.*type*: Disable Day Static Shadow: True/False
>
> Scene.(Terrain).Models.*type*: Disable Night Static Shadow: True/False
>
> Specifies whether to disable static shadows for the model during the day/night.
>
> Default: depends on the model.

You can manually disable shadows during the day and/or night using the Disable Day Static Shadow and Disable Night Static Shadow properties. For most models, the default is False, so shadows aren't disabled (and are enabled). Some models instead default one or both values to True. Models that can move (dynamic models and multi-stage models) sometimes disable their static shadows to avoid a ridiculous-looking shadow.

Bug: Many multi-stage models are missing this default disable of static shadows, so you have to do it manually for each one. Otherwise, for example, you could get a static shadow for a tower that hasn't been built yet. In general, a missing shadow looks better than an extra shadow.

Bug: Many dynamic models are also missing the default disable, although this tends to be less obvious simply because the model and its shadow are smaller. In the below screen shot, I've used a truck to push the sedan and barrel a little bit away. Note that their shadows are now on the wrong side compared to the static tower to the left.

Occlusion

An occlusion shadow results from blocking the omnidirectional glow from the sky rather than the directional glow from a light source.

Occlusion shadows are baked into models and cannot be modified. Models that can move (dynamic models and multi-stage models) typically disable their occlusion shadows to avoid looking ridiculous.

To view the occlusion shadow in the Editor after plaing a model, rebuild the terrain. The occlusion may be easier to see if Disable Day Static Shadow is True.

The screen shot below shows a shed which I have displaced from its occlusion shadow to make the occlusion shadow more obvious. (I also disabled the shed's static shadow to keep the occlusion distinct.)

Special Models

The following categories of models benefit from additional explanation.

Cargo

If you place a model on the map with one of the cargo_* model types, I call it a 'cargo model'. A cargo model is different than 'unpacked cargo' that has been unpacked after loading, was generated from a manual loading zone (page 177), or was spawned from an objective stage (page 202).

Bug: A cargo model isn't labeled on the navigation map, and it cannot be delivered. If it is lost, it cannot be respawned.

The player can convert a cargo model to regular cargo by lifting it onto a vehicle with a crane and packing it. It can then be delivered to a destination that doesn't require manual unloading, or it can be unpacked for manual

unloading. However, if the player attempts to transport it directly to the manual unloading zone without ever packing the cargo model, she won't be able to deliver it.

Cargo models don't disappear after a collision like other dynamic models, but the player can remove a cargo model in the same way as other packed or unpacked cargo. But like any other dynamic model, a cargo model stays in its original position on the navigation map regardless of how it is moved or transformed.

If you change the Scale of a cargo model, it reverts to the default scale when the player packs it on a truck or trailer. You probably shouldn't do that.

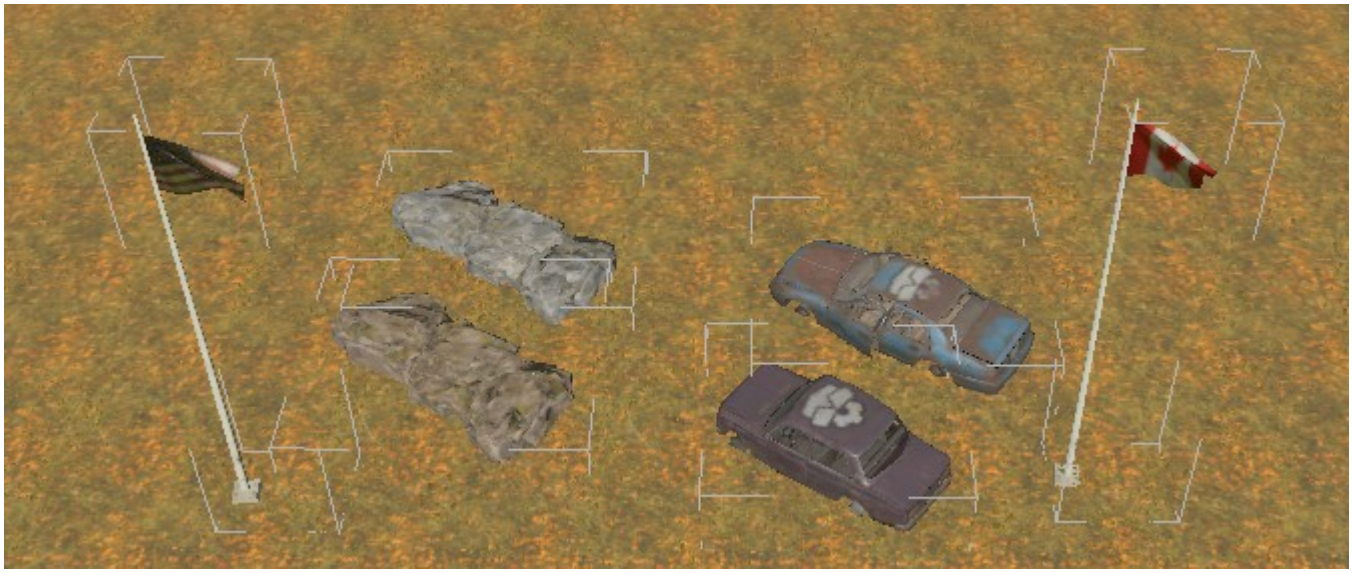
Country-Specific Models

Many models are labeled with us (or usa), ca or rus in their names, representing the US, Canada, and Russia.

In many cases, a model is only available for one country. Since Saber generally builds models for use in a specific map, many models associated with a specific type of location may only be available for one country (e.g. rocket factory buildings in Russia). Keep this in mind when planning a map.

Not all country-specific models are labeled as such. Check each model carefully for any writing on it that may tie it to a specific country.

If two models have the same name except for the country label, then the models often (but not always) have the same shape with different applied textures.



Nothing prevents you from using a mix of models from different countries on the same map if you feel that they look good together.

Winch Attachment Points

Some models include winch attachment points (similar to trees). Besides the familiar power poles and streetlights, many other pole-like structures have winch attachment points. (Other structure shapes can't support winch attachment points as well because nothing prevents the winch line from passing through the structure.)

Lying Trees

A couple of models appear to perfectly duplicate trees of the same name: pine_lying_a and spruce_lying_02. The major difference that I've found is that the trees are firmly rooted to the ground (despite being tipped over), while the models are easily pulled out of position. That makes the models much less useful as winch attachment points.

Lights

Some models include lights. SnowRunner has two types of lights, and a model often includes both types.

A "flare" is an ambient glow in the air. For an omnidirectional light, this is usually a ball of diffusely brighter color around the light source. For a directional light, this may be a cone of brightness that fades with distance from the light.

> **Bug:** The SnowRunner Editor does not display flares, even in Night mode.

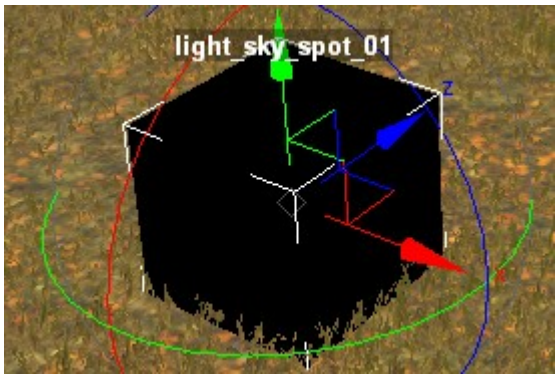
The screenshot below shows two red flares for the truck's marker lights, two orange flares for the running lights on the roof, and two streetlamps with slightly different colors of flares below and in front of them.

A “light” illuminates objects and terrain within range of the light. A light cannot be seen directly; it can only be detected by the change in brightness of other objects. A light can be omnidirectional or directional. The screenshot below shows streetlights with two different colors, plus some models of “broken” streetlights.

Bug: In the game, using the Change Time option in the Tools menu to switch to nighttime does not trigger lights to shine. Instead, use the Skip Time feature from the navigation map to change the time to Night.

To view the effect of lights in the Editor, you must explicitly rebuild the terrain.

light_sky_spot_01



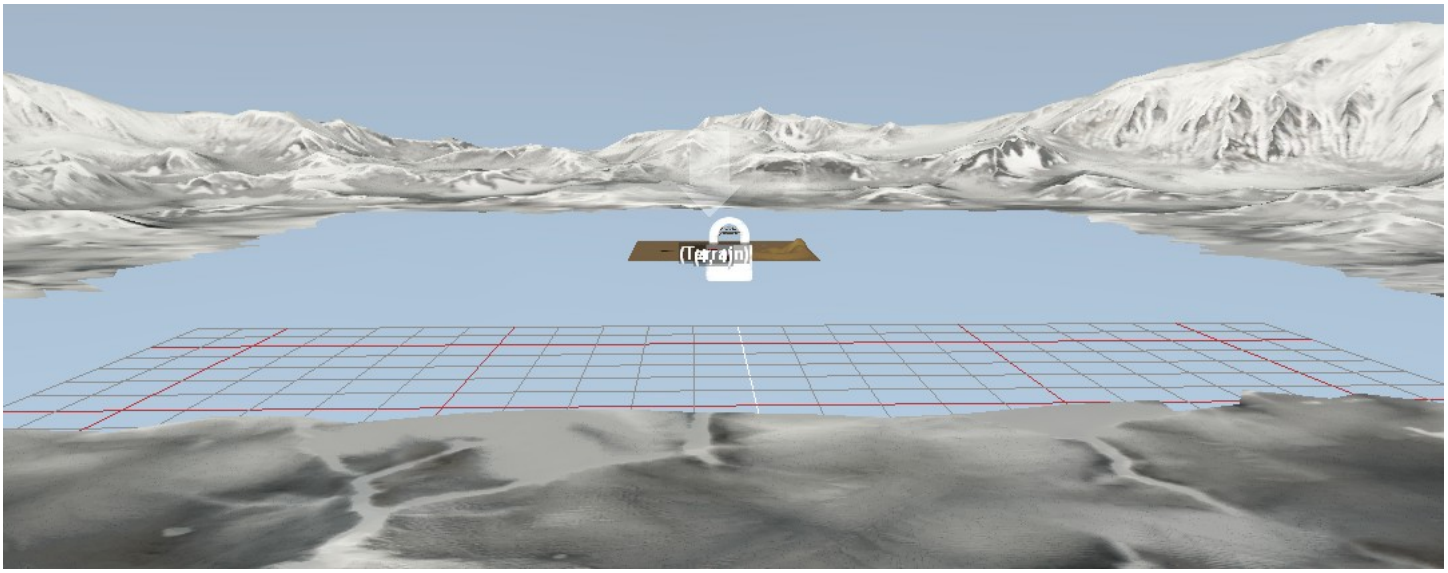
You might run across the light_sky_spot_01 model and wonder what it does. It includes a model of a black cube, but that shouldn't be visible in the game because you're expeced to bury the model underground.

The model comes with a **very** dim spotlight that hovers 25 meters above the cube and shines down on it. It's basically a nightlight for a small area of the map so that terrain and objects in it can be seen without other sources of illumination (e.g. when a truck's headlights aren't pointed at it).

In the screenshot below, I have buried the light_sky_spot_01 model about 20 meters deep, so its light source is about 5 meters above the ground. Since it's a directional light, it lights a circle below it. I placed a large, snow-covered flat surface below it to make the light effect more visible (including glints of reflection off the snow). It's very subtle in the Editor. If it's visible in the game, it requires a darker sky than I have yet tested with.

Farplane Models

A few models can be used as a backdrop beyond the borders of your map. These models have the names farplane_*.



You can scale or shift the position of a farplane model as you like, although it is not intended to match the borders of your terrain. Instead, the idea is that your terrain has high terrain near the edges so the the player is never looking more than slightly downwards when looking beyond the terrain borders, and the farplane model is then visible in the distance.

Some farplane models have a low side. These are appropriate when used with background water that extends to the horizon (page TBD).

Animated Models

Some models have animated motion, e.g. the country flags mentioned above. Animation is show in the Editor only when the model is deselected. Animation can be paused in the Editor with the Pause toolbar button.

Freeze Physics has no effect on animations.

Animals

I am aware of the following animal models:

- bear_01: a stationary white bear. It can only be seen at a distance. It disappears (fades away) if the player drives closer.

- wolf2: an unblinking pair of not-at-all wolf-like eyes. These appear at closer distances, but only at night. (TBD: unidirectional or bidirectional)

- birds_flying_01 and birds_sea_01: a flock of animated birds flying in circles and figure-8 patterns. Make sure to give them some height so that they aren't just skimming over the ground. Visible at any distance, day or night. The bird models are silent. To get associated sound effects, see page TBD.

- buterfly_flying_01: a cluster of butterflies flying in large circles and figure-8 patterns (similar to the birds above). Visible at any distance, day or night. The butterfly model isn't very good; unlike the birds, the butterfly wings are one sided, so the player can typically see only one at a time. The default scale is also way off. A Scale value of about 0.2 seems to strike a decent balance between being too big to be plausible and too small to be visible.

Bug: The wolf, hunter_01, and yeti_snow_man models appear to be broken. I can see the yeti_snow_man in a narrow range of distances in the Editor, but not in the game.

None of the animal models can be collided with.

Multi-Stage Models

Multi-stage models change state in response to completion of an objective stage (page 190) or delivery of cargo (page 201). These models have _objective in their name. The state change can be purely visual, or it can remove a barrier that was blocking trucks from passing. Some models have a fancy animation to show a change, while others simply swap to the new shape with a sound effect and a cloud of dust.

When viewed in the Editor, a multi-stage model shows all possible states simultaneously (but not the animations between the states).

Model ID

To associate a multi-stage model with an objective, you must give it a model ID in the Tag property.

> Scene.(Terrain).Models.*type*: Tag: string
>
> Specifies a unique identifier for this model for use by objectives.
>
> Default: blank; objectives cannot refer to this model.

The Tag property is optional. If filled, it should provide a unique identifier for this exact model. I.e another model should not have the same Tag value. This ID allows the model to be linked to the progress of an objective.

For safety, I recommend that the model ID be limited to a combination of lowercase a – z, uppercase A – Z, 0 – 9, and _. It should not include other characters, including spaces.

Use One Stage of a Multi-Stage Model

Some of the multi-stage models have an associated model without the _objective in its name. These single-stage models have only a single appearance, generally the “intact” stage of the associated multi-stage model. These are your best bet if you just want the model without its multi-stage progression.

Other multi-stage models don’t have an equivalent single-stage model, however. In that case, you can use a single stage of the multi-stage model with a bit more hassle. If a multi-stage model is not connected to an objective, then the game renders it in its default stage. A reference guide on page 252) describes the general appearance of each multi-stage model in each state. If you instead need the model in one of its other stages, then you’ll need to connect it to an objective.

Tip: To use a single-stage of a multi-stage model, associate that model stage with objective stage 0 of any objective (page 190).

Animation Camera

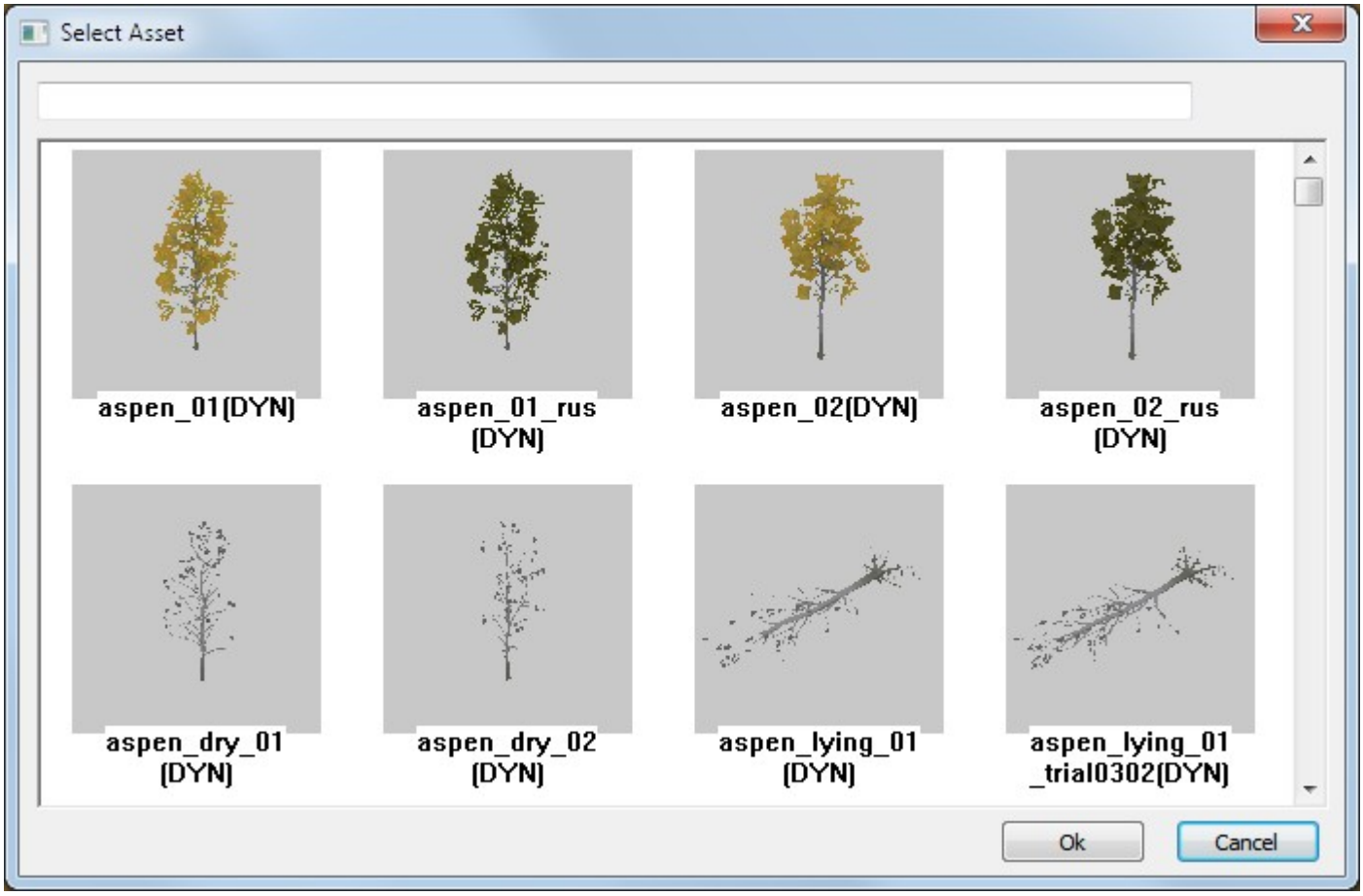
> Scene.(Terrain).Models.*type*: Animation Camera Frame Name: dropdown menu
>
> Specifies one of the model-specific camera angles to use for animation between model stages.
>
> Default: depends on the model type.

If a multi-stage model has an animation between stages, it can designate a camera position and movement from which to watch each animation. The Editor automatically fills the Animation Camera Frame Name property with a valid camera ID for the model, which is almost always objective_camera_0 for models with an associated camera. The property can be edited, but the only model with more than one available camera is bridge_road_01_a_objective. Its cameras all sweep along the same sides of the bridge at each stage, but each views a different amount of the surrounding terrain. They range from a low position that gives a long field of view in objective_camera_0 to a high position with a short field of view in objective_camera_5.

Individual Plants

You will add most plants to your map using distributions, described on page TBD. Distributions are great for randomly placing many plants, but you don't have exact control of them. Alternatively, you can also add an individual plants in much the same was as a model. This lets you place a plant exactly where and how you want it.

To add a plant, choose Add Plant from the context menu. A window appears where you can select what type of plant you wish to create.



Bug: The first time you click here, the editor may lock up for 5 or more seconds as it creates icons for all of the many add-ons. It will usually be faster for subsequent uses. It may be occasionally slow when it checks for any changes to its cached assets, although never as slow as that first time.

There are actually a few non-plants listed in the asset window, particularly rocks and ice chunks. These are grouped with the plants because they can be distributed in large numbers like plants. There are also some plants (e.g. grasses) that cannot be individually placed. They can only be created by distributions or materials.

Once the type of plant has been selected, you cannot change it. You can only delete the plant and add another.

> Scene.(Terrain).Plants.*type*: Brand: read-only string
>
> Indicates the plant type.

As with a model, you can left click a plant in the main panel to select it.

Bug: Something is very wrong with the Editor's rendering of individual plants. Many trees and bushes disappear at lower levels of detail, but only when selected. Some rocks radically change height if they are selected vs. deselected. I hope these bugs get fixed soon.

Bug: cane_chunk_a displays numerous warning dialogs about missing billboards and then displays as red rectangles at lower levels of detail. This plant isn't used in any distributions, and with this bug you shouldn't place it as an individual plant, either.

Move, Rotate, or Scale a Plant

> Scene.(Terrain).Plants.*type*: Position.X, Y, Z: numeric, in meters
>
> Specifies the initial location of the plant.
>
> Default: ground level in the center of the main panel.

> Scene.(Terrain).Plants.*type*: Rotation.X, Y, Z: numeric, in degrees clockwise rotation
>
> Specifies the initial orientation of the plant.
>
> Default: 0, 0, 0; facing east (although what counts as the "front" of a plant is rather arbitrary).

> Scene.(Terrain).Plants.*type*: Scale: numeric
>
> Specifies the size of the plant as a fraction/multiplier of the default size.
>
> Default: 1.0.

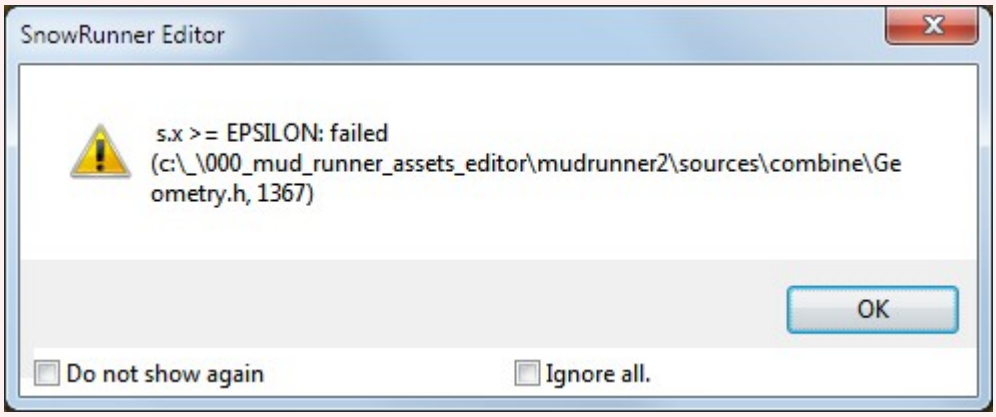
Move or rotate a plant in the same way as for trucks (page 54). Scale a plant in the same way as for models (page 73). Unlike models, plants all look fine with Scale = 1. As with models, you are free to adjust the scale to get a pleasing size.

Do Land

> Scene.(Terrain).Plants.*type*: Do Land: True/False
>
> Specifies whether the plant's position is tied to the ground.
>
> Default: True.

The Do Land property for plants is similar to the Land property for trucks (page 56).

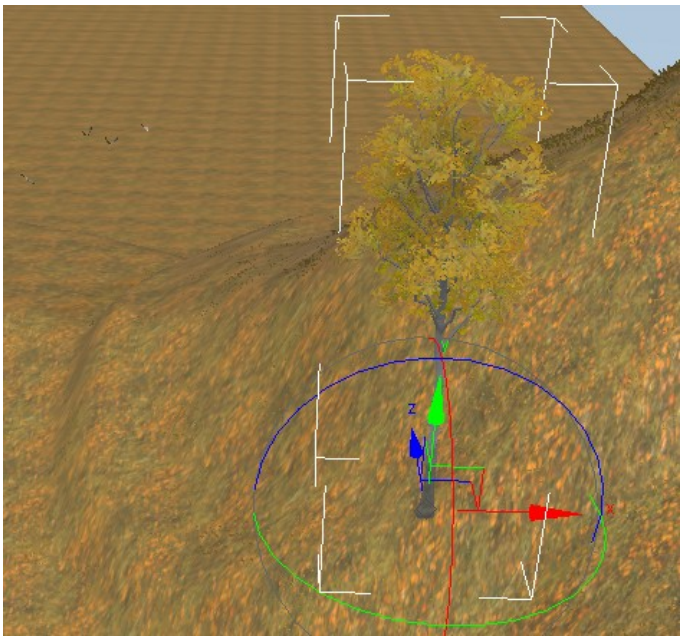
Bug: The Editor will sometimes change a plant's Scale to 0 while you're editing it. I haven't figured out the pattern, but it seems to occur most often (or only?) when I set Do Land to False. The 0 Scale causes a few identical warning dialogs to pop up whenever you do anything else. The Editor continues to change the Scale to 0 if you try to fix it, so the only workaround is to delete the plant.

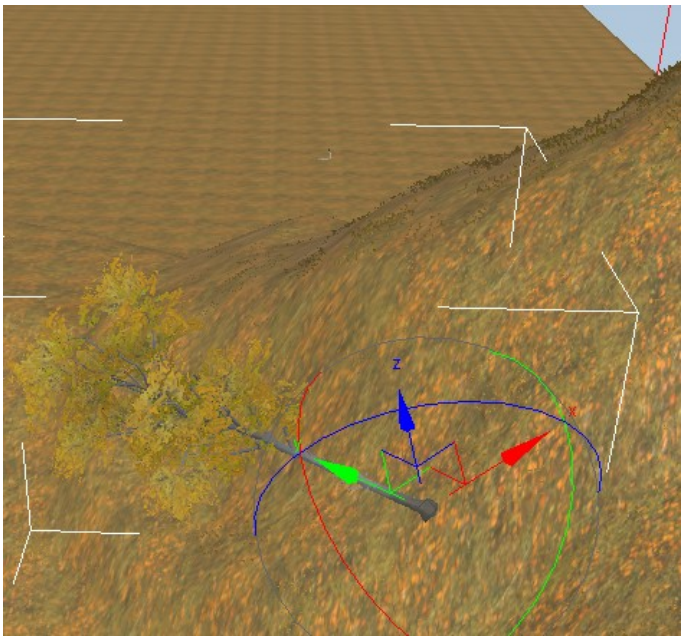


Perpendicularity

> Scene.(Terrain).Plants.*type*: Perpendicularity: numeric slider from 0.0 – 1.0
>
> Specifies the fraction of the ground's orientation to transfer to the plant's orientation when Do Land is True.
>
> Default: 0.0.

By default, plants are oriented vertically, even when Do Land ties them to angled terrain. Especially for bushy type plants, it may make sense to have them grow perpendicular to a hillside rather that vertical. You can rotate the plant by hand, but for this task it may be easier to increase a plant's Perpendicularity property.





A perpendicularity of 1.0 (100%) doesn't make sense for large trees, but it may make sense for pumpkins. And slight perpendicularity may also improve the appearance of larger plants.

Bug: Decreasing the Perpendicularity property doesn't stand the tree back up again. You have to set the Perpendicularity value to 0, then set the Rotation.Y and Rotation.Z values to 0 by hand, then increase Perpendicularity to the desired value.

Bug: The rotation widgets don't work when the Perpendicularity is greater than 0. If you try to use any rotation (even a local rotation around the plant's axis), the plant will rapidly tilt toward 100% perpendicularity. The rotation widgets only work when the Perpendicularity is exactly 0.

Tip: If the bugs in Perpendicularity give you a headache, just set Perpendicularity to 0 and rotate the plant by hand. With Local Transform enabled, you can rotate the plant to face the slope, and then it's easy to tilt it downslope.

Plant Physics

The Editor annotates each model with what type of physics it implements, the same as for models (page 75). Most plants are [DYN] or [DESTR], but there a couple of [STATIC] rocks.

Unlike most dynamic models, most plants (except chunks) are physically attached to their origin, even if that origin is floating in space. The main consequence is that you don't have to worry about plants sinking into mud or snow. They also generally bend in place rather than being pushed around, although that changes if the plant breaks into chunks.

Plants also don't physically interact with models or other plants, so plants can be placed close to obstacles without worrying that they may have adverse collision events. Although Collision Notification highlights "collisions" by plants, these serve at most as a warning that the visual appearance may look odd.

Freeze Physics

> Scene.(Terrain).Models.*type*: Freeze Physics: True/False
>
> Specifies whether to force the plant to behave as a static object.
>
> Default: False.

Like models, you can change a plant into a static object by setting its Freeze Physics property to True. It still blows in the wind, but won't budge after a truck impact or bend in response to winching. Since plants are quite performance efficient even in great numbers, there is little reason to use Freeze Physics with them, and it has the potential to greatly confuse the player.

Procedural Snow Coverage

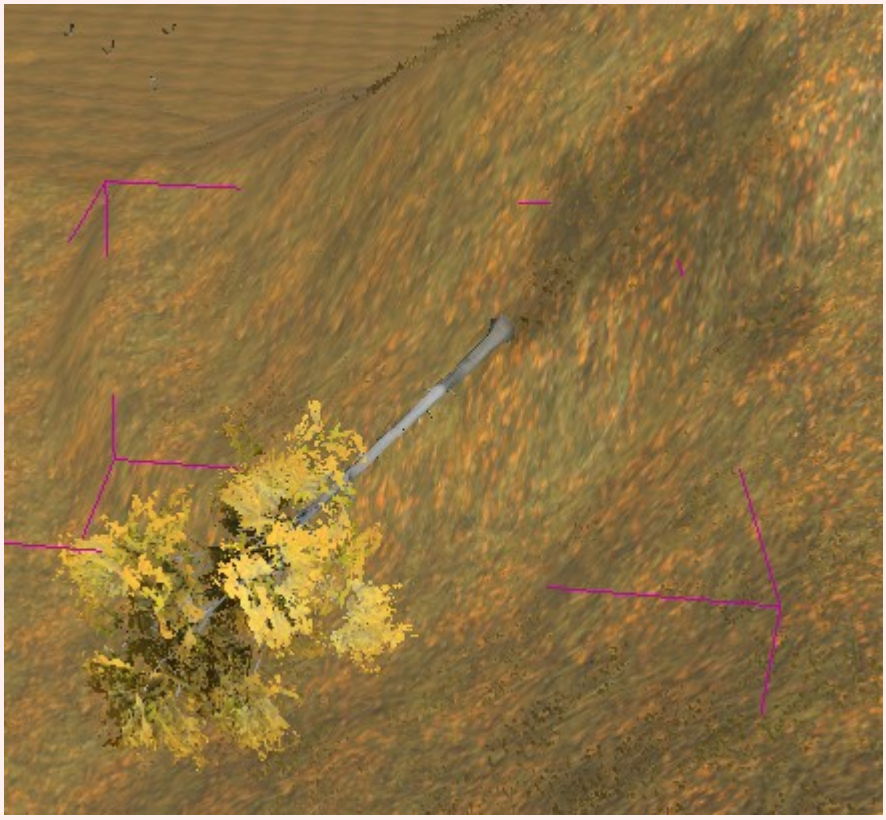
Procedural snow cover can be added to plants in the same as for models (page 77), and solid plant surfaces are covered similarly.

For plant foliage, snow cover on the leaves tends to increase until they are fully covered at a Procedural Snow Coverage value of about 1.9. The pattern of increasing coverage varies for each plant. E.g. for bush_b_rus with Procedural Snow Coverage = 1.0, roughly every other leaf is fully snow covered, while the remainder are mostly snow free.

Shadows

Plant shadows are implemented similar to model shadows (page 81). However, whether or not static shadows are displayed for plants is strictly a property of the plant type and cannot be changed in the Editor.

Bug: Static plant shadows are calculated as if the plant is standing in its default orientation.



Country/Season-Specific Plants

Because the initial US maps were set in autumn, many of the ‘default’ plant names use autumn colors, and those with rus in their names have summer colors. Feel free to use whatever mix of plants looks good to you.

Winch Attachment Points

Trees have winch attachment points. Mature trees (regardless of scale) tend to be unbreakable and therefore solid winch points. Younger trees tend to bend and may break when the winch stress becomes large.

Point Sounds

The game can periodically play a random sound when the player drives close to a particular location. To create such a sound, choose Add Sound from the context menu.

Name

> Scene.(Terrain).Sounds.*name*: Name: string
>
> Specifies the name of the sound to display in the Scene View at all times and in the main panel when the sound is selected.
>
> Default: blank; displays as (Terrain Sound).

To help keep your sounds straight, you can assign a name to each sound in its Name property. The name has no game purpose, however.

Sound Location

> Scene.(Terrain).Sounds.*name*: Position.X, Y, Z: numeric, in meters
>
> Specifies the position of the source of the sound.
>
> Default: ground level in the center of the main panel.

The sound doesn't have a scale or orientation, only a position. You can move it similar to a truck (page 54). Although the sound is initially created at ground level, there is no advantage to attaching it to the ground, so there is no equivalent to Land or Do Land for sounds.

Keep in mind that the "microphone" for the sound is in the game's camera, not in the truck. In 3rd-person view (outside the truck), you may need to move drive slightly away from the center of the sound in order to center the camera in its vicinity. Also note that in 1st-person view (inside the cockpit), all sounds are attenuated.

Sound Volume and Radius

> Scene.(Terrain).Sounds.*name*: Volume: numeric in the range 0.0 to 1.0
>
> Specifies the maximum volume of the sound as a fraction of its volume in the sound file.
>
> Default: 1.0.

> Scene.(Terrain).Sounds.*name*: Distance.Min: numeric, in meters
>
> Specifies the distance from the 3-D sound position at which the sound volume begins to fade to zero.
>
> Default: 5.

> Scene.(Terrain).Sounds.*name*: Distance.Max: numeric, in meters
>
> Specifies the distance from the 3-D sound position at which the sound volume becomes inaudible.
>
> Default: 35.

The sound plays at the specified volume anywhere within the spherical radius given by Distance.Min. Outside this radius, the sound attenuates to zero at the radius given by Distance.Max. There is no visual indication of these distances in the main panel. Be sure to include the height of the sound in your calculations.

Bug: The Editor will crash when you pack your map if Distance.Min ≥ Distance.Max.

Sound Files

> Scene.(Terrain).Sounds.*name*: Sound file: string
>
> Specifies the relative path and base filename of the sound file(s) to play.
>
> Default: blank; triggers a File not found error when the map is packed.

You can use either the game's built-in sounds or your own custom sounds, and you can select a single file or a set of files to be played in a random order.

Built-in Sound Files

SnowRunner keeps its sound files in its shared_sounds.pak archive, described on page 28. If you've extracted these files somewhere, you can browse them to find a sound that you like. You'll need a sound player that can handle PCM format, such as VLC or Audacity.

Saber recommends that point sounds be in mono format (not stereo). If you are using a built-in sound file, make sure it's in the right format. In VLC, ctrl-J shows the media format. (Although my non-attuned ears think that stereo sounds fine.)

The best built-in point sounds are probably in the [sound]/actors directory. All of these are mono except in the actor_construction and actor_rocket subdirectories, which have sounds intended for multi-stage model animation (page 90).

Single File

Set the Sound file value to the sound file's path within the shared_sounds.pak archive, excluding the initial [sound] directory name and the trailing file extension. For example, if the sound is in [sound]/actors/actor_rocket/actor_rocket_launch.pcm, then set the Sound file property to actors/actor_rocket/actor_rocket_launch. Either / or \ can be used as directory separators.

File Set

The shared_sounds.pak archive includes many subdirectories with sets of files ending in two underscores and one or more digits, e.g. __1 through __7. If you specify one of these files without that __*n* ending, then the game will play all of the files in a random order. E.g. actors/actor_bird_woodpecker_rnd_set/actor_bird_woodpecker_rnd.

The game never repeats a sound when playing a set in a random order. That means that a set with only two sound files will always alternate between them.

Custom Sound Files

If you create your own sound files, they **must** be in 16-bit WAV format and have a *.wav extension. The Editor will convert it to PCM format when you pack your map.

(Technically, WAV files also internally use a PCM encoding, but these are not the same encoding as the built-in sounds. In fact, if you try to copy one of the built-in files to your own directory, the Editor will try to convert it and fail.)

Saber recommends an encoding frequency of 44.1 kHz, but any frequency that sounds good is fine. Saber also recommends the mono format for point sounds.

Put the *.wav file(s) in %USERPROFILE%\Documents\My Games\SnowRunner\Media\sounds or one of its subdirectories. You may need to create the sounds directory. The *.wav files are shared by all of your maps, so they don't go in Media/prebuild.

> **Tip:** If your sound files will be used by only one or a few maps, give them their own subdirectory to keep them separate from the sound files of other maps.

Some mistakes are better than others:

- If your Sound file refers to a file that the Editor can't find in your sounds directory (or in shared_sounds.pak), it will pop up a warning dialog when you pack your map.
- If your Sound file refers to a *.wav file in your sounds directory that is not in the proper format, the Editor will pop up some warning dialogs when you pack your map.

- If your Sound file refers to a file in your sounds directory with a different extension, the Editor doesn't pop up any warning dialogs, and the sound won't play in the game. Always use the correct file extension to avoid unnecessary frustration.

Single File

Set the Sound file value to the sound file's path within the sounds directory, excluding its extension. E.g. if your sound file is in Media/sounds/test/evil.wav, set Sound file to test/evil. Either / or \ can be used as directory separators.

File Set

To play a number of sound files in a random order, end each file name with two underscores and one or more digits (__*n*). E.g. if your sound files are in Media/sounds/test/good__1.wav through Media/sounds/test/good__7.wav, set Sound file to test/good. The numbers at the end of the filename don't have to be consecutive.

Delay Between Sounds

> Scene.(Terrain).Sounds.*name*: Loop delay.Min: numeric, in seconds
> Scene.(Terrain).Sounds.*name*: Loop delay.Max: numeric, in seconds
>
> Specifies the range of values for the random pause after each sound before the next sound is played.
>
> Default: 0 – 0; results in continuous play of back-to-back sounds.

After playing a sound, the game inserts a pause of random length before playing the next sound. If the Min value is 0, then sounds can be played back to back. If the Max value is also 0, then sounds are always played back to back continuously.

Interestingly, the game appears to pick a random value between Min and Max even if Min > Max. You should stick to Min ≤ Max, however.

Sound Conditions

> Scene.(Terrain).Sounds.*name*: Conditions: string
>
> Specifies the conditions required to play the sound.
>
> Default: blank; the sound is played unconditionally.

You can set a point sound to play only during the day or only at night by setting the Conditions value:

- DAY_FOREST, DAY_WIND: play the sound only during the day.
- NIGHT_FOREST, NIGHT_WIND: play the sound only at night.

Bug: In the game, using the Change Time option in the Tools menu to switch to day or night does not trigger sound conditions. Instead, use the Skip Time feature from the navigation map to change the time.

Note: A point sound works reliably only when the DAY_FOREST condition is used together wit DAY_WIND or when NIGHT_FOREST is used with NIGHT_WIND. If only a *_FOREST condition or only a *_WIND condition is used, the sound might not work. When listing both the *_FOREST and *_WIND conditions, the comma is important, but the space is not.

Ambient Sound Domains

An ambient sound domain is similar to a point sound, but it plays in a certain region without a specific point source. Ambient sounds always play continuously back to back. To create an ambient sound domain, choose Add Ambient Sound Domain from the context menu (or SoundDomains – Add SoundDomain).

The Editor lumps the new ambient sound domain in the SoundDomains feature category along with other types of sound domains. An ambient sound domain uses a music notes icon next to the name:  .

Name

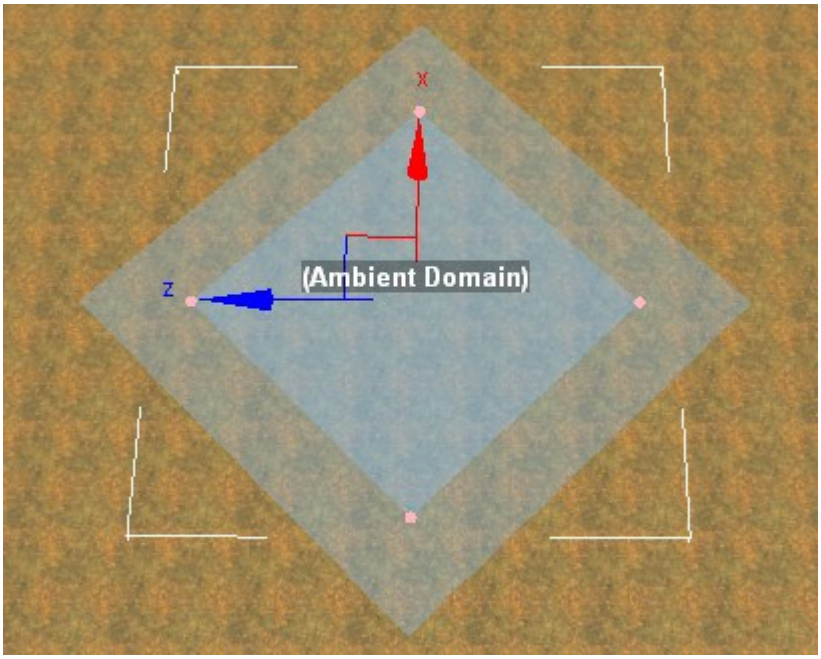
> Scene.(Terrain).SoundDomains.  *name*: Name: string
>
> Specifies the name of the sound domain to display in the Scene View at all times and in the main panel when the sound domain is selected.
>
> Default: blank; displays as (Ambient Domain).

To help keep your sounds straight, you can assign a name to each sound domain in its Name property. The name has no game purpose, however.

Tip: If you have trouble remembering the meaning of the icons for each type of sound domain, consider including the sound domain type in its name.

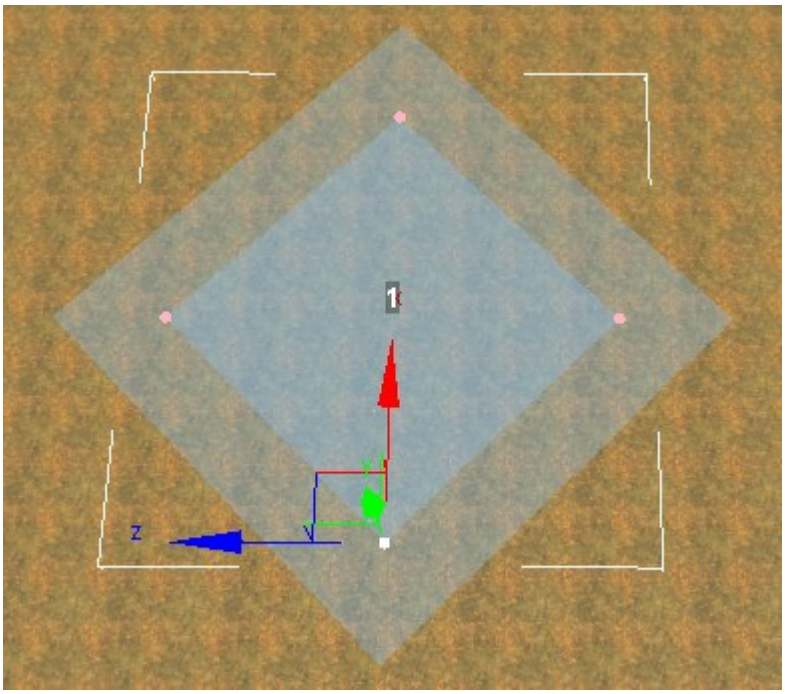
Sound Domain Location and Shape

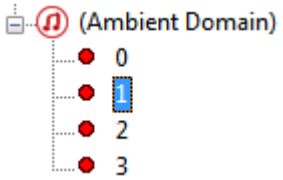
The sound domain doesn’t have a scale or orientation, only a position and a shape.



With the sound domain selected, you can move it in the ground plane. But because the shape encloses a 2-D area, you cannot change its height.

To change the shape of the sound domain, you can select a vertex from the main panel, or you can expand the sound domain in the Scene View and select a numbered vertex there. I find that it is sometimes difficult or impossible to click a vertex in the main panel, so I have to select it in the Scene View instead.





> Scene.(Terrain).SoundDomains.*name*.*n*: Position.X, Y, Z: numeric, in meters
>
> Specifies the position of the selected vertex.

Once the vertex is selected, you can move it in the usual ways, either by dragging in the main panel or by typing a new value into its Position X or Position Z property.

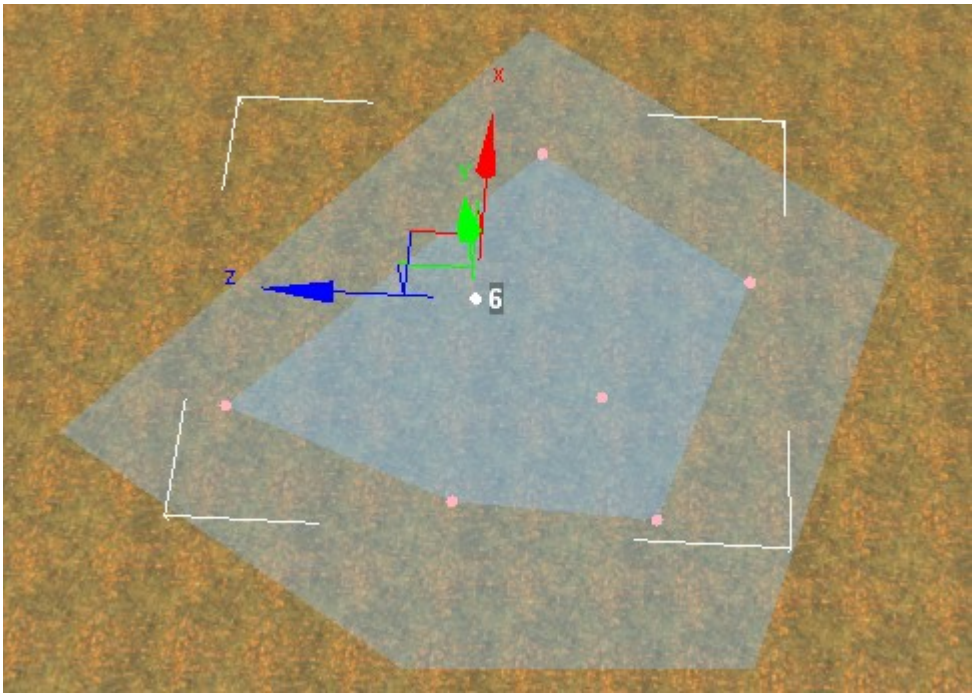
Bug: Despite having height control widgets in the main panel and a Position Y property, a sound domain vertex is always attached to ground.

Note that the sound domain as a whole does not have position coordinates in the property panel. When you move the whole sound domain in the main panel, you are actually moving all of its vertexes together.

The more transparent polygon outside of the vertices represents the fading distance, described on page 107.

Update Vertices

No matter how you arrange your vertices, the sound domain is always the convex shape formed by the outermost vertices, igoring the order of vertices, and ignoring inner vertices.



With the whole sound domain or any vertex selected, you can choose Update from the context menu to rationalize the vertices. The Editor removes unnecessary inner vertices and renumbers the outter vertices to follow a clockwise path around the perimeter. The vertex with the smallest X coordinate becomes vertex 0.

(Exception: if the domain has only three vertices, Update does nothing, even if the vertices are currently laid out counterclockwise.)

Bug: If the domain has a triangular counterclockwise perimeter formed by the first three vertices and has additional higher-numbered vertices inside the perimeter, Update causes the Editor to crash. Similarly, if the domain has four or more vertices all exactly in a line, Update causes the Editor to crash.

Add a Vertex

If an outer vertex is selected, you can choose Add Vertex from the context menu to add a new vertex to the middle of the segment connected to the next vertex in clockwise order (regardless of the current vertex numbering). If an inner vertex is selected, or the whole sound domain is selected, Add Vertex behaves as if the vertex with the smallest X coordinate is selected.

(Exception: if the domain has only three vertices, the new vertex is added between the selected vertex and the next higher numbered vertex. If the whole sound domain is selected, the new vertex is added between vertices 0 and 1.)

Delete a Vertex

With a vertex selected, choose Delete from the context menu or press the Delete key to delete the vertex. Delete is not an option when the domain only has three vertices.

Sound Files

> Scene.(Terrain).SoundDomains.♫ *name*: Sound file: string
>
> Specifies the relative path and base filename of the sound file(s) to play.
>
> Default: blank; triggers a File not found error when the map is packed.

Choose a sound file or set of sound files in the same way as for a point sound (page 99).

Since it doesn't have an aural direction, stereo sound is officially permitted for an ambient sound domain.

Overlay

> Scene.(Terrain).SoundDomains.♫ *name*: Overlay: numeric slider from 0.0 – 1.0
>
> No known purpose.
>
> Default: 1.0.

Saber claims that the Overlay property determines the volume mix between the ambient sound domain and the map-wide sound domain (page TBD), but I don't hear any difference. I suspect that it doesn't do anything.

Volume and Fading Distance

> Scene.(Terrain).SoundDomains.♫ *name*: Volume: numeric slider from 0.0 – 1.0
>
> Specifies the maximum volume of the sound as a fraction of its volume in the sound file.
>
> Default: 1.0.

> Scene.(Terrain).SoundDomains.♫ *name*: Fading distance: numeric, in meters
>
> Specifies the approximate distance from the edge of the shape through which the sound fades to zero volume.
>
> Default: 5.

The sound plays at the specified volume anywhere within the defined shape. Outside this shape, the sound attenuates to zero approximately at a distance given by Fading distance (in meters).

The actual fading distance is shown by the more transparent outer portion of the the polygon representing the sound domain in the main panel. There's some weird stuff going on with the way it handles corners, but what you see pictured is where the sound will play in the game, corners and all.

Bug: If Fading distance is zero, the game will hang when it tries to load the map.

Keep in mind that the "microphone" for the sound is in the game's camera, not in the truck. Whether the sound plays and its volume is determined solely by the camera's 2-D position, regardless of height.

Sound Conditions

> Scene.(Terrain).SoundDomains.*name* : Conditions: string
>
> Specifies the conditions required to play the sound.
>
> Default: blank; the sound is played unconditionally.

You can set an ambient sound domain to play only during the day or only at night by setting the Conditions value in the same way as for a point sound (page 101).

One-Shot Sound Domains

A one-shot sound domain is similar to an ambient sound domain, but it plays each sound in a different random direction from the player and at a different random volume. A one-shot sound domains is rather awkwardly implemented, and its behavior doesn't match what is described in Saber's documentation. It seem best suited to short sounds of only a few seconds duration that are well separated in time from each other. Saber uses the sound of a rockfall as an example.

To create a one-shot sound domain, choose Add Oneshot Sound Domain from the context menu.

The Editor lumps the new one-shot sound domain in the SoundDomains feature category along with other types of sound domains. A one-shot sound domain uses a sound wave icon next to the name: .

Name

> Scene.(Terrain).SoundDomains. *name*: Name: string
>
> Specifies the name of the sound domain to display in the Scene View at all times and in the main panel when the sound domain is selected.
>
> Default: blank; displays as (Oneshot Domain).

To help keep your sounds straight, you can assign a name to each sound domain in its Name property. The name has no game purpose, however.

Sound Domain Location and Shape

> Scene.(Terrain).SoundDomains. *name.n*: Position.X, Y, Z: numeric, in meters
>
> Specifies the position of the selected vertex.

The one-shot sound domain location and shape can be manipulated in the same way as for an ambient sound domain (page 104).

Sound Files

> Scene.(Terrain).SoundDomains. *name*: Sound file: string
>
> Specifies the relative path and base filename of the sound file(s) to play.
>
> Default: blank; triggers a File not found error when the map is packed.

Choose a sound file or set of sound files in the same way as for a point sound (page 99).

Stereo sound is officially permitted for an one-shot sound domain.

Volume Range

> Scene.(Terrain).SoundDomains.*name*: Volume.Min: numeric slider from 0.0 – 1.0
> Scene.(Terrain).SoundDomains.*name*: Volume.Max: numeric slider from 0.0 – 1.0
>
> Specifies the range of values for the random maximum volume chosen for each sound playback.
>
> Default: 0.4 – 1.0.

The game appears to pick a random value between Min and Max even if Min > Max. You should stick to Min ≤ Max, however.

Radius Range

> Scene.(Terrain).SoundDomains.*name*: Radius.Min: numeric, in meters
>
> Specifies the approximate distance from the edge of the domain shape at which the sound remains at the maximum volume.
>
> Default: 10.

> Scene.(Terrain).SoundDomains.*name*: Radius.Max: numeric, in meters
>
> Specifies a distance over which the sound decreases in volume.
>
> Default: 35.

The sound never entirely goes away, even at distances much beyond Radius.Max. The sound only stops playing at the end of the sound file. However, smaller Max values cause the volume to fall off more quickly than larger values.

If Min > Max, the game doesn't crash, but I have even less idea of what it's actually doing with the numbers. You should probably stick to Min ≤ Max.

The absolute aural direction of the sound (e.g. northeast) appears to be chosen randomly for each sound, and it stays the same throughout playback no matter where the player moves the truck and camera.

Delay Between Sounds

> Scene.(Terrain).SoundDomains.*name* : Interval.Min: numeric, in seconds
> Scene.(Terrain).SoundDomains.*name* : Interval.Max: numeric, in seconds
>
> Specifies the range of values for the random pause after each sound before the next sound is played.
>
> Default: 10 – 15.

After playing a sound, the game inserts a pause of random length before playing the next sound. If the Interval.Min value is 0, then sounds can be played back to back. However, this seems to cause poor volume mixing as the game transitions from the old sound to the new sound. I recommend a Min value of at least 1.

If the Interval.Max value is also 0, then sounds are always played back to back continuously.

The game appears to pick a random value between Min and Max even if Min > Max. You should stick to Min ≤ Max, however.

Weather Intensity Threshold

> Scene.(Terrain).SoundDomains.*name*: Weather intensity threshold: numeric slider from –1.0 to 1.0
>
> No known purpose.
>
> Default: –1.0.

Saber recommends leaving this property value at its default of –1.0.

Sound Conditions

> Scene.(Terrain). SoundDomains.*name* : Conditions: string
>
> Specifies the conditions required to play the sound.
>
> Default: blank; the sound is played unconditionally.

You can set a one-shot sound domain to play only during the day or only at night by setting the Conditions value in the same way as for a point sound (page 101).

Terrain Height

Clicking any of the items under (Geometry) in the Scene View allows you to edit the terrain. Terrain features don't have named properties, and they are not saved in the map's XML file. Instead, the terrain is divided up into small squares, and data for each square is recorded as a pixel in a bitmap. The "brightness" for each pixel in the bitmap determines the strength of a corresponding feature in the terrain. For example, if the terrain height increases smoothly from minimum height on the left to full height on the right, it is recorded in the height bitmap as image that fades from black on the left to white on the right.

This concept of terrain features as bitmaps is reflected in the metaphor for editing those features: painting the terrain using a brush.

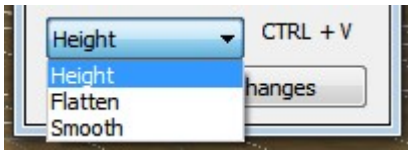
There are many features within the (Geometry) category, but this section is devoted to the (Geometry) feature itself.

Click (Geometry) under (Terrain) in the Scene View to edit the terrain height. A Brush dialog pops up that allows you to edit your brush parameters. Then draw on the map by holding down the right mouse button while moving the mouse (right-click and drag).

If you are satisfied with the results, left click in the main panel or on something else in the Scene View to deselect the (Terrain) tool. This commits your changes and writes them to the underlying bitmap file. (No separate save step is necessary for terrain changes.)

On the other hand, if you don't like your changes, click Undo Current Changes in the Brush dialog to revert your changes without updating the underlying bitmap file. Changes are reverted across all brush modes back to the point when you first entered the (Geometry) brush. You can then try drawing again or left click to exit drawing mode.

The (Geometry) brush has three modes: Height, Flatten, and Smooth. Each of these modes is described in the sections below.



Height Brush Mode

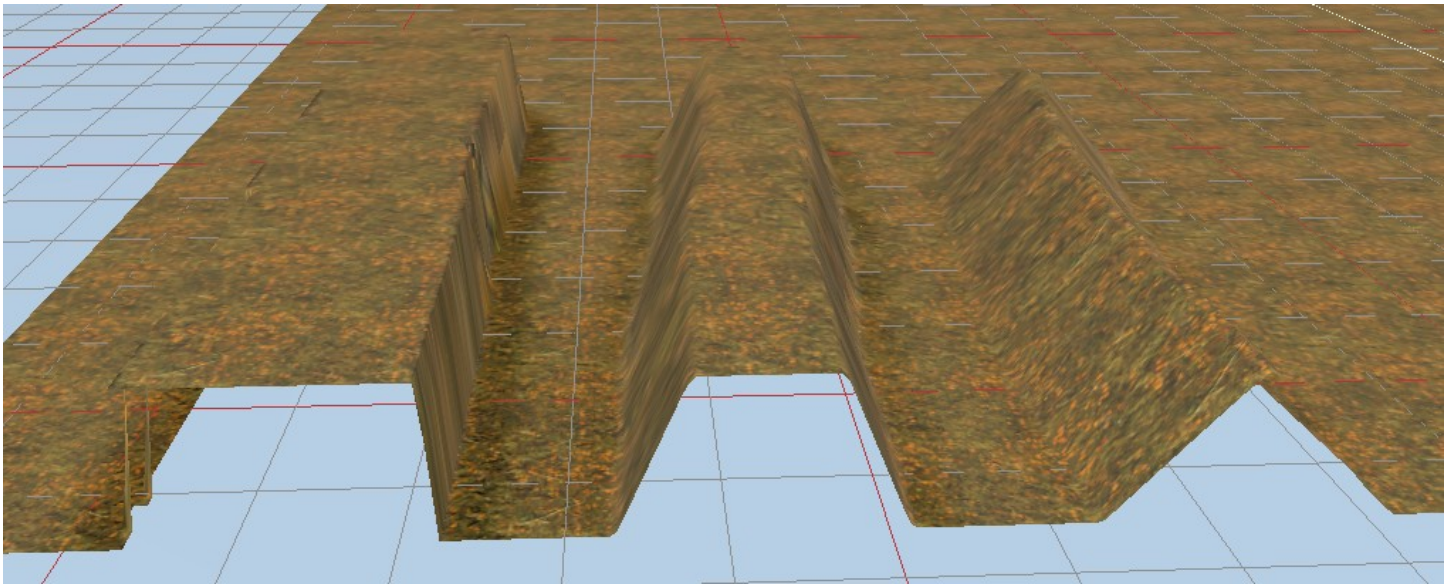
The default Height mode changes the terrain height directly.

The Size parameter in the Brush dialog sets the radius of the brush in meters. The slider is approximately logarithmic, so you have fine control of small brush sizes and coarse control of large brush sizes.

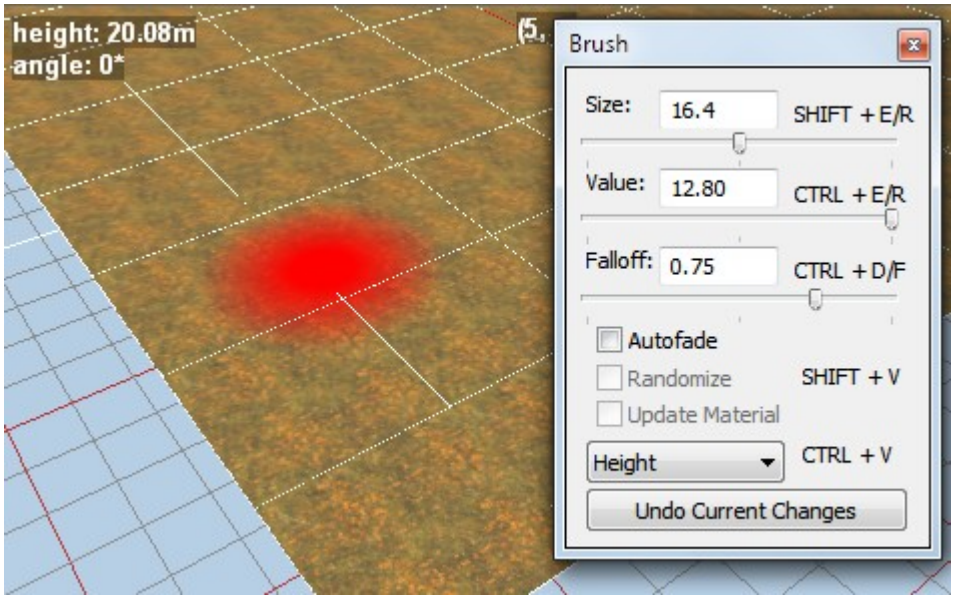
The Value parameter sets how much the terrain height changes at the center of the brush during a paint stroke. The value is measured in meters. Positive values increase the terrain height, and negative values decrease the terrain height.

The brush has an inner region in which the terrain height is changed by the full Value and an outer region where the change falls off to zero. The Falloff parameter determines the fraction of the brush radius that is dedicated to the outer brush region, where the change falls off. With a Falloff of 0.00, there is no outer region, and the entire brush paints the terrain with the full change set by Value. With a Falloff of 1.00, falloff is spread across the entire brush, so only the point at the center paints the terrain with the full change. In most cases, some intermediate value is desirable.

The figure below shows the effect of different Falloff values, with 0.00 on the left, 0.50 in the middle, and 1.00 on the right.



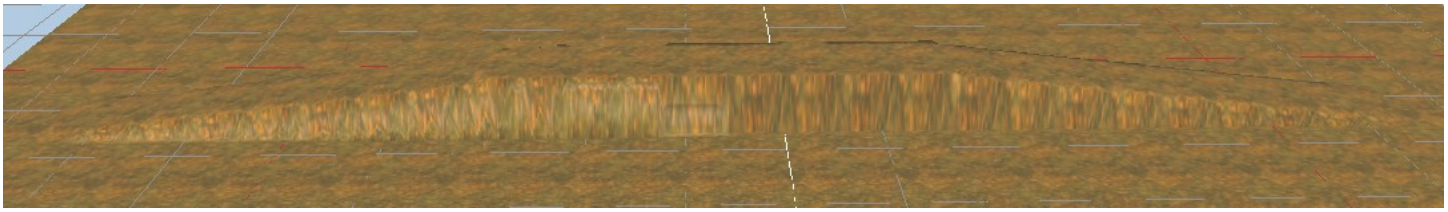
While in Height mode, a disc is drawn on the terrain under the mouse to preview the brush parameters. A red disc is drawn for a large positive Value, and a black disc is drawn for a very negative Value. The brush Size and Falloff are also represented on the disc. This provides a quick sanity check that your parameters make sense before you start to paint.



The disc is less strongly colored at reduced values. Unfortunately, when drawing with a small Value, the disc can be nearly invisible.

Autofade

The Autofade checkbox is mostly useful for drawing tire tracks in mud, not for drawing height, but you can experiment with it here. With Autofade enabled, when you right click and begin dragging in some direction, the editor adds a tail on your tint that fades in as if you had slowly lowered the brush onto the terrain while approaching the spot where you actually clicked. And when you let go of the right mouse button, the editor adds a tail that fades out as if you had slowly lifted the brush while continuing to drag in the most recent direction. The length of the tail is about 10x the width of the brush.



I find Autofade very difficult to control, but your milage may vary.

Autofade is available for (Geometry) when using the Height mode.

Flatten Brush Mode

Once you've painted some height changes onto the terrain, it is very difficult to get it flat again using the Height mode. To easily flatten terrain, change the Brush dialog to the Flatten mode. In this mode, the terrain at the point where you start your brush stroke determines the target height. Painting in this mode shifts the height of the painted terrain toward the target height.

The Editor keeps track of the most recent brush parameters you used in each mode. When you switch modes, the parameter values are reset to whatever you last used in that mode.

The Size parameter still has the same purpose; it sets the radius of the paint brush.

The Value parameter is now a value between 0 and 1, and it determines how quickly the painted terrain should change toward the target height. With a Value of 1.00, the terrain at the center of the brush jumps fully to the target height. With a lesser Value, the terrain under the brush shifts only the specified fraction toward the target height. But this is deceptive! For every pixel that you drag the brush, the Editor re-applies the smoothing effect across the entire brush. For example, with a Value of 0.50, as you start dragging the brush, the terrain near the center of the brush, shifts halfway to the target value, then shifts hals of the rmaining amount, etc. This is very different than the Height mode, where the brush effect is re-applied only if you start a new stroke.

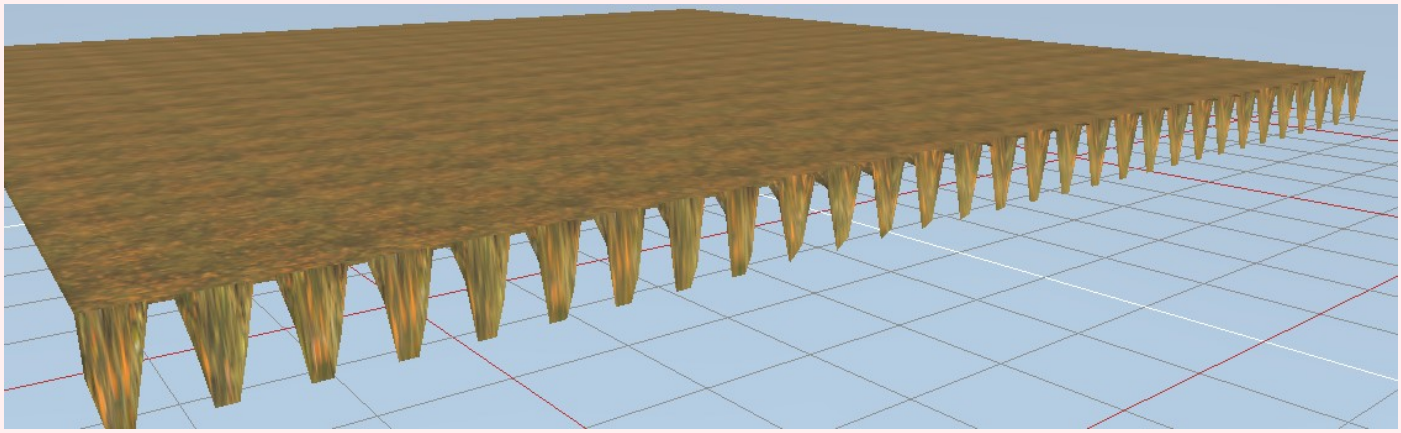
The Falloff parameter still has the same purpose; it sets how much of the brush is dedicated to the falloff region.

If you just want everything to be perfectly flat, it is fine to set the Value to 1.00 and the Falloff to 0.00. However, if you want a more subtle effect (e.g. at the edges of the flattened region), I recommend a reduced Value and a higher Falloff. For example, a Value of 0.50 and a Falloff of 0.80 allows me to stroke overlapping loops with my brush that result in perfectly flat terrain where I've been looping and smooth slopes at the edges.

In the Flatten mode, the brush preview disc is drawn in red for a Value above 0.50, black for a Value below 0.50.

Bug: Normally the brush preview disc increases in intensity as the Value changes away from some neutral value that has no effect when painted. In Flatten mode, however, the brush preview disc treats 0.5 as the neutral Value when 0.0 is the actual neutral Value, This means that the preview disc is invisible near a Value of 0.50 even though this Value causes a large effect.

Bug: A freshly created map is mostly at a single middling height, but one edge often has some weird height drops. Use Flatten mode to fix it.



Smooth Brush Mode

In the Smooth mode, the target height at each point under the brush is a weighted average of the heights around the edge of the brush. E.g. the target height in the middle of the brush is exactly the average of the heights around the edge of the brush, but the target height near the left side of the brush is closer to the height of the left edge than the right edge. In other words, Smooth mode tries to create a smooth slope.

The Size parameter still has the same purpose; it sets the radius of the paint brush.

The Value parameter indicates the magnitude of the effect. In this case, a Value of 1.00 does not fully smooth the terrain as soon as you click, but it takes very little dragging of the brush to achieve a completely smooth slope.

The Falloff parameter again specifies the falloff characteristics of the brush. In this mode, however, falloff is counter-productive. The results can be very non-smooth if terrain is fully smoothed at the center of the brush while the terrain at the edges stays close to its original heights.

Because of the way the Smooth brush works, it is perfectly reasonable to set the Value to 1.00 and the Falloff to 0.00.

Bug: Normally the brush preview disc increases in intensity as the Value changes away from some neutral value that has no effect when painted. In Smooth mode, however, the brush preview disc treats 0.5 as the neutral Value when 0.0 is the actual neutral Value, This means that the preview disc is invisible near a Value of 0.50 even though this Value causes a large effect.

Brush Keyboard Shortcuts

Keyboard shortcuts are listed on the Brush dialog box. E.g. shift+E decreases the Size by a step, and shift+R increases it by a step. Each step is 5% of the slider length, but remember that the slider has an exponential relation ship to the Size value.

Shift+V is for the Randomize checkbox, which is disabled for all (Geometry) brush modes.

Ctrl+V rotates among the (Geometry) brush modes: Height, Flatten, and Smooth.

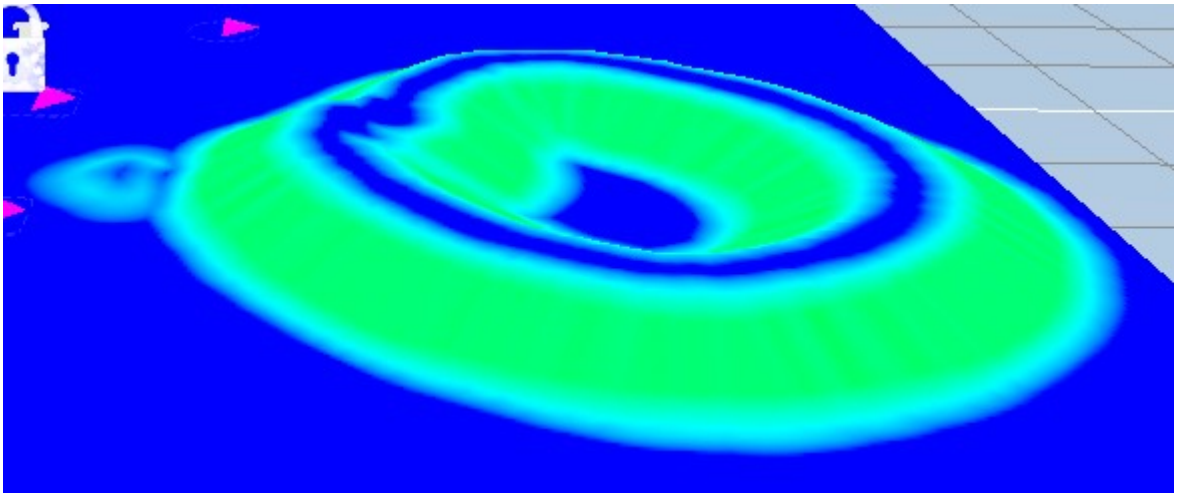
Ctrl+Z is an unlisted shortcut that corresponds to the Undo Current Changes button.

The keyboard shortcuts can be used not just when the Brush dialog box has focus but also when the main panel or the Scene View have focus. In other words, they can be used essentially any time you are using a brush. If you learn and use these shortcuts, your painting speed will be much faster with all brushes.

Height Information

Whenever the mouse is over the terrain in the main panel, the editor annotates the current terrain height and angle in the upper left corner. The angle is 0° for flat terrain, increasing to near 90° for a cliff.

You can get a quick read of the terrain angles at all points at once by enabling Show HeightMap Angles in the toolbar.



The heightmap colors correspond as follows:

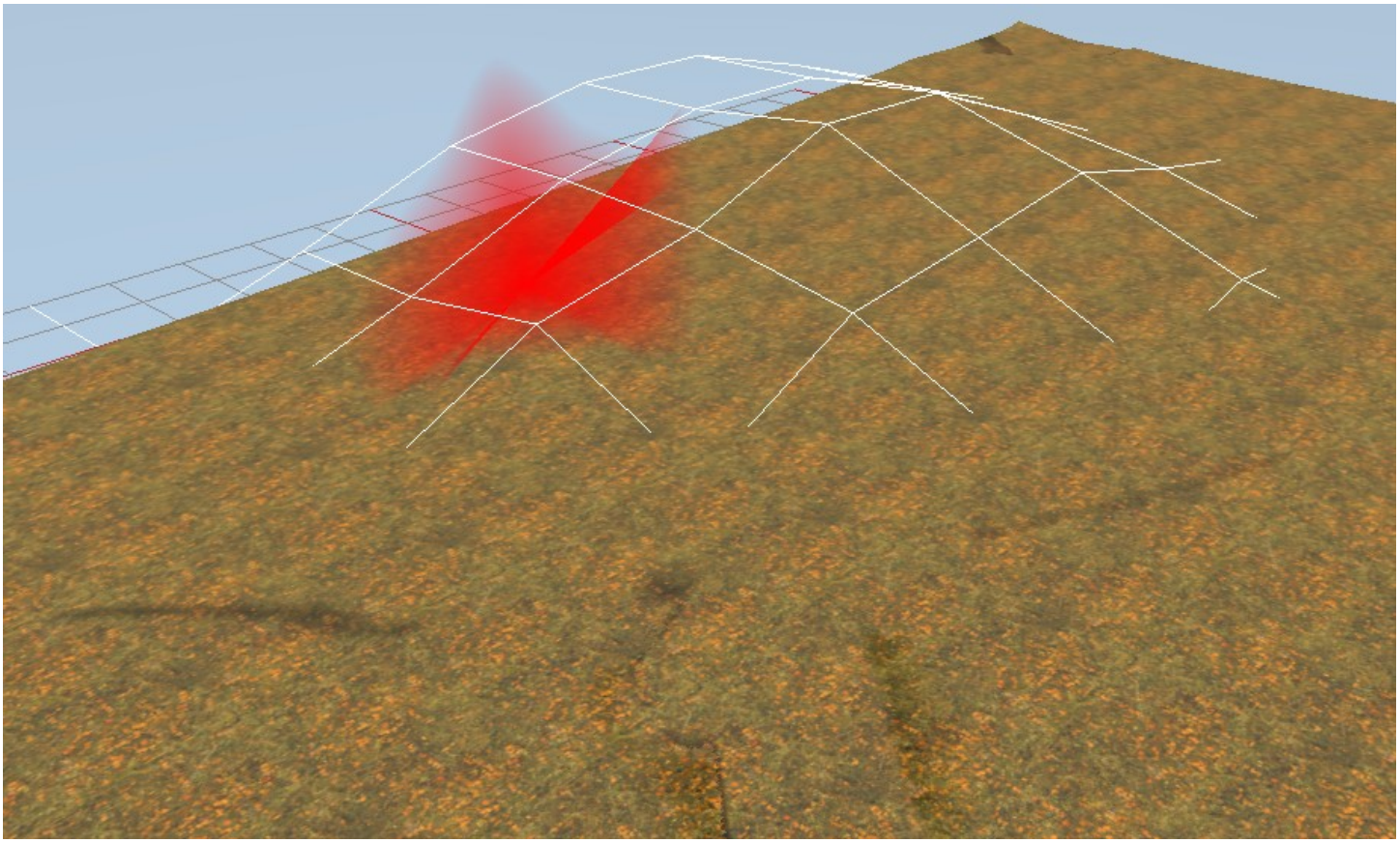
- dark blue: 0°
- cyan: 1° – 24°
- green: 25° – 34°
- yellow: 35° – 44°
- orange: 45° - 59°

- red: 60° - 90°

In ideal conditions in SnowRunner, a well-prepared stock can climb a slope up to about 60°. For comparison, according to the Guinness World Records, the steepest street in the world is 19°.

Deceptive Height Cues

While editing the terrain height, the Editor changes the way it draws the terrain. However, it doesn't change the way it draws everything else until you commit your changes and rebuild the terrain. For example, I have (almost) completely flattened the terrain in the screenshot below, but it doesn't look flat at all.



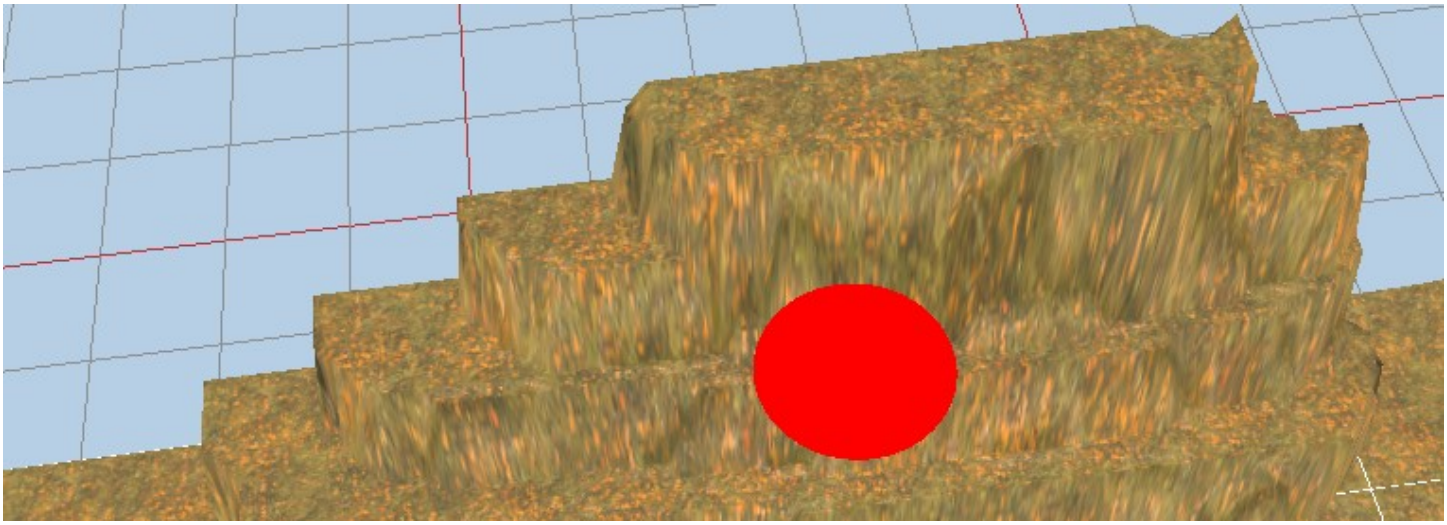
The deceptive cues above are as follows:

- The terrain is still colored with its original shadows, showing where steep slopes used to exist.
- The base of the terrain blocks (the white wireframe mesh) remains at its original height.
- The brush preview disc is warped to follow the contours of the original terrain.

Another deceptive cue not shown here is the height and angle displayed in the upper left of the main panel, which continue to show the values for the original terrain.

The only cue that isn't deceptive is the edges of the map. The map edges are straight, which implies that the terrain must be flat (or at least smooth). A couple of hitches near the corner imply that I didn't quite get it flat there.

The brush preview disc is particularly deceptive because its horizontal position is determined by where I'm pointing on the original terrain, which can be very different from the horizontal position where I'm pointing in the edited terrain. In the screenshot below, I've built up the terrain from a flat plane. The brush preview disc appears to be pointed at the middle of the slope, but it actually represents a position right at the edge of the map. (Compare the disc position to the virtual line connecting the flat edges of the terrain at the left and right.) Drawing at this position will actually draw at the very edge of the map at the top of the slope.



Rotating the camera and watching the motion of the terrain can help a little. But really the best solution is to commit your big changes, see how they look, and then make smaller changes to fix up the details.

Navigate While Painting

While right-click-dragging, it can be frustrating to run into the edge of the main panel. If the mouse leaves the main panel and re-enters somewhere else, the brush paints a straight stroke from where it left to where it re-entered.

To avoid this, you can navigate while a brush stroke is in progress. While continuing to hold down the right mouse button, you can then press the control key and the left mouse button to switch to dragging the terrain around. If you then release the left mouse button while continuing to hold the right button down, you can resume your paint stroke at the same position on the terrain, but a new position within the main panel.

This trick only works well with control + left-drag because the mouse moves exactly with the terrain. With a regular left-drag or with shift + left-drag, the terrain moves away from the mouse, so you end up with a stray brush stroke when you release the mouse.

Restore a Previous Terrain State

Click Undo Current Changes in the Brush dialog to revert your changes without updating the underlying bitmap file. Changes are reverted across all brush modes back to the point when you first entered the (Geometry) brush. However, the Editor doesn't keep any further backups of previous files.

I recommend that if you're happy with your terrain, manually make a copy of the prebuild directory to an archival location. Then, if you accidentally mess something up, you can simply restore one or all of the archived terrain bitmap files back into your prebuild directory.

Unlike when you update the map's XML file, the MudRunner Editor won't automatically notice changed bitmap files. Right click in the main panel and select "Rebuild Terrain" to redraw the map from the files.

Mud

Mud is tricky to get right. Expect to experiment a lot to get a feel for it.

Note that many of the terrain materials are already a little soft, including the default dirt material. This means that if a truck drives over it repeatedly, it will eventually cut ruts into the terrain and lose traction. Adding mud makes the material even softer and more slippery.

Mud can be painted in four different ways:

- The (Geometry) → (Mud) brush in Extrudes mode paints mud that is hard to see.
- The (Geometry) → (Mud) brush in Automatic mode can paint wide swaths of visible mud.
- The (Geometry) → (Mud) brush in Automatic mode can also paint narrow muddy ruts that look like a truck drove through.
- The (Geometry) → (QuickMud) brush paints a wide area of mud that appears churned by many trucks.

Extrudes Mode

Expand the (Geometry) category in the Scene View and click (Mud) to start using the mud brush. The default mode is Automatic, but we'll start with the Extrudes mode.

The mud brush allows painting with relatively fine detail. For a particular Size value, a mud brush is half the size of any other terrain brush. In other words, the mud brush size effectively measures the brush diameter, whereas all other brushes measure the brush radius.

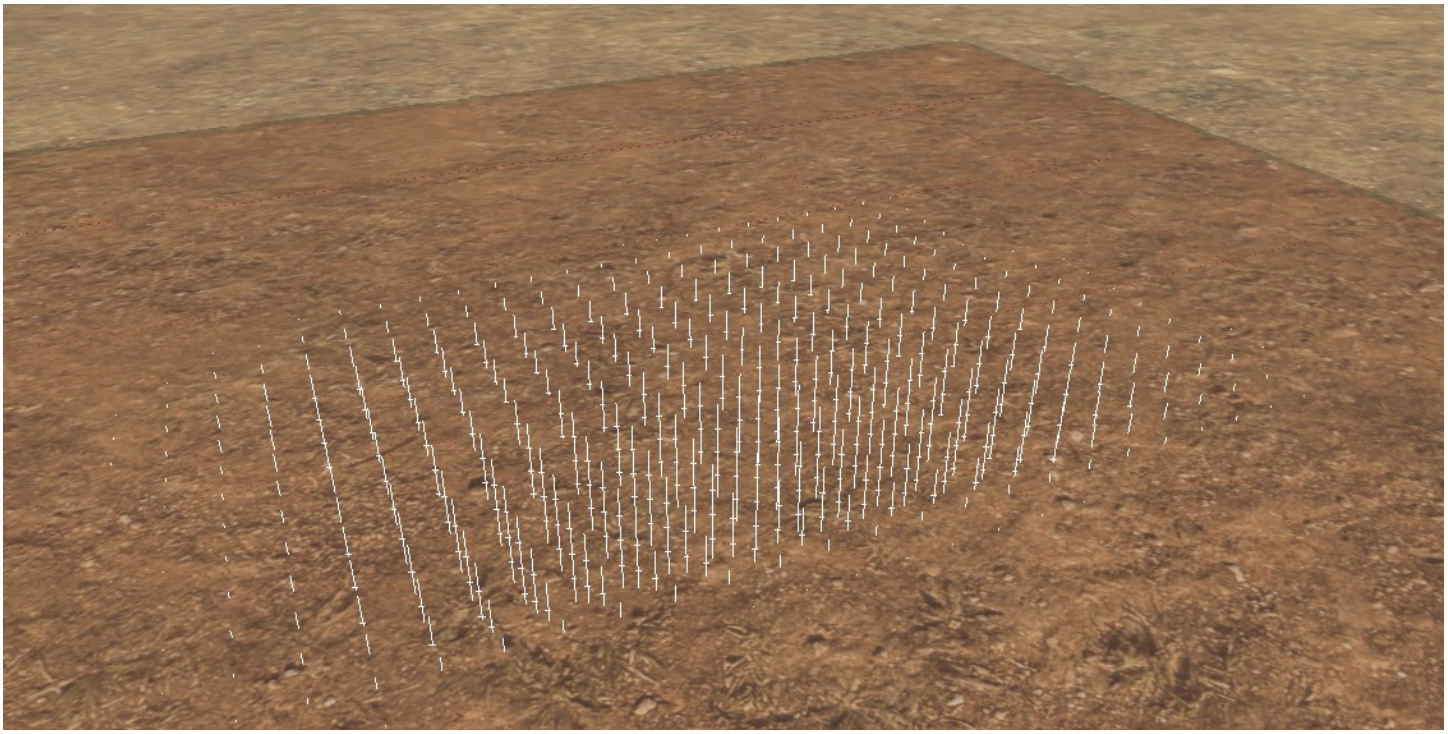
A Value greater than 0.50 increases the amount of mud, and a Value less than 0.50 decreases the amount of mud. When increasing mud in Extrudes mode, the brush preview disc is red. When decreasing mud, the brush preview disc is black.

Bug: Using 0.50 as the neutral Value is a holdover from SpinTires and MudRunner where all brushes used the same slider range. It would make more sense to have a range of –1.00 to +1.00 with a neutral value of 0.00. This neutral value of 0.50 is common to many of the brushes, so you'll eventually get used to it.

Adding mud with the Extrudes brush softens the terrain in the painted area, making it easier for a truck to sink into it. By default, mud painted with the Extrudes brush is completely invisible. However, while the mud brush is active, the Editor indicates the presence of mud with two visual cues:

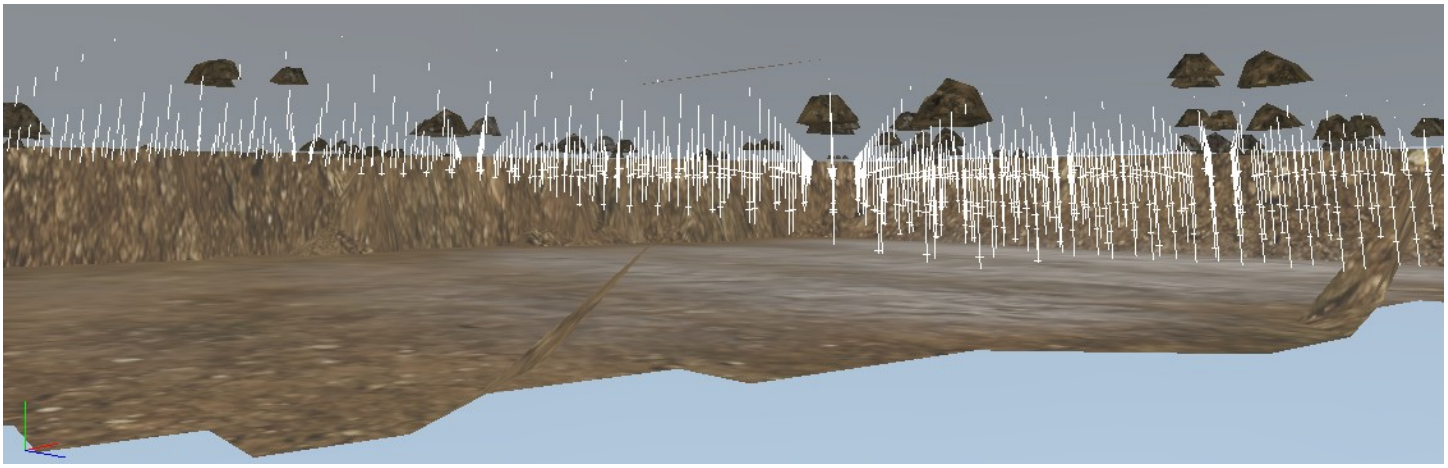
- A brown tint covers a large block of terrain, indicating that mud is present.

- Thin white lines extend down from the terrain surface where mud wsa painted. The length of each line represents the softness of the mud in that area. A tick mark is placed on the line for every 25% of the maximum mud softness that is present. E.g. if a line has one tick mark near the bottom, the mud is just over 25% of the maximum possible softness. This is generally passable even by 2WD vehicles with highway tires. Two tick marks is about 50% of the maximum softness, and is generally passable by AWD vehicles with offroad tires. Beyond that, only the toughest, most mud-capable trucks will find passage.



If you left click to exit the mud tool, there is (by default) no visual indication of the mud at all. This is true within the game as well. I'll address the visibility of Extrudes mud in the next section.

While a mud brush is activated, you can move the Editor camera slightly underground to see a bit more about how mud is handled in the game. Wherever mud is present, a second terrain layer is added beneath the main surface. This second layer provides another surface for both visual rendering and for physics when the truck breaks through the main surface and starts churning up the mud.



Bug: The first time you paint a new section of mud, the main surface is temporarily erased and only the second, underground layer is rendered. If you exit the mud brush and the re-select it, the main surface is drawn correctly.

This second terrain layer highlights an important point: although the thin white lines imply a mud depth, they really only indicate its softness. If a heavy truck persists in churning up the mud, it can always dig all the way to the second layer.

In fact, the game automatically creates a second layer wherever the truck starts churning up dirt. With enough persistence, it is possible to dig deep ruts even without mud.

Tip: Avoid sudden transitions between deep mud and no mud. The truck will bounce as if it hit a low wall as it is suddenly forced up and over the solid dirt.

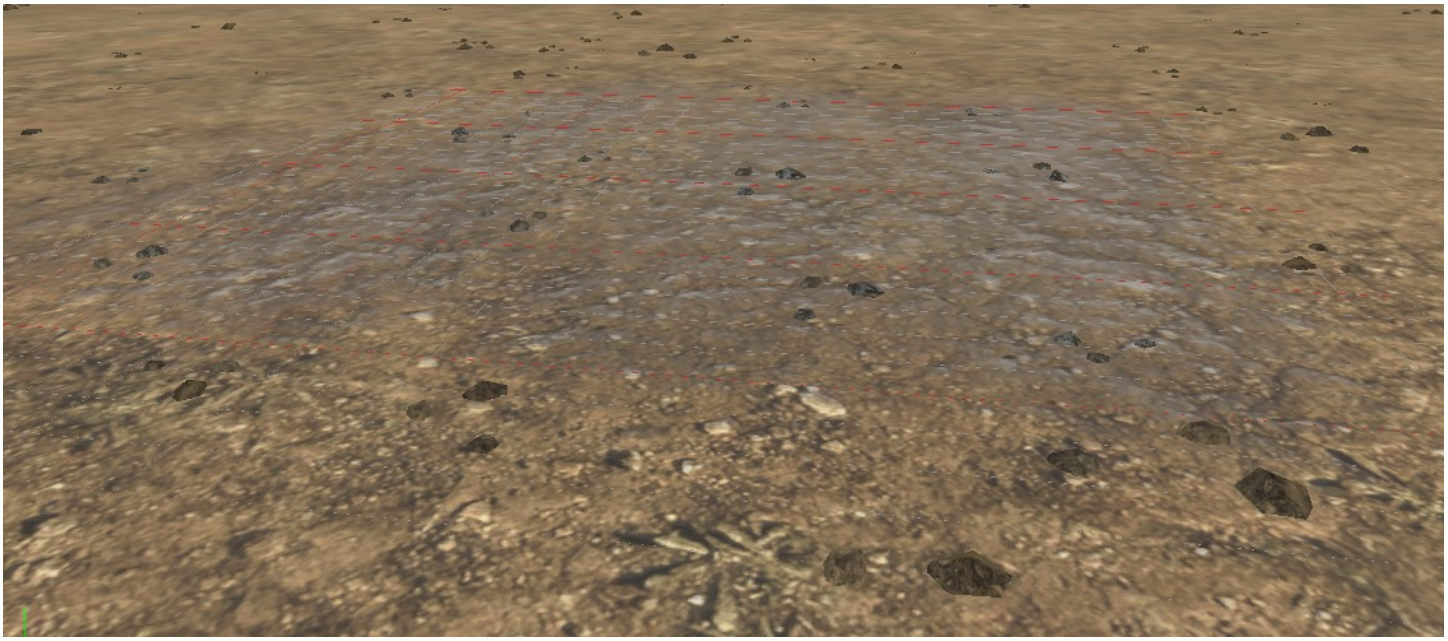
Extrudes to Wetness

The (Terrain) feature in the Scene View has a property called Extrudes to Wetness with a default value of 0.0. This property causes wetness to be added to the terrain in proportion to the Extrudes softness. Wetness provides a subtle visual indication that mud is present. It also makes the mud more slippery.

The Extrudes to Wetness value is multiplied by the mud softness (as a fraction of the maximum mud softness) to get the derived wetness (as a fraction of the maximum wetness). The resulting calculated wetness is capped at 100% if necessary.

Tip: Saber recommends an Extrudes to Wetness value of 1.0.

Newmud painted in Extrudes mode is rendered with the current Extrudes to Wetness value. But to apply the parameter to existing mud you must select Rebuild Terrain from the context menu.



A second property just below Extrudes to Wetness sets the -threshold at which it applies. Mud that is less deep than the -threshold value (as a fraction of the maximum mud softness) has no wetness applied. The wetness is unaffected for mud that is deeper than the threshold.

Tip: Saber recommends a -threshold value of 0.20 for a winter environment and a smaller value such as 0.05 for a warmer environment.

Automatic Mode

The (Mud) brush also has an Automatic mode that is used to draw obvious mud. Its behavior is different if the Size is greater than 2.5 meters vs. less than or equal to 2.5 meters.

As in the Extrudes mode, a Value greater than 0.50 increases the amount of mud, and a Value less than 0.50 decreases the amount of mud. When increasing mud in Automatic mode, the brush preview disc is white. I'll describe decreasing (erasing) Automatic mud in a later section.

The Falloff parameter is disabled in Automatic mode because paint strokes have their own unique behavior in this mode.

Automatic Mode With Size > 2.5

Automatic mode is called "automatic" because the brush does multiple things at once. With Size > 2.5 it does the following.

- It increases the softness of the mud in the same way as the Extrudes brush.
- It adds a dark tint to the painted area to indicate where subsurface soil has been exposed.

Since the Automatic mode records mud softness in the same way as the Extrudes mode, the Extrudes to Wetness property applies to both. So for mud that is deep enough, it can have both the dark tint of subsurface soil, plus additional tint and shinyness from wetness.

Automatic Mode With Size ≤ 2.5

With size less than or equal to 2.5 meters, the Editor adjusts Automatic mode to paint a rut as if a truck drove through. This increases the softness of the mud and adds a dark tint as above, and it also adds lumpiness to the surface where the mud is pushed down in the center of the rut and pushed up slightly on each side of the rut.

Tip: Saber recommends a Size of 0.5 – 0.8 meters to represent a rut from a truck with narrow or wide tires.

Tip: Saber recommends a Value of 0.54 – 0.7 to represent the range from very shallow to very deep ruts.

Tip: A pair of parallel ruts best simulates the passage of a single truck (or of multiple trucks driving the same path). Position a truck of the appropriate size near the ruts to gauge how far apart they should be.

Tip: The Autofade checkbox is well suited for painting a rut that begins and ends with a smooth transition. However, it takes a steady hand to get the Autofade tails to be parallel for each pair of ruts.

With the recommended values, the mud in the ruts is not particularly soft, and it won't generate much wetness from the Extrudes to Wetness parameter. A truck driving through these ruts is more effected by the lumpiness of the ruts than the softness of the mud. You can then use the Extrudes mode to increase the softness of the mud and/or manually add wetness to make it more slippery.

Decrease Automatic Mud

Mud can be decreased or erased with a brush Value less than 0.5. The effect is different depending on whether the brush is in Extrudes or Automatic mode.

The Extrudes mode only directly effects the mud softness. The lumpiness and dark tint are unaffected until the mud softness reaches zero, at which point the lumpiness and tint are removed. However, the lumpiness and tint remain recorded and will return if the Extrudes mode is used to soften the mud again.

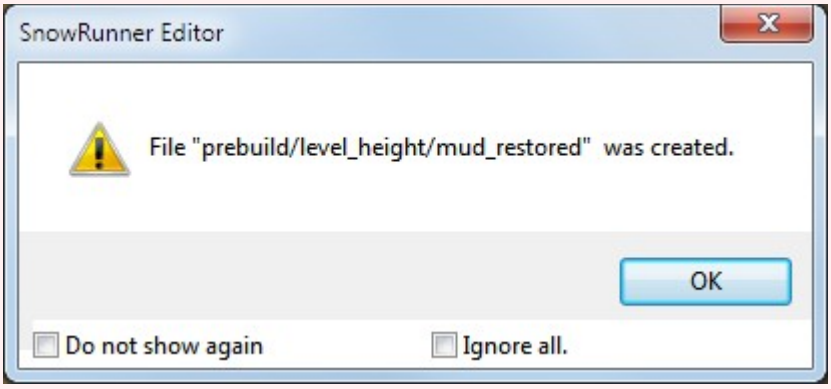
The Automatic mode reduces the mud softness, its lumpiness, and the dark tint simultaneously. However, it does a really bad job reducing the lumpiness and tint. Depending on how you apply the brush, you can end up removing the tint while leaving the lumpiness, or remove the deep pat of the rut while leaving the pushed up parts, etc. If you aren't satisfied with the visual appearance of a rut, it is generally easier to entirely erase it and start again. Use Automatic mode with Value 0.0 and a Size large enough to entirely cover the affected area. This also erases the mud softness, so you'll need to reapply that with the Extrudes mode if desired.

Note that the lumpiness from the mud is recorded separately from the base terrain height. Erasing the mud restores the original terrain appearance.

Restore a Previous Mud State

As in other terrain brushes, you can click Undo Current Changes in the Brush dialog to revert your changes without updating the underlying bitmap file. Changes are reverted across all brush modes back to the point when you first entered the (Mud) brush.

Bug: All (Geometry) items in the Scene View have a context menu item called Restore version "mud" from "data.stg". Since data.stg is only updated on a save, this appears perfect, but unfortunately it doesn't work. Selecting the action pops up a dialog to indicate that it is done:



If I replace my mud file with the new mud_restored file and reload the map, it appears to have worked (although with some leftover wetness tints that are presumably stored separately). However, rebuilding the terrain reverts back to the unwanted mud state. I suspect that the "restore" process discards the unsaved data.stg in memory and reads the saved data.stg from disk, but it doesn't actually restore the mud file properly.

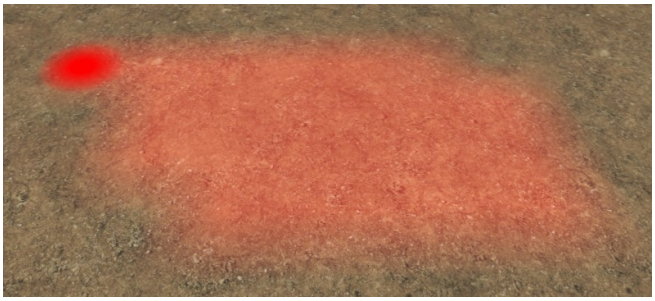
I recommend that if you're happy with your mud, manually make a archival copy of your map in progress. Then, if you accidentally mess something up, you can simply copy the archived mud file back into your prebuild directory.

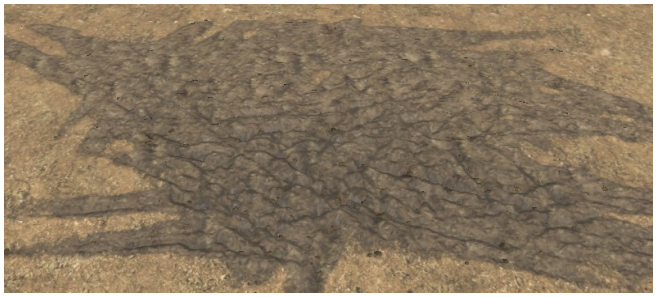
QuickMud

Expand the (Geometry) category in the Scene View and click (QuickMud) to quickly paint a large swath of heavily churned mud. This feature works differently than other terrain brushes because you don't have direct control of the results. Instead, you simply indicate the areas where you want mud, and the Editor decides how it should look.

A Value greater than 0.50 increases the amount of mud, and a Value less than 0.50 decreases the amount of mud. When increasing mud, the brush preview disc is red. When decreasing mud, the brush preview disc is black.

The figures below show the act of painting with the QuickMud brush and the result.





Essentially, you paint an area with a red tint, and the Editor creates many pairs of truck ruts at random orientaions in order to churn that area (and somewhat beyond, since it uses autofade).

The mud generated by the QuickMud brush is kept separate from the mud created by the Mud brush, so you can't directly observe the softness of the generated mud. Howeer, experimentation reveals that where the QuickMud tint is brightest, the mud randomly ranges in softness from 0% to 35% of the maximum softness. Where the QuickMud tint is weaker, the maximum softness is reduced.

The brightness of the QuickMud tint also appears to have a subtle effect on the visual depth of the generated ruts, but I'm not completely positive about that. At the very weakest tints, QuickMud generate light patches of mud tint without any ruts.

Materials

The PbrMaterials category in the Scene View holds one or more materials that make up the surface of the terrain. These materials include surfaces such as grass, dirt, gravel, sand, asphalt, and snow. Each possible surface is refered to as a texture layer.

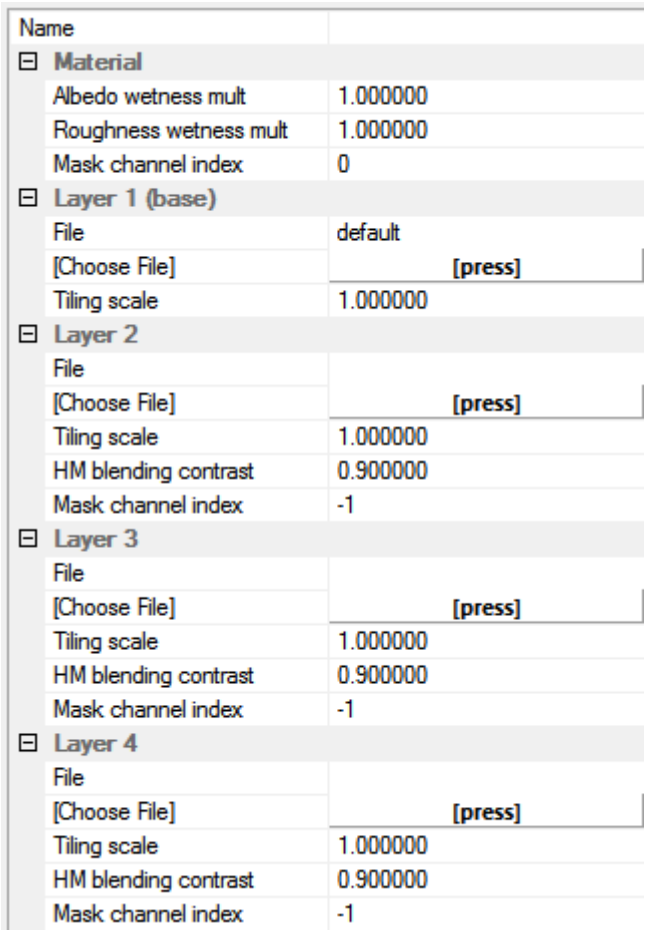
A texture layer obviously describes the texture of the terrain, i.e. the bitmap that is applied to the terrain. It also enables associated 3-D objects such as grass and pebbles to set on the surface of the terrain.

The texture layer also describes intangible aspects that can’t be seen in the Editor, e.g. whether dust can be kicked up from the terrain and what it sounds like when tires skid on the terrain. For ease of description, I’ll mostly refer to the materials’ impact on the texture of the terrain. The impact on other aspects is implied.

One terrain block (24m×24m square of the terrain) can use only one material, but that material may include up to four texture layers arranged however you like across that terrain block. I’ll begin by describing how texture layers are handled in one material before we start adding more materials.

Material Properties

Your map is initialized with its first material, called (PBR Terrain Material) under PbrMaterials in the Scene View. Click on it to bring up its properties. This also brings up a brush tool, but we'll deal with that in the next section.



The Name property is blank, which is why this material has the default name of (PBR Terrain Material). You can enter any name you want for the Name property. This Name is displayed in the Scene View to help you keep track of the purpose of this material.

The Albedo wetness mult property influences how much that wetness darkens the tint of the terrain. A higher value leads to darker wetness. A value of zero removes the tinting effect of wetness.

The Roughness wetness mult property influences how wetness is displayed on the terrain. A higher value makes wet terrain look smoother. A value of zero removes the smoothing effect of wetness.

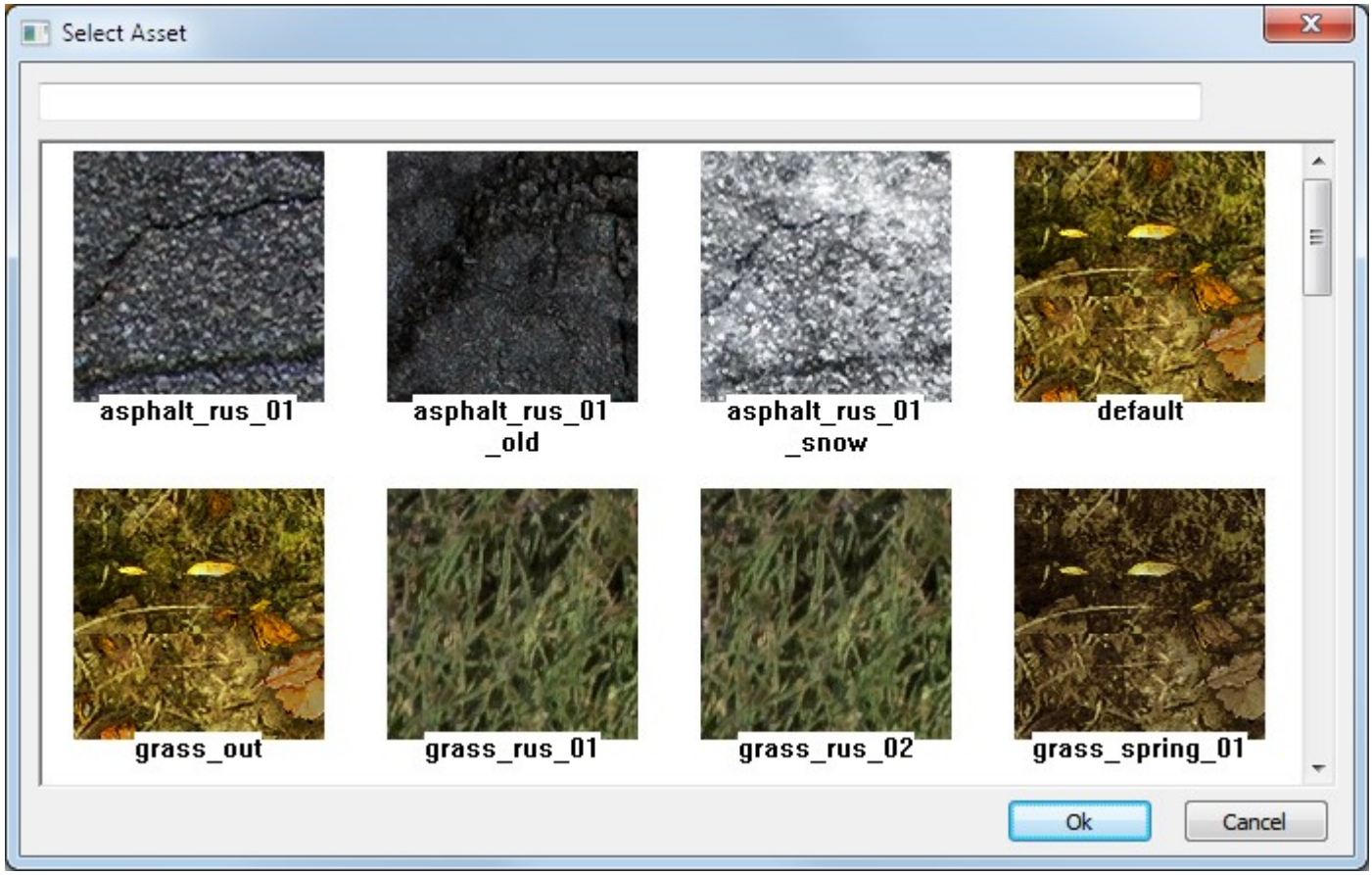
Tip: Saber recommends for winter environments AlbedoWetnessMult = 1.0 and RoughnessWetnessMult = 1.0. They recommend for warmer environments AlbedoWetnessMult = 1.0 and RoughnessWetnessMult = 1.0.

The Mask channel index cannot be edited and is always 0 for the first material.

Texture Layer Properties

The remaining properties for the material are specific to each texture layer.

Choose the texture for a layer by clicking [press] next to [Choose File]. Choose your desired texture from the dialog box that pops up.



By default, four copies of the texture are tiled in a 2×2 pattern to fill each terrain block. But for textures that require finer detail, we want smaller tiles, with more of them to fill a block. The Tiling scale property specifies how much smaller each copy of the texture should be. E.g. a value of 5 shrinks the texture by a factor of 5, so it takes a 10×10 pattern of tiles to fill a terrain block. The value does not need to be an integer. If the edge of a terrain block is reached in the middle of a tile, the tile continues smoothly into the next terrain block as long as it uses the same texture.

Bug: The proper Tiling scale is always the same for each particular texture, but instead of specifying the property with the texture itself, the Editor makes us specify the property every time we use a texture. If we forget to do that, the result can look very weird. E.g. enormous leaves in the default texture.

Tip: A good-looking Tiling scale is **not** 1.0 for most terrain layers. See the Texture Layer Reference section on page 237 for recommended values.

Layers 2 – 4 have additional properties:

When different textures within the same material are placed next to each other on the terrain, the edges of those textures are blended together to avoid a sharp demarcation. The HM blending contrast propety specifies the sharpness of that blend line. The effects of this property are described in more detail in the Edge Blending section on page 135.

Bug: The proper HM blending contrast should be specified where the texture is described, not where it is used.

Tip: Saber recommends the following HM blending contrast values for various textures:
- soft surfaces: 0.7 – 0.8
- hard surfaces: 0.8 – 0.9

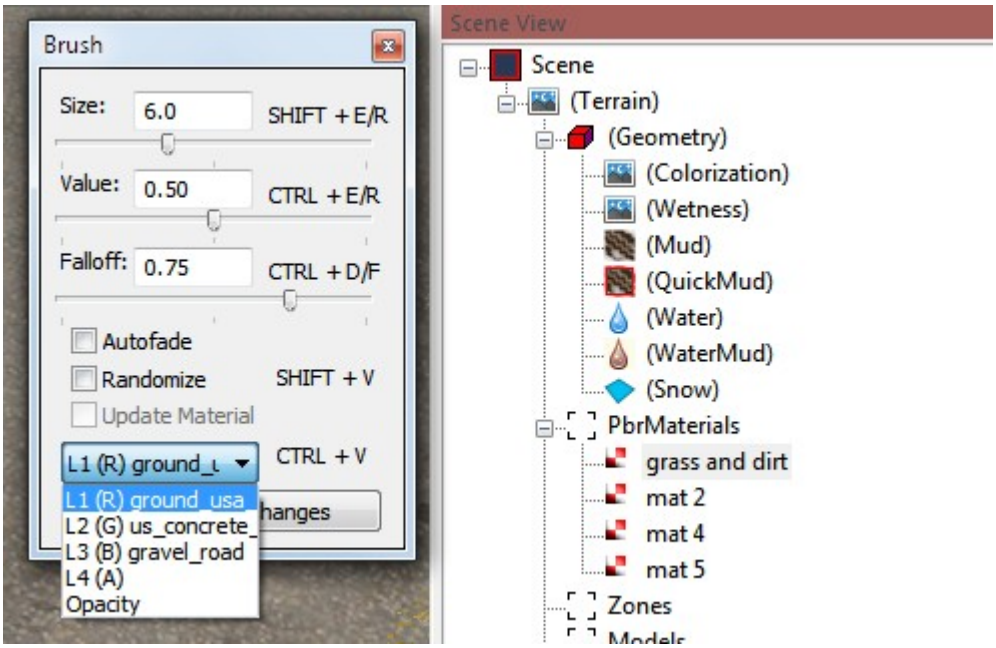
The Mask channel index cannot be edited and is always 2 less than the layer number. E.g. layer 3 uses Mask channel index 1. This per-layer property is separate from the per-material Mask channel index described above.

Paint a Texture Layer

Texture is painted onto the terrain in 0.5×0.5 meter squares. Each square typically displays the characteristics of a single texture layer, but some blending can occur as described in the sections below.

Textures are painted onto the terrain in layers like separate coats of paint. Each texture uses a stencil so that it is painted in some squares but in others. Whichever texture is painted last is most visible, but a weakly painted texture will allow color to bleed through from the underneath textures.

To begin painting, click the desired material under PbrMaterials, then select the desired texture in the mode selection dropdown of the Brush dialog box.



The bottommost texture is always painted across the entire terrain block, so it cannot be selected in the Brush dialog. Each of the three other textures in the material can be selected and painted over top.

Bug: The three other textures are labeled Layer 2, Layer 3, and Layer 4 in the property panel, but they are confusingly abbreviated L1, L2, and L3 in the Brush dialog.

The L4 (A) brush mode is used only for snow. The Opacity brush mode is used to select which material to apply. I'll describe each of these in later sections.

A Value greater than 0.50 increases the strength of the texture, and a Value less than 0.50 decreases the strength of the texture. When increasing the strength, the brush preview disc is red (R), green (G), or blue (B), depending on which texture is selected. When decreasing the strength, the brush preview disc is black.

Remember that the paint is layered, so if you paint a strong Layer 4 (L3), nothing you paint in the other layers will be immediately visible. However, if you then erase some of the paint from Layer 4, your changes to the lower layers become visible.

Value Blending

Where a texture layer is not painted at full strength, the display engine blends it with the lower texture layers. There is some blending at the pixel level, but mostly the engine looks for the implied high spots in each texture and gives them priority for display

The below screenshot shows an extreme close-up of a gravel texture that is painted only weakly over a grass and leaves texture in the center left and a concrete texture in the center right. The images of the gravel pebbles are dominate the grass and leaves, but the grass and leaves texture is stronger in the gaps between gravel pebbles. Interestingly, even though the gravel is painted over the concrete, our eyes interpret it as a thin layer of concrete partially covering the low spots in the gravel.

Edge Blending

When a layer of paint has little falloff between regions of “paint” and “no paint”, then blending with the lower layers is based on the HM blending contrast property of the upper layer. The below screenshots show the same edge in which the concreate has an HM blending contrast of 0.6 on the left and 1.0 on the right. With a value of 1.0, the edge wiggles a bit to follow the implied high spots in each texture, but colors are never blended between the two textures.

These screenshots also illustrate that even though paint is applied in 0.5×0.5 meter squares, the display engine rounds off corners to avoid an unnatural-looking sharp bend. It also uses diagonal lines to connect paint squares that are painted in a stairstep fashion.

When the HM blending contrast property value is below about 0.55, lower texture layers start to bleed through even where the upper layer is at full strength. See the Material Properties section on page 130 for recommended values.

Bleed of 3-D Objects

A texture layer can have associated 3-D objects such as grass and pebbles. After painting or removing a texture layer, select Rebuild Terrain from the context menu to redraw these 3-D objects in the correct places.

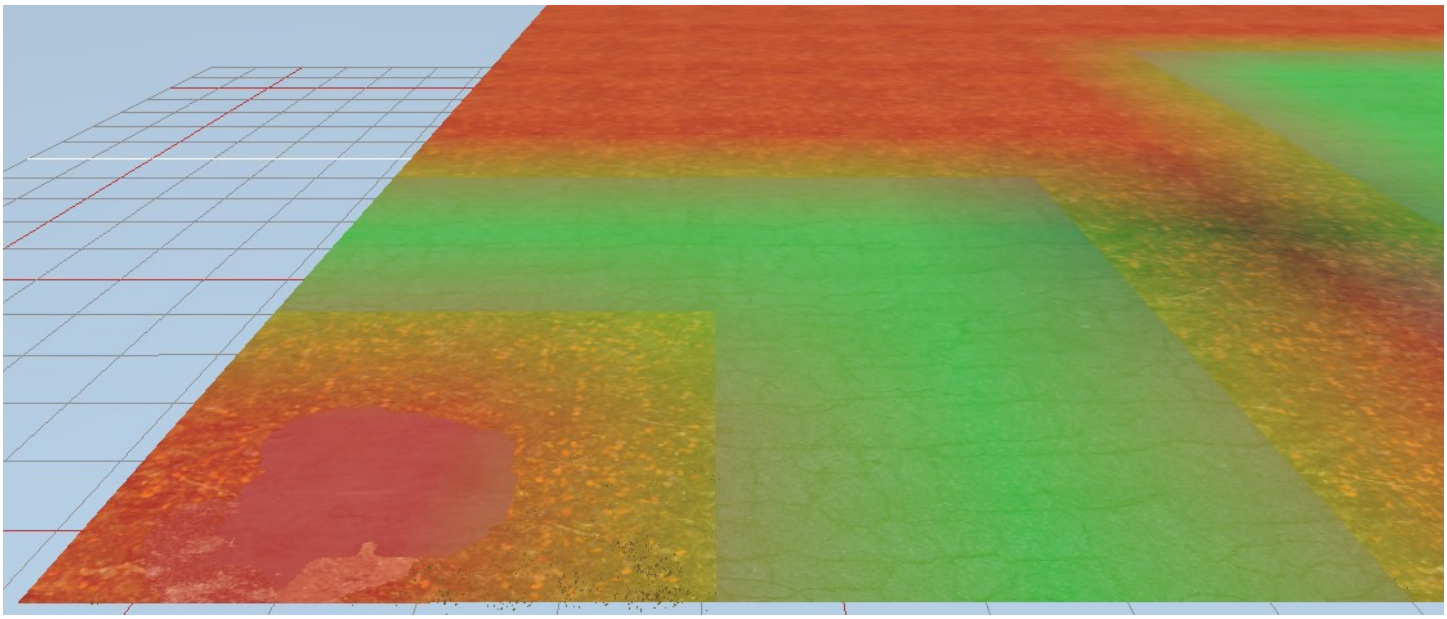
Where two texture layers are smoothly blended, 3-D objects from each texture can be intermixed depending on the strength of each paint.

Where two texture layers have a sharp line dividing them, a 3-D object from one may intrude on the other simply because of the size of the 3-D object. This is especially noticeable for grass when viewed closely. However, it is much less noticeable with the typically high camera position while driving in the game.

Additional Materials

To add additional materials, select Add Pbr Material from the context menu. Each material can have its own combination of texture layers.

Each terrain block can use only a single material. The material used in each terrain block is chosen by painting the material onto that terrain block. To begin painting, click the desired material under PbrMaterials, then select Opacity in the mode selection dropdown of the Brush dialog box. Because the default texture layer may be shared across different materials, the Editor helps distinguish which material is used where by tinting them in different colors.



Each map can use up to four different materials. While painting with Opacity, each material is tinted with a different color:

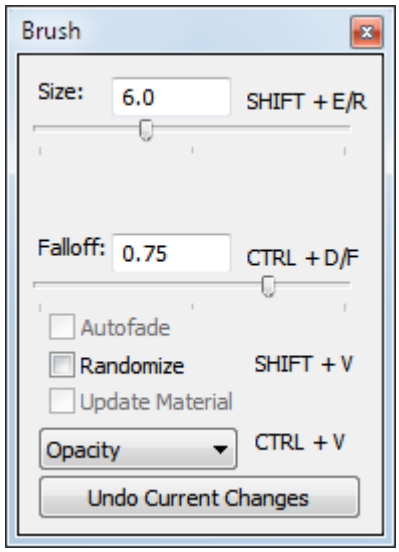
- The first material is tinted red.
- The second material is tinted green.
- The third material is tinted blue.
- The fourth material is tinted black.

Bug: Although each material has a distinct boundary at the edge of a terrain block, the tinting is smoothly blended between the centers of terrain blocks. This can make it harder to discern the true tint of small regions of material.

Although you can add more than four materials and attempt to paint with these additional materials, the changes will revert as soon as you Rebuild Terrain.

Bug: Although Saber documents that the Editor support up to four materials on a map, painting with the fourth material doesn't work correctly. If you Rebuild Terrain, these terrain blocks revert to the first material instead. This can be fixed up by hand with some tricky bitmap editing outside of the Editor. See the Material Layout Bitmap section on page 262 for details.

Painting with Opacity is all or nothing, so there is no Value parameter. The Size allows you to potentially paint many terrain blocks at once. Oddly, the Falloff paramter can still be adjusted, but it does nothing.



To paint the selected material onto the terrain with Opacity, right click on the desired block. Or right click and drag to quickly paint multiple blocks. There is no brush preview disc. As long as the Size is small, it's simply the terrain block under the mouse. If the Size is large, it is the many terrain blocks within that range of the mouse.

Material Opacity is somewhat different than other painting modes in that selecting a different material requires a new selection from the Scene View rather than a quick change to the Value or brush mode.

Bug: While painting with Opacity, right click updates both the artificial tinting and the textures on the terrain to match the new material. However, Undo Current Changes only reverts the changes to the tint. To correctly revert the textures on the terrain you must then select Rebuild Terrain from the context menu.

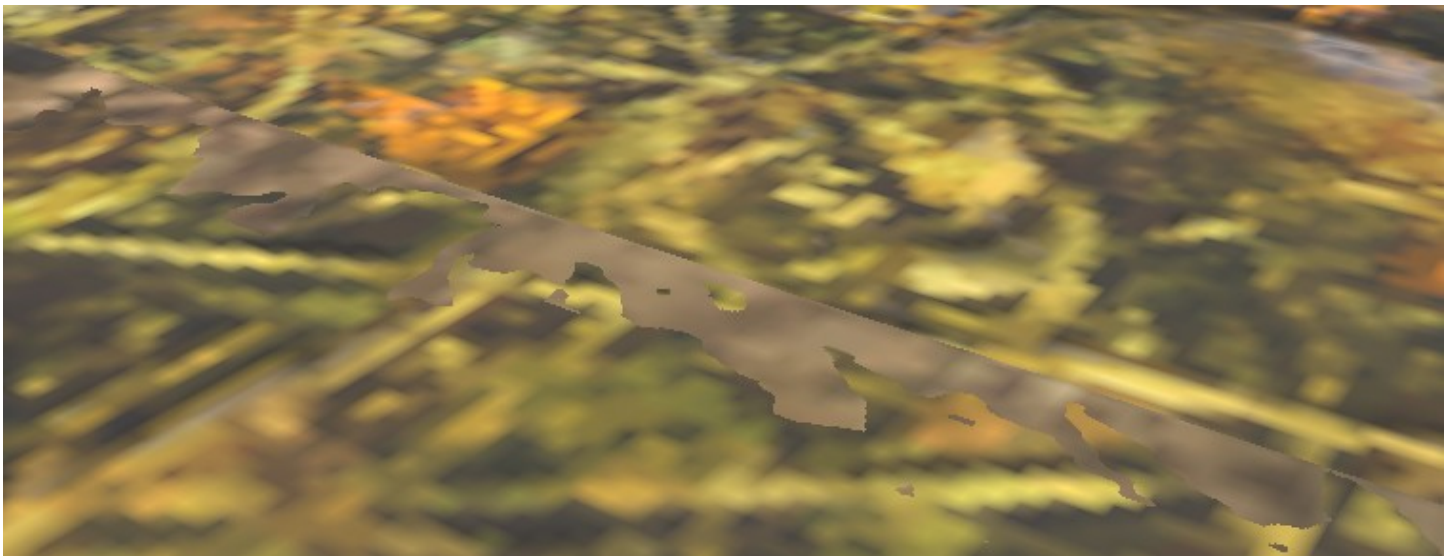
When you change the material assigned to a terrain block, the paint stencils for layers 1 – 4 are retained, although they may now point to different textures in the new material.

Texture Edges Between Materials

Different terrain blocks can use different materials and thus different combinations of texture layers. However, adjacent terrain blocks must have at least one matching texture which is used along the boundary between the terrain blocks. Otherwise, a sharp line results where the texture suddenly switches from one block to the other.

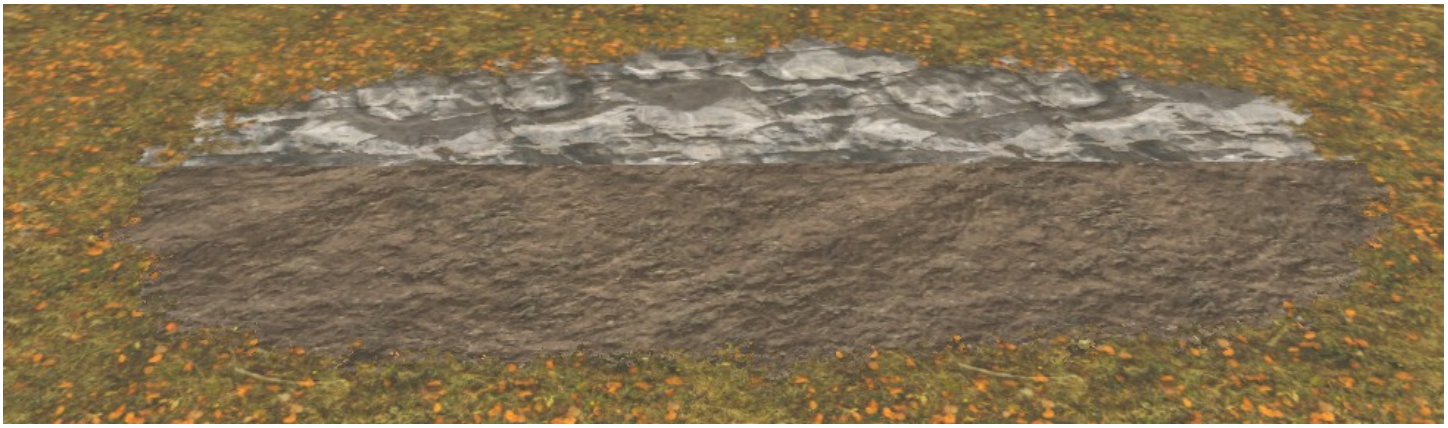
Textures can also be blended among layers at boundary between terrain blocks, but only if the materials in both blocks use the same textures in all blended layers. Needing to share multiple textures across materials sharply limits how many different textures can be used throughout the map, so typically you will share only one texture and avoid any blending of textures near material boundaries.

BTW, if two adjacent materials use the same texture, but in different layers, the display engine will blend between the layers separately on each side of the boundary. I.e. if a block uses grass in layer 0 adjacent to grass in layer 1 in an adjacent block, the first block displays a blend of layer 0 and layer 1 near the boundary. Since layer 1 is some other texture, this results in color bleed from an unwanted texture and a sharp line at the block boundary. Therefore, textures shared across materials should always be shared in the same layer.



If four materials are used in a map, and each material holds up to four texture layers, but at least one texture is shared across each pair of materials, then the maximum total number of textures per map is 4+3+3+3 = 13.

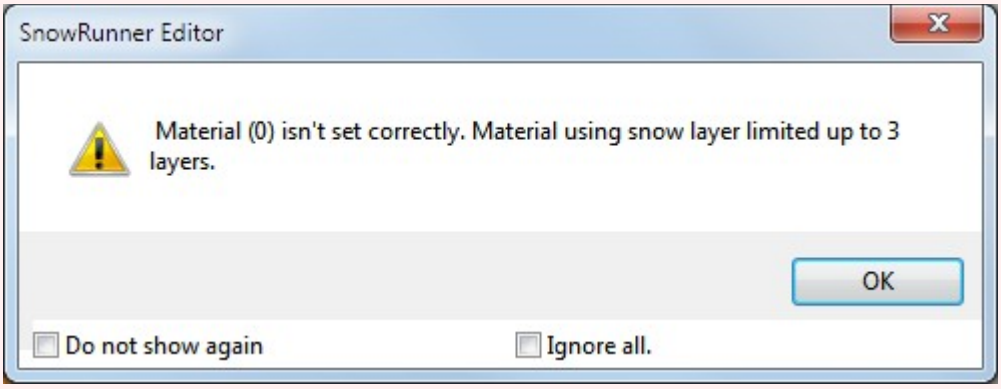
Caution: Although you select a particular material to begin painting a texture layer, the brush actually paints all materials with the selected texture layer number. So if you paint across the boundary of two materials, the painted texture takes on the charcteristics of whatever texture is assigned to that layer in each material.



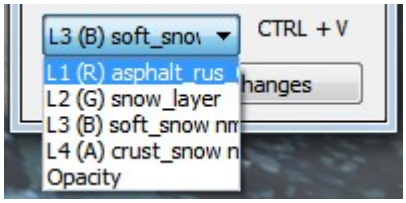
Snow Layers

The snow_layer texture is special and is officially supported only in Layer 3 of a material. When snow_layer is used in Layer 3, Layer 4 must not have any texture assigned to it.

Bug: There is no way in the Editor to remove a texture that has been assigned to a layer, so once Layer 4 has a texture assigned, snow_layer cannot be added to that material. Instead, you must create a new material and then delete the old one. See the Delete a Material section below for details and cautions regarding deleting material.



The snow_layer texture comes with two additional snow variations that can also be painted as layers. To get these additional layers, first assign the snow_layer texture to Layer 3, then **rebuild the terrain**. If you re-select the material, the brush now shows two additional paint layers: soft_snow and crust_snow.



Soft snow is snow in which patches are partially melted. It is mostly associated with plants since plants tend to retain heat longer than the air, and plants also provide a dappled mix of sun and shade.

Crusty snow is snow in which chunks of snow have partially melted and then refrozen, so it is a mix of snow and chunks of air-filled ice. It is mostly associated with snow that has been plowed or blown off the road.

The snow types differ only in how the display enginge renders the light bouncing off the snow. Unfortunately, the display engine used by the Editor doesn't model light bouncing at all, so the effect can only be seen in the game.

Bug: The Editor could assist in distinguishing the snow types, e.g. by applying a tint while a snowy texture is selected, but it does not. This means that there is absolutely no way in the Editor to tell where the different snow types have been painted. So paint with care, and try to do it only once, e.g. once everything else is nailed down.

The difference between types of snow is very subtle in the game. I arranged the below screenshot with the best lighting and camera angle that I could find, using sharp edges between the snow regions to make them more apparent. There is a slanted vertical stripe of soft_snow on the left and another slanted vertical stripe of crust_snow on the right. The normal snow_layer is in the center and at the edges.

Paint with snow_layer to put the snow texture on the ground. They optionally paint with soft_snow or crust_snow to change its appearance.

The snow_layer texture also supports deep snow to be piled underneath it. See the Snow Depth section on page 144 for details.

Delete a Material

You can delete a material using the context menu. When you delete a layer, the Editor makes the following changes:

- Any terrain blocks that use that material revert to using the first material instead. (This is the new first material if the previous first material was deleted.)

- The Mask channel index of all following layers are updated.
- The hidden bitmap that assigns materials to terrain blocks is updated with the new channel index values.

Bug: The Editor may crash when you delete a material. Save your work before deleting a material.

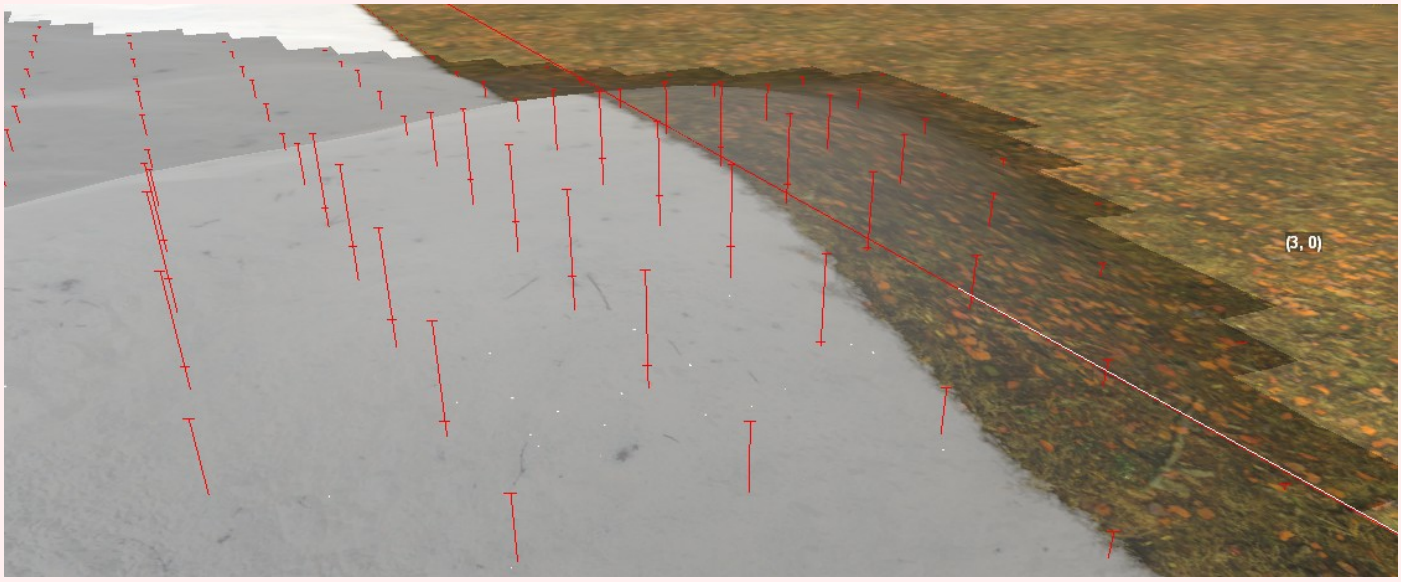
Bug: The Editor may render the terrain oddly after you delete a material. Rebuild the terrain to fix it.

Bug: The Editor is guaranteed to crash if you rebuild the terrain when there are no materials. If the map has only one material, and you want to delete it, I recommend that you first add a new material, then delete the old one.

Snow Depth

Select (Geometry)→(Snow) to paint deep snow on the terrain. By default, deep snow can only be painted where the snow_layer texture is present. The snow depth is ignored everywhere else.

Bug: Snow depth is ignored when it is painted elsewhere on a material that uses the snow_layer. However, if it is painted on a material that doesn't use snow_layer, it artificially raises the height of the terrain texture (as if there is snow there), but doesn't create any visual or physics effect associated with snow.



Paint snow depth by selecting Depth as the brush mode. This could alternatively be called snow height since the snow is piled on top of the base terrain height.

The Value range is -4.0 to +4.0. As with terrain height, whatever Value you set determines how much the snow depth changes with a single right-click and drag, regardless of how many times you drag over the same area with the mouse. You have to release the mouse button and make another pass to change the height further. As far as I can tell, a Value of 4.0 corresponds to about 1 meter.

Bug: Unlike anywhere else, 1.0 in the Value here corresponds to 20 centimeters, so +4.0 corresponds to +80 centimeters of snow depth.

While painting, the terrain is dynamically updated, but the Editor also draws thin red lines to help illustrate the depth of the snow. Tick marks are spaced every 0.5 meters along each line, starting from the top.

Although snow can be painted as deep as you want, a truck can never sink more than about 40 cm into it. To allow a truck to sink deeper, you must paint mud under the snow. If the snow is deeper than the mud plus about 40 cm, then only snow will be churned up as a truck sinks deep. But if the mud is deep enough to allow the truck to reach the base terrain, then it will start to churn up mud and soil as well.

Update Texture Layer

You can paint snow depth and the snow_layer texture layer simultaneourly onto the terrain. To do this, enable the Update Material checkbox in the (Snow) brush dialog. Any snow depth you add then automatically paints snow_layer onto the terrain as well. Even the smallest amount of snow paints the snow_layer with an effective Falloff of zero, so it may need some touch-up afterward.

If the terrain material does not include the snow_layer texture in Layer 2, Update Material has no effect.

Bug: If you click Undo Current Changes in the brush dialog, it undoes the snow depth changes, but it doesn't undo the changes to the snow_layer material.

UI Text

I use the term “UI text” to refer to any text that the game shows to the player, such as a zone name or objective information.

UI text can include almost any characters, including Unicode characters. (For the truly esoteric characters, you’ll need to check whether they are supported by the game’s font.)

Special whitespace characters such as a tab or linefeed won’t display properly, but normal spaces are fine.

Although the double quote " and backslash \ have special meaning in the objective_settings.json file, you don’t need to worry about them when using the Zone Settings Editor. The Zone Settings Editor escapes it appropriately.

SnowRunner uses the square brackets [and] and the vertical bar | for its own purposes, so they should not be used in UI text. Using them risks screwing up the message displayed to the player.

SnowRunner also allows certain HTML-style tags to format some types of UI text. This book explicitly states which types of UI text support formatting tags. If nothing is stated, then tags are implicitly not supported. (E.g. the objective name above does not support tags.)

The portion of text to format is surrounded by an opening tag and a closing tag. E.g. if a goal description is Deliver to the <y>Swamp</>:, then it is formatted as below:

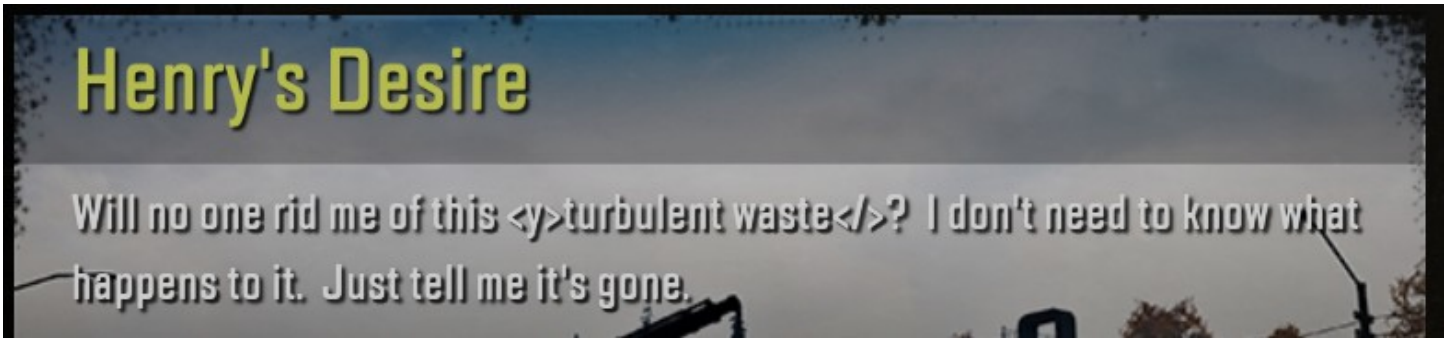
The formatting tags are as follows:

- <y> starts yellow text, and </> ends it.
- <r> starts red text, and </> ends it.
- <b> starts blue text, and </> ends it.
- <g> starts gray text, and </> ends it.
- starts italic text, and ends it.
- <u> starts underlined text, and </u> ends it.

Tags can be nested, e.g. to get yellow italic underlined text. In that case, tags must be closed in the reverse order of how they were opened. E.g. <y><u>SHOUTING</u></>. Unrecognized tags are quietly suppressed. I don’t know of a way to escape < so that it appears in the text.

Bug: If the UI text ends after a < without a following >, the game will crash when it tries to display the text. This may happen even when formatting tags are not supported in that text type.

This book explicitly states which types of UI text support these formatting tags. If nothing is stated, then tags are implicitly not supported. E.g. in the objective description, tags have no effect on the offer text, but they do properly format the text in the objective details, **but** the raw tags are displayed in the completed task in the player's profile. So I consider tags to be unsupported in the objective description.



Saber has not yet documented the formatting tags, and the only tag that Saber uses in the main campaign is <y>. I would guess that all formatting tags are safe to use where supported, but you should pick your own risk tolerance.

Localization

Any UI text can be localized into different languages. By default, a property value simply specifies the text to display. However, the property value can instead be an ID that is cross-referenced into a file that is different for each supported language. For example, the property value FREDS_GARAGE can display as "Fred's Garage" for someone playing in English or "Гараж Фреда" for someone playing in Russian.

To support localization, a file for each desired language must be placed in the Media/prebuild/<level_name>/texts directory. If you don't have this directory yet, you'll have to create it. The filenames for each language are as follows:

```
strings_brazilian_portuguese.str
strings_chinese_simplified.str
strings_chinese_traditional.str
strings_czech.str
strings_english.str
strings_french.str
strings_german.str
strings_italian.str
strings_japanese.str
strings_korean.str
strings_polish.str
strings_russian.str
strings_spanish.str
```

These files must use the little-endian UTF-16 encoding. The easiest way to ensure this encoding is to copy and modify an existing file. Windows Notepad or Emacs can read little-endian UTF-16 and will automatically

write the file back out in the same format. To get an example file to start with, choose File→Create default truck mod in the Editor, then copy one of the files from Media/Mods/<truck_name>/texts.

Each line of a localization file is an ID followed by the localized text enclosed in double-quotes, e.g.

```
FREDS_GARAGE                                "Fred's Garage"
```

For safety, I recommend that the localization ID be limited to a combination of lowercase a – z, uppercase A – Z, 0 – 9, and _. It should not include other characters, including spaces. The localization ID appears to be case insensitive, but I recommend using the same case everywhere.

The localized text can be almost any Unicode text, but with the following restrictions. If you want to use a standard a double-quote " in the quoted localized text, escape it with a backslash, i.e. \". A backslash should also be escaped with another backslash, i.e. \\.

As with non-localized text, special whitespace characters such as a tab or linefeed won't display properly, but normal spaces are fine. SnowRunner uses the square brackets [and] and the vertical bar | for its own purposes, so they should not be used here, even with a backslash escape. If this book says that formatting tags are supported in the text, then formatting tags are equally supported in the localization. The left angle bracket < should only be used as part of a supported formatting tag.

Bug: If the UI text ends after a < without a following >, the game will crash when it tries to display the text. This may happen even when formatting tags are not supported in that text type.

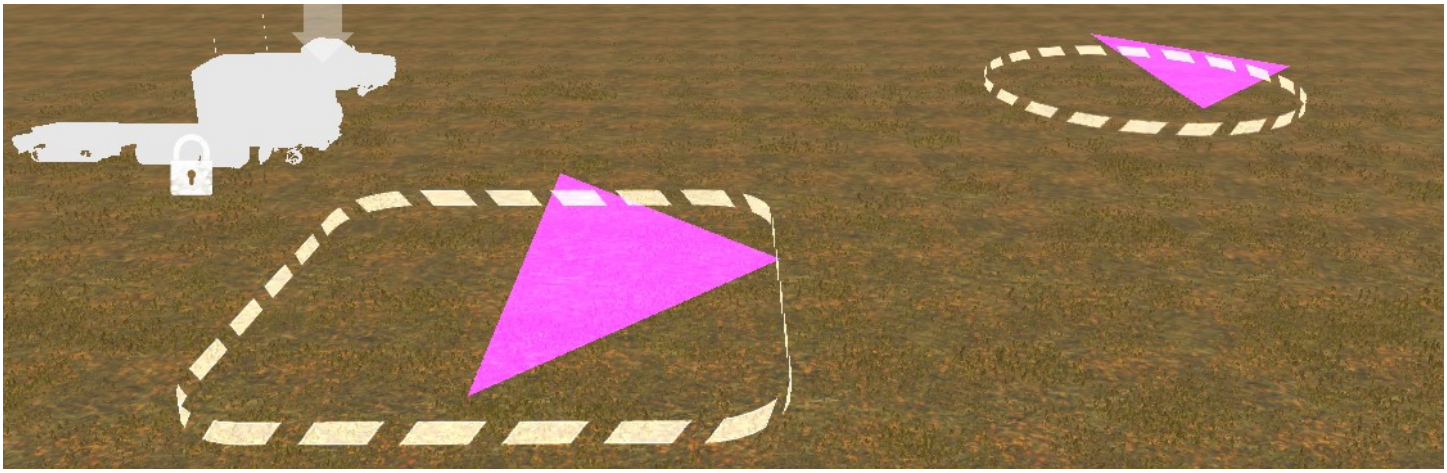
The separator between the ID and the localized text can be any mix of spaces and tabs, although Saber recommends tabs only.

Note that if an ID isn't found in the appropriate localization file, the ID itself is displayed. Thus, you can always simply specify the desired text directly in the property value rather than using a localized ID.

Bug: If SnowRunner can't find the desired localization for a particular ID, it displays the ID itself even if a localization exists in English or another language. Thus, if you've localized your map into only a subset of languages, the best option for the other languages may be to simply copy over the English file.

Zones

Zones define locations on the map that are used for various purposes such as truck services or objective destinations. Each zone is surrounded with a dotted line in the Editor and the game which I call the “ribbon”. In the game, the ribbon is animated to slide around the perimeter of the zone. In the Editor, the dotted line is static. The Editor draws an additional pink triangle to point in the direction of the zone’s orientation. The orientation of the zone makes a difference for certain zone types.



Zones are commonly associated with 3-D models such as garages, fuel pumps, etc., but the model is just set dressing. Only the zone has the associated game logic.

The main SnowRunner Editor can only specify the properties of the zone locator, i.e. its ID, location, size, and (potential) appearance. These properties are described below.

The **functions** of a zone are edited by the Zone Settings Editor (page 157). Permanent zone functions are specified by the zone type (page 169). Zone functions that apply only for the duration of an objective are specfied by the objectives (page 180) and the objective stages (page 194).

ID

Each zone must have a unique Id property value that links it to its separate type-specific properties in the Zone Settings Editor. I have found that the Id can be almost any text that doesn’t include a ", but Saber strongly recommends sticking to a combination of lowercase a – z, 0 – 9, and _, with no spaces.

Name and Icons

The Name property specifies the name that is displayed for the zone on the navigation map. Unlike the Id property above, the Name does not need to be unique. If no Name is specified, the Id is used instead.

The Name property isn't displayed while the player is driving. E.g. if the player unlocks a garage named "Fred's Garage", the game only displays a generic "Garage Unlocked" message.

The Name property is UI text and can be localized. UI text is described on page 146. Formatting tags are not supported.

The Icon 40x40 property specifies the name of an icon that is displayed above the zone when the player drives close to it. The same icon is also displayed on the navigation map.

The Icon 30x30 property specifies the name of an icon that is displayed in the list of locations on the left side of the navigation map.

Unfortunately, there is not a convenient menu of icon images to choose from the Editor. Refer to the Zone Icons section on page 242 for a table of icons and their corresponding names. Be sure to copy each name exactly as it appears in the table, even if it looks like the name has a typo. If an icon name is specified incorrectly, the game will display a small red and yellow placeholder image ( ) instead.

If a 40×40 icon is used in place of a 30×30 icon or vice versa, it will look generally OK, but it won't be properly centered.

If an icon property value is left blank, then no icon is drawn. However, other cues are present as usual to indicate the presence of the zone. The screenshot below shows two fuel stations on the map, one with associated icons and one without.

The Is Visible On Minimap property defaults to True. If you change it to false, then the zone is never displayed on the navigation map or on the list to the left of the map. In that case, the Name property doesn't matter because it is never used. The zone boundary still appears during normal driving. However, if Is Visible On Minimap is False, the icon above the zone is also suppressed.

Bug: The SnowRunner Editor and its documentation consistently misuse the term "minimap" when they mean the full-size navigation map. SnowRunner does not have a true minimap.

SnowRunner automatically suppresses any display of the zone in most circumstances where doing so is appropriate (e.g. when a zone is only used for an inactive objective), so there is usually no reason to change Is Visible On Minimap from its default value. Any exceptions are noted where appropriate.

For more information on zone visibility, see page 218.

Zone Perimeter

The Shape Type property determines the shape of the zone boundary: either a rectangle (with rounded corners) or a circle.

When the Shape Type is Rectangle, the length of the sides parallel to the zone's direction are set by the Length or diameter property, measured in meters. The length of the perpendicular sides are set by the Width property.values (in meters). The radius of its rounded corners is determined by the Rounding radius value (in meters).

When the Shape Type is Circle, the diameter of the circle is set by the Length or diameter property value (in meters). The Width property is ignored.

Bug: When the Shape Type is Circle, the game ignores the Width value. However, the Editor still draws the zone's front-facing arrow using the Width as its width. For better visuals in the editor, you might want to set the Width equal to the diameter for a Circle zone.

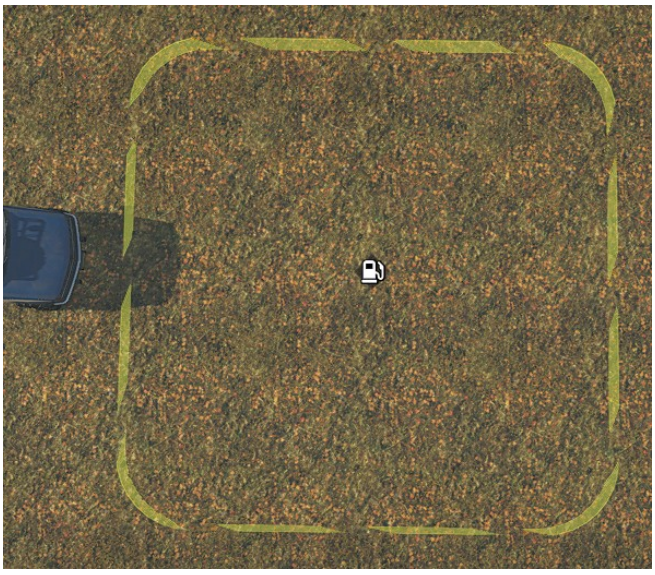
Ribbon Shape and Size

A ribbon of dotted lines is animated around the perimeter of the zone. By default, the Is wall property is False, and the ribbon lies flat against the ground. If the Is wall property is set to True, the ribbon is vertical.

The width of the ribbon is determined by the Thickness property, but it is not measured in meters. Instead, the ribbon width is the Thickness value × 0.5 meters.

Vertical Ribbon

When Is wall is True, the layout of the ribbon in the top-down view is determined by the Length or diameter, Width, and Rounding radius properties as described above. Independently, the layout of the ribbon in the side view is determined by the Thickness and Height properties.



The Rounding radius isn't constrained by the Thickness of the ribbon, so the minimum Rounding radius is 0, entirely removing the rounding from the corners. The maximum Rounding radius is half the minimum of Length and Width. If the Rounding radius is larger, the rendering engine reduces it as necessary.

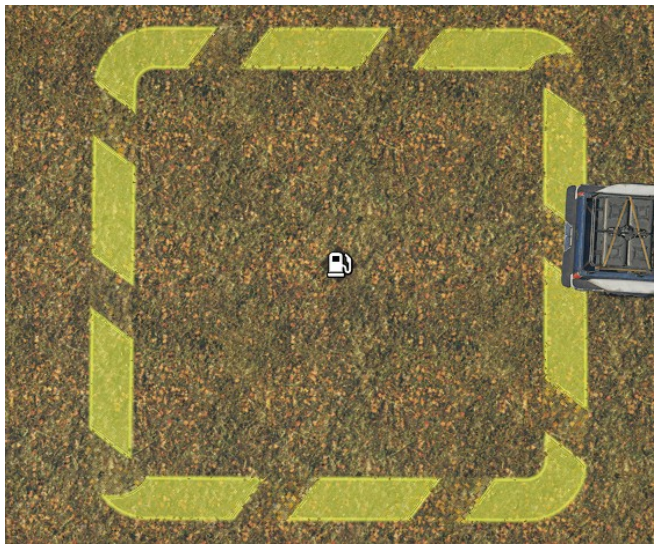
When the Height value is 0, the vertical ribbon is centered at ground level. Its height is adjusted by the Height value, measured in meters.

Bug: The Editor places the forward-facing pink arrow (visible only in the Editor) at **half** the height given by the Height value.

To raise the bottom of the ribbon to exactly ground level, the Height must be set to half the ribbon width. However, since Height is measured in meters but Thickness is measured in half meters, the Height value must be one-quarter of the Thickness value.

Flat Ribbon

When Is wall is True, the layout of the ribbon in the top-down view is determined by a combination of its Length or diameter, Width, Rounding radius, and Thickness properties.



Bug: Although the game draws the ribbon either flat or vertical, depending on the Is wall property, the Editor always draws the ribbon vertically. The Editor still ignores the Height property when placing the ribbon, however, instead drawing the bottom of the ribbon above the ground by half its width. On the other hand, the Editor uses the Height property to set the height of the forward-facing arrow.

This makes it much harder to experiment with different property values to achieve the desired look. So the calculations below that determine the exact ribbon placement may be especially useful to you if you don't mind the math.

When the ribbon is flat on the ground, it has additional invisible padding equal to half of the ribbon width on each side. Since the ribbon width itself is half the Thickness value in meters, the total padded ribbon width is

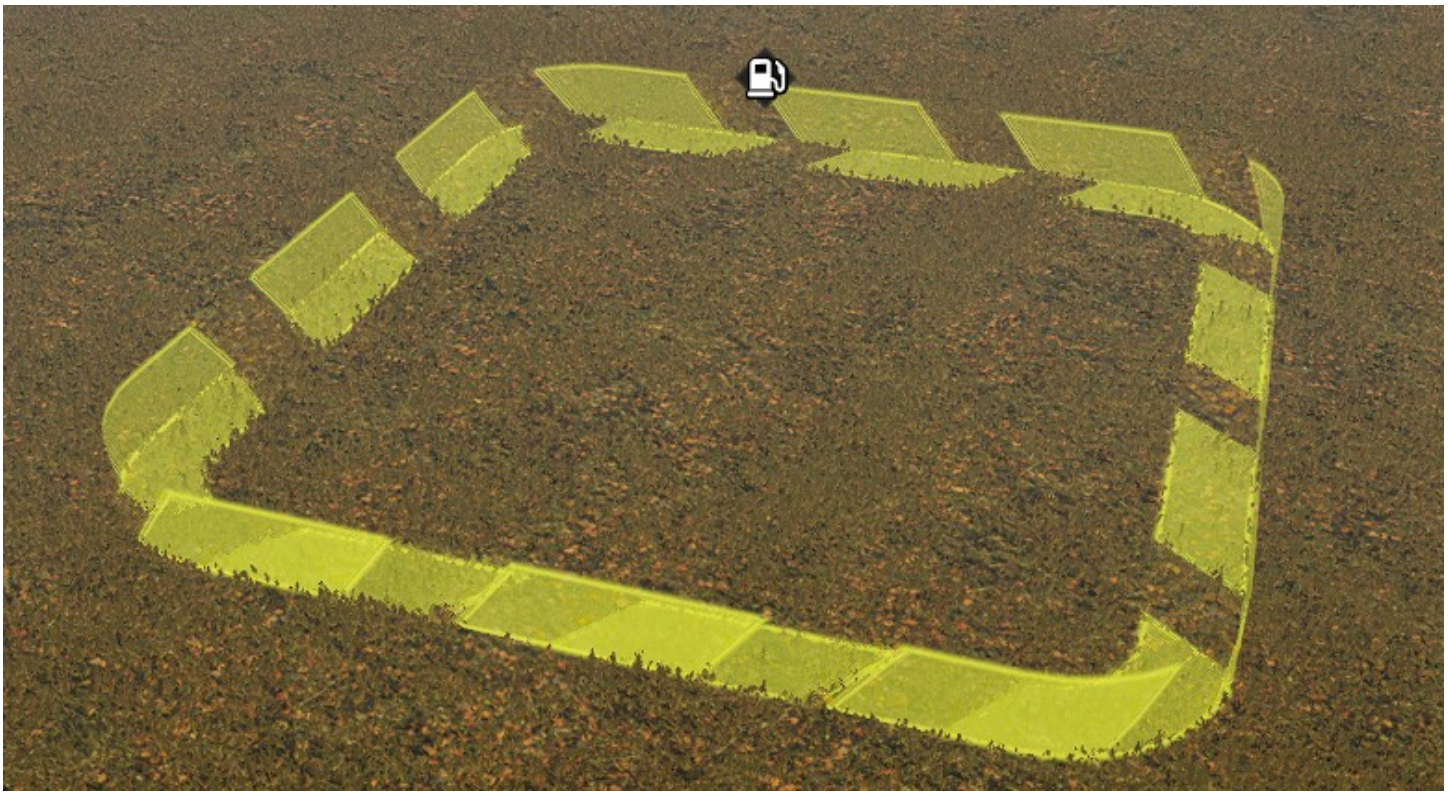
exactly the Thickness value in meters. This padded ribbon width appears to be what SnowRunner uses for its internal calculations, and it is important when calculating the precise placement of the ribbon.

The ribbon is placed so that the outer edge of its **padded** width aligns with the dimensions provided by Length or diameter and Width. This means that the ribbon itself is inset by half its width.

The Rounding radius sets the **outer** radius of the **padded** ribbon width, in meters. The inner edge of the ribbon necessarily has a smaller radius. To avoid a negative radius, the game automatically reduces the **padded** ribbon width to no greater than the Rounding radius. This means that the visible ribbon width is no more than half the Rounding radius, and the inner edge of the visible ribbon is always rounded, never a sharp corner.

The actual detection area for the zone is determined by the outer edge of the padded ribbon width, which is exactly set by the Length or diameter, Width, and Rounding radius properties. However, you might prefer to set the visible part of the ribbon to particular dimensions. If so, use the calculations below:

- Length = desired visible outer length + 0.5 × Thickness
- Width = desired visible outer width + 0.5 × Thickness
- Rounding radius = desired visible outer radius + 0.25 × Thickness



Other Ribbon Properties

The speed of ribbon animation is given by the Scroll speed and direction property, in clockwise meters per second. The animation is counter-clockwise if the value is negative. If Is wall is False, the speed is measured at the outter edge of the padded ribbon.

The brightness of the ribbon is determined by the Opacity property. Since the ribbon color is added to the ground color, it is never truly opaque, but larger values make it appear more opaque, and smaller values make it appear more translucent.

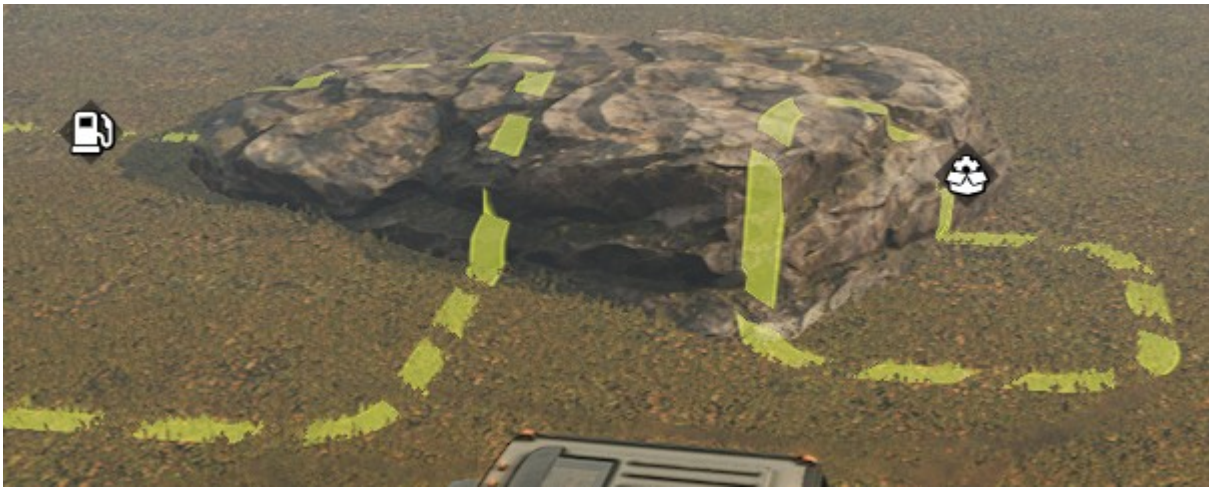
Bug: The Opacity property only affects how the ribbon looks in the Editor. In the game, the ribbon's color and brightness are determined by the zone type, and the Opacity property is ignored.

Hilly Zones

If the terrain in a zone is not flat, the zone boundary gets distorted. If Is wall is False, the ribbon conforms to the terrain surface. If Is wall is True, the ribbon follows a straight line from the height at one corner to the height at the next corner (modulo the Rounding radius). The below screenshot shows the difference between the two styles. They look goofy when seen juxtaposed like this, but each one individually is fine.

Zone Ribbon on Models

A flat zone perimeter ribbon not only conforms to the terrain, but also to most rock models. This is only visible in the game, not the Editor.



The zone ribbon also crawls over the legs of the log_scavange_02 model

Many bridges and platforms also have XML that says the zone ribbon should be placed on the model, but it only works if the model surface is placed at exactly ground level. It’s unclear how this is useful for bridges, and it is not at all useful for platforms.

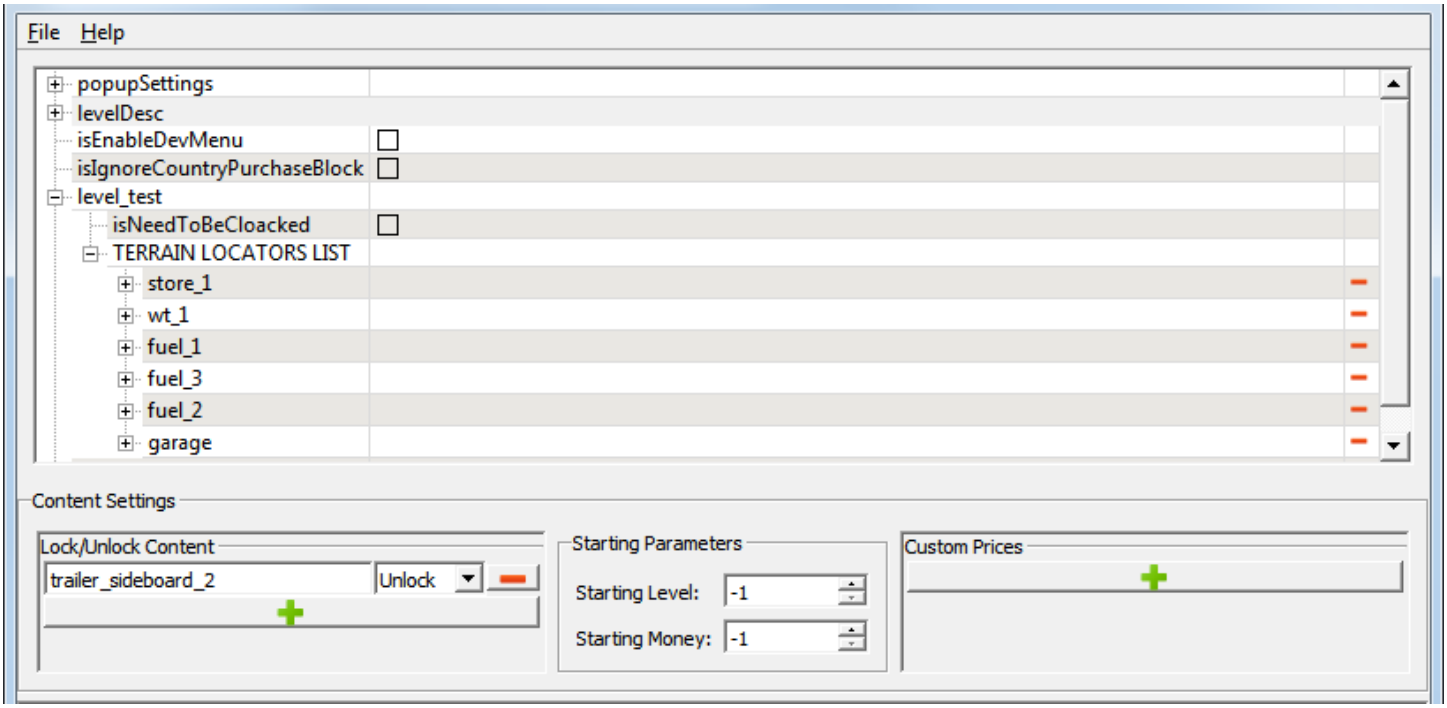
Other miscellaneous models also claim to support a zone ribbon. I have not tested that assertion.

Zone Settings Editor

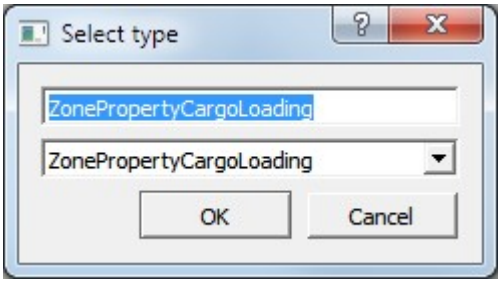
The Zone Settings Editor allows you to set the type of each zone and the various properties that are specific to the zone type. For unknown reasons, it also includes a random assortment of other non-zone-related properties.

To start the Zone Settings Editor from the main SnowRunner Editor, click the Zone settings button on the toolbar (the second button from the right). **Caution:** while the Zone Settings Editor is open, the main SnowRunner Editor window is completely unresponsive. Close the Zone Settings Editor to return to the main SnowRunner Editor.

This section describes only the zone properties. In the hierarchical list, open the container for your level name, and then the subcontainer TERRAIN LOCATORS LIST. This list is automatically populated with the IDs for each zone that you set in the main Editor. If a zone does not have an ID, it is not listed.



To add properties to a zone, open its container, then click the green plus to the right of props. Then select the desired zone type from the dropdown menu in the dialog box and click OK.



Confusingly, the dialog box has two editable fields, and selecting an item from the dropdown menu updates both of them. Details TBD.

The zone type properties are weirdly named. The Zone Types section on page 169 describes the various zone types and the corresponding property. Depending on the zone type, the Zone Settings Editor may make a number of subproperties available.

Manage the Zone ID

The loose connection between the zone Id in the main SnowRunner Editor and the zone properties in the Zone Settings Editor is a conceptual minefield. Although the two Editors attempt to keep your changes in sync, it is easy to lose confidence in what they are doing.

Add or Delete a Zone

The main Editor saves most of its property data in Media/prebuild/<level_name>.xml, while the Zone Settings Editor saves the zone properties in Media/levels/<level_name>/zone_settings.json.

If you add or delete a zone in the main Editor, it doesn't touch the zone_settings.json file directly. Instead, when you start the Zone Settings Editor, the main Editor tells it what zones currently exist. Thus, the Zone Settings Editor allows you to edit the same zones that are present in the main Editor.

Note that if you don't select Save in the Zone Settings Editor, no changes are made to the zone_settings.json file.

Tip: If you delete a zone in the main Editor the zone and its properties won't be shown in the Zone Settings Editor. However, as long as you select Save in the Zone Settings Editor, the zone settings can be recovered. Close the Zone Settings Editor (without saving), re-create the zone in the main Editor, restart the Zone Settings Editor, and the zone's properties should be there once more.

Rename a Zone ID in the Main Editor

If you change the Id of a zone in the main Editor, the Editor keeps track of this rename. The next time you start the Zone Settings Editor, the main Editor tells it to change the name(s) in the zone_settings.json file before it opens the window where you can edit the zone settings.

This is weird, so let me say it again. If you renamed a zone Id in the main Editor, the zone_settings.json file can change even though you never selected Save in the Zone Settings Editor.

This oddity only applies to renamed zone IDs. If there are also new or deleted zones, those still aren't updated in zone_settings.json until you select Save in the Zone Settings Editor.

Note that if you then close your map in the main Editor without saving, your zone IDs will be inconsistent between the <level_name>.xml file and the zone_settings.json file.

This inconsistency between files doesn't happen in the other direction, however. If you save your work in the main Editor while it is tracking a renamed zone ID, the main Editor quietly starts the Zone Settings Editor in the background and tells it to change the name in the zone_settings.json file so that both files are in sync.

Rename a Zone ID in the Zone Settings Editor

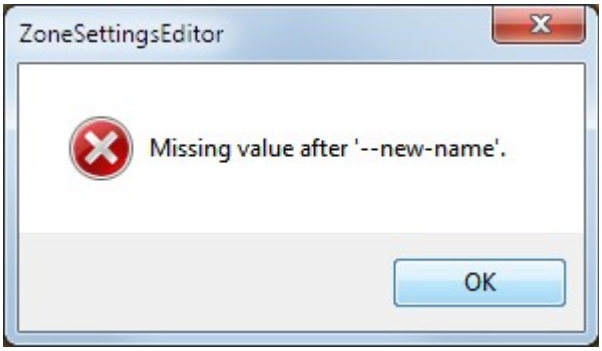
If you change the ID of a zone in the Zone Settings Editor, it does not attempt to make a corresponding change in the main Editor. If you save your change in the Zone Settings Editor, the two files will be out of sync.

After closing the Zone Settings Editor, you can immediately make the same change in the main Editor. This will attempt to make a corresponding rename in the zone_settings.json file, but since the old ID no longer exists in that file, it is harmless.

After you manually change and save both files in this order, the two are correctly in sync. However, it's easier to rename the zone ID in the main Editor first and let it deal with synchronizing the change.

Clear the Zone ID

If you clear the Id of an existing zone, the main Editor tracks this as a rename event. The next time you save your changes in the main Editor or start the Zone Settings Editor, you'll get an error that a name is missing.



This is harmless. All of the other zones are presented correctly, including any that have been renamed to valid IDs.

Uncommitted Id Edits

When you type a new name into a zone's Id property in the main editor, the change is not committed right away. This uncommitted status can be seen in the Scene View, which doesn't update the zone name even when you press return. To commit your edit, left click in the main panel or click something else in the Scene View.

If you create a new zone, assign it an Id, and then start the Zone Settings Editor while the Id change is uncommitted, then the Zone Settings Editor won't know the new zone's ID, and it won't include it in the list of zones.

Bug: After you close the Zone Settings Editor, the main Editor will no longer have the zone selected, but it still won't have committed the Id change. Select the zone again, then click elsewhere to deselect it, and it should finally be committed. You can now restart the Zone Settings Editor to see and edit the new zone.

Things are worse if you have an existing zone in the zone_settings.json file, and you rename its Id in the main Editor without committing the change. If you then start the Zone Settings Editor, the main Editor tells it to rename the zone according to the uncommitted change, but the main Editor **also** says that the current list of zones includes the zone with its old ID.

Bug: Oh yeah, that's a bug.

As an example, assume you have an uncommitted ID change from x to y, and you start the Zone Settings Editor. the ID is changed in zone_settings.json file from x to y. However, the Zone Settings Editor is also told that the list of zones includes x and not y. Since the Zone Settings Editor no longer has any properties for zone ID x, it lists x without any properties. If you select Save now, you'll lose your properties for the zone. If you realize the situation, close the Zone Settings Editor without saving. Select the zone again the main Editor, then click elsewhere to commit the change, then restart the Zone Settings Editor.

Tip: Make a habit of clicking elsewhere to commit a change to a zone's ID before doing anything else.

Duplicate IDs

Now that you have some understanding of how ID changes are synchronized, it is apparent how confusing it is to ever have two zones with the same ID.

Bug: If you change the Id of a zone multiple times in the main Editor, the Editor tracks this as a series of renames, not as a single change from the original ID to the newest ID. If any of the intermediate IDs were a duplicate of another zone's ID, this will cause problems even though the Zone Settings Editor was never invoked while duplicate names were present.

Of course, if you choose Duplicate on a zone, then the new zone has the same ID. You could avoid using Duplicate, but it can be convenient for duplicating zone locator properties or aligning zones near each other.

If you use Duplicate to make one zone into multiple, whichever one you rename first ends up with the original zone's properties. Although the main Editor presents the other renames to the Zone Settings Editor, there is no longer a zone with that ID to rename.

Tip: You can actually take advantage of the above bug. After duplicating a zone, rename the original zone, then rename the duplicates, then rename the original zone back to it's original name. When you start the Zone Settings Editor, the original zone will have its original name **and** its original properties.

Zones Used by Objectives

Some properties under the objectiveSettings category in the Zone Settings Editor take a zone Id as a value. However, the Editors make no attempt to keep these IDs in sync as you make changes. If you change the zone ID in one place, it's up to you to also change it in the other place.

Zone Settings Editor Controls

The hierarchical list in the Zone Settings Editor is similar to the ones in the main Editor, but some of the controls are different. The complete list of confusing, context-sensitive Zone Editor Controls is below. Controls that are useful in all contexts are **highlighted**.

Some mouse controls can be performed with a either left or right click:

- **Click on a + to the left of a hierarchical container to expand that container, showing its contents.**
- **Click on a – to the left of a hierarchical container to contract that container, hiding its contents.**
- **Double click on an editable string to edit it.** Property values are generally editable, as are zone IDs and ZoneProperty names.
- Double click on a non-editable container to expand or contract a container.

Others can only be performed with a left click:

- **Left click on a green + to the right of a property to add a sub-property within it.**
- **Left click on a red – to the right of a property to delete it and all of its contents.**

Certain actions can also be performed using the keyboard:

- The up and down arrows move the selection up and down the list.
- The right arrow does different things in different contexts:
  - On an unexpanded folder or container, it expands the container.
  - On an expanded folder or container, it moves to the first item in the container.
  - On a non-container item, it does nothing.
- The left arrow does different things in different contexts:
  - On an expanded folder or container, it hides the items in the container.

  - On an unexpanded folder or container, it moves to the container of the current item (up one level in the hierarchy).
  - On a non-container item, it does nothing.

- F2 edits the selected item. This can only edit its name if the item was selected via the keyboard. It can edit the value if you selected the item by clicking in the value field.

- The Home key moves the selection to the top of the list.

- The End key moves the selection to the bottom of the list.

- The Page Down key moves the selection down by approximately one page.

- The Page Up key moves the selection up by approximately one page.

- **Ctrl-S saves changes.**

> **Bug:** A change isn't saved if it hasn't been committed. If the cursor is still visible in a text field, the edits to that field aren't yet committed. Commit your changes by pressing Enter or selecting another item.

- **Ctrl-Q exits the Zone Settings Editor.**

> **Bug:** If the only change hasn't yet been committed, the Zone Settings Editor won't prompt you to save before quitting.

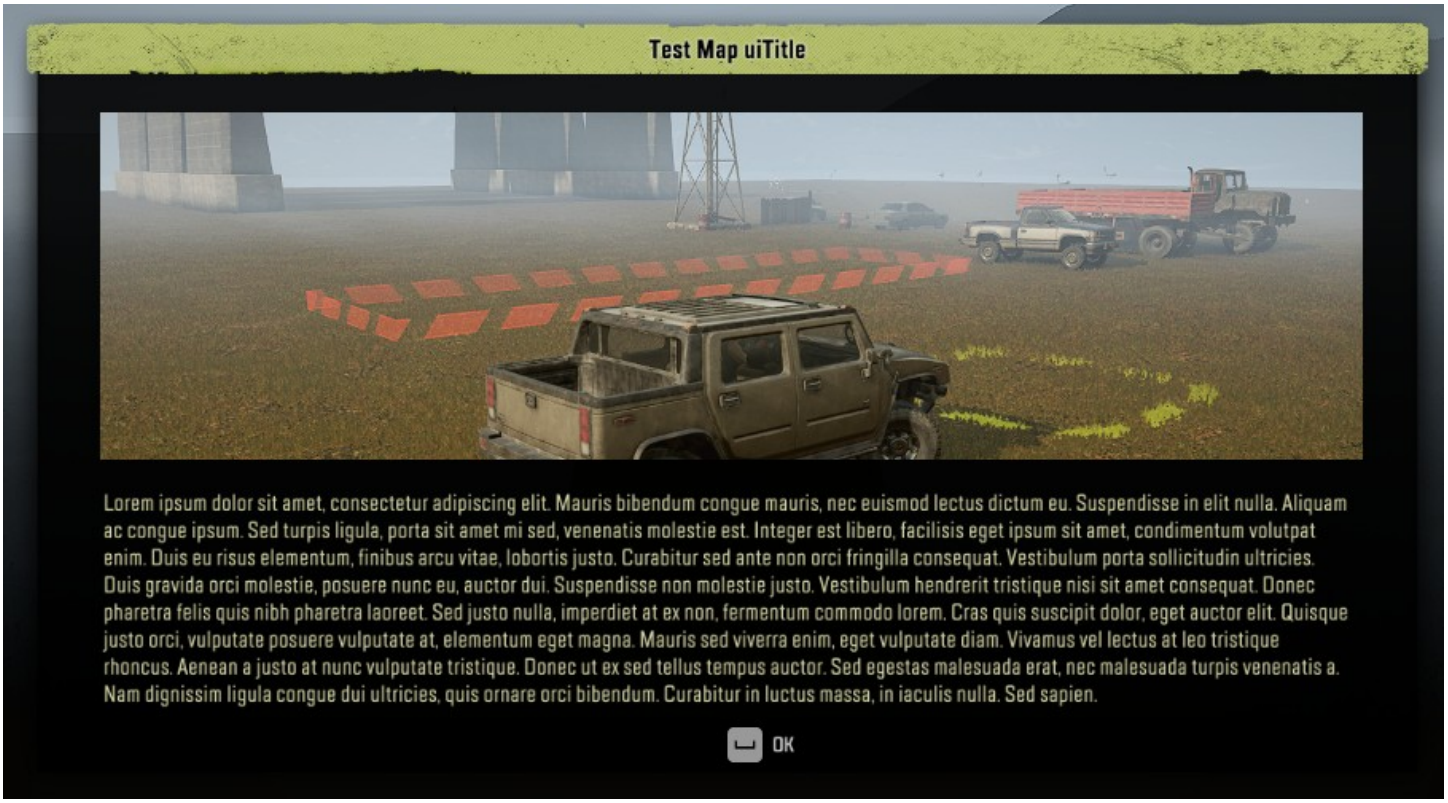
In all cases, the panel automatically scrolls to keep the selected item in view.

Map Initialization

The Zone Settings Editor includes a number of miscellaneous properties that apply to the map as a whole. These properties generally describe how the map should be initialized and how it fits into its region, but there isn't really a coherent difference between these properties and the map properties that are specified within the main Editor (page TBD).

Map Introduction

When the player enters a fresh map, an optional window displays to introduce the map. The appearance and behavior of this window are described by the properties below.



uiTitle

> popupSettings.uiTitle: UI text (page 146)
>
> Specifies the name of the map to use at the top of the introduction window.
>
> Default: blank; the window does not appear when the uiTitle is not specified, and the remaining popupSettings properties are ignored.

uiText

> popupSettings.uiText: UI text; supports formatting tags, but not <i> (page 146)
>
> Specifies a description of the map to use at the bottom of the introduction window.
>
> Default: blank; no text is displayed.

The space allocated to uiText is a fixed size, regardless of how much text there is. To fill the space, about 1200 characters is appropriate. The example screenshot above has exactly 1200 characters of uiText. If there is more uiText than will fit, the text is cut off.

uiIcon

> popupSettings.uiIcon: string
>
> Specifies the image to use in the center of the introduction window.
>
> Default: blank; no image is displayed.

The image should be 1188×326 and should be saved as a PNG file in prebuild/*map_name*/ui/textures/*image_name*.png

Set the uiIcon value to the *image_name* without the .png extension.

if the image is too big or too small, it will overfill or underfill the window and be incorrectly centered.

If the image is not found with the specified name, a filler image is displayed instead. (Weirdly, uiIcon is the opposite of other properties in using the filler image in place of a broken image, and not in place of an unspecified image.)

objectives

Any number of objectives can be activated when the map is entered, but **only** if uiTitle is specified as above. If uiTitle isn't specified, then the game doesn't pop up the introduction window, **and** it doesn't activate initial objectives.

> popupSettings.objectives+[*n*]: string
>
> Specifies an objective to activate.
>
> Default: blank; ignored.

The objectives to be activated can be tasks, contracts, or contests, although automatically activating a contest is kind of rude to the player.

The game also begins tracking the first activated objective, specified in objectives+[0].

Map Images

You can specify images to use to represent the map in the global map view and in a save slot.

Map Image in Global View

> levelDesc.globalMapPreview: string
>
> Specifies the image that represents the map in the global view.
>
> Default: blank; a default image is used.

The image should be 360×216 and should be saved as a PNG file in prebuild/*map_name*/ui/textures/*image_name*.png

Set the globalMapPreview value to the *image_name* without the .png extension.

If the image is too big, only the upper left portion of the image is shown. If the image is too small, then black borders are added to the right and below the image.

If the image is not found with the specified name, a black rectangle is displayed instead.

Bug: Although a global view is available without a regional map, the map image is ignored in this case. The map image can only be seen when playing a region that includes the map.

Map Image in Save Slot

> popupSettings.saveSlotMapPreview: string
>
> Specifies the image to use in a save slot when the active truck is on the map.
>
> Default: blank; a default image is used.

The image should be 360×203 and should be saved as a PNG file in prebuild/*map_name*/ui/textures/*image_name*.png

Set the saveSlotMapPreview value to the *image_name* without the .png extension.

if the image is too big or too small, it will overfill or underfill the save slot window and be incorrectly centered.

If the image is not found with the specified name, a default image is used instead.

Tip: Because the game fades the bottom of the image to black, a 360×216 image works just as well, which means that you can reuse the globalMapPreview image. However, you're taking a chance that the game may change and break your image.

Dev Tools Menu

The Dev Tools menu (page 222) is always shown for a map in development. You can choose whether the Dev Tools menu is shown or hidden when the map is published. Normally it would be hidden, but you can choose to show it if your map is intended as a "proving grounds" type map.

> isEnableDevMenu: checkbox
>
> If checked, the dev tools menu is shown when playing the published map.
>
> Default: unchecked; the dev tools menu is hidden in the published map.

Ignore Country in Garage Store

> isIgnoreCountryPurchaseBlock: checkbox
>
> If checked, the player can buy trucks from any country in the garage.
>
> Default: unchecked; the player can only buy trucks from a single country in the garage.

Bug: There is no way to specify what country a map is associated with. If isIgnoreCountryPurchaseBlock is unchecked, the player can only buy US trucks in the garage. This checkbox is more useful at the regional level.

Trailers are not associated with a particular country, so all trailers are available from the trailer store.

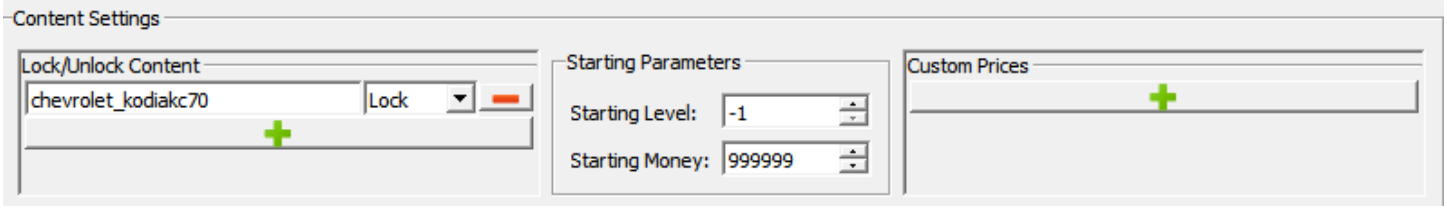
Map Cloaking

> *map_name*.isNeedToBeCloacked: checkbox
>
> If checked, the navigation map starts completely black (except for the regions revealed by the active truck).
>
> Default: unchecked; the navigation map starts dimly revealed in gray everywhere.

The process of revealing the map is described on page 225.

Content Settings

The Zone Settings Editor has a few extra widgets in the middle of the window for adjusting the availability of vehicles and add-ons.



Bug: The Zone Settings Editor doesn't keep track of when these widgets are modified, so it won't warn you if you close the window without saving changes. Make sure to save your changes.

Bug: The blank bottom section of the Zone Settings Editor doesn't appear to be used for anything.

Lock/Unlock Content

By default, trucks, trailers, and add-ons are unlocked when the player reaches a certain rank, after which they can be purchased in the garage (trucks and add-ons) or the trailer store (trailers). If the player discovers a locked vehicle on the map, she can then purchase in the garage or trailer store even if its rank requirement hasn't been met. Similarly, if an add-on is awarded to the player, she can purchase more of that add-on without needing to meet the rank requirement.

The whole rank system can be disabled, however, effectively giving a vehicle or add-on an infinite rank requirement. Do this by clicking the green + below Lock/Unlock Content and typing the truck type or add-on name into the blank field. When content is locked in this way, the player can only unlock it by discovering a locked vehicle or by being awarded an add-on. If the player doesn't do this, or if the vehicle or add-on isn't made available for discovery or reward on the map, then it remains locked forever.

You can change the Lock field to Unlock. This reverts the item to its usual rank requirement. This is equivalent to removing the item from the list, but is easier to reverse (e.g. for testing purposes).

There is no way to lock all content by default. Even if you listed every item in the game, the player may install a new DLC or mod, which is then unlocked by default. So you have to rely on the rank system and the economy and the player's own sense of fair play to keep overpowered content out of her hands.

Tip: The real purpose of locked content seems to be to avoid a situation where a discovery or reward is anti-climactic because the player has already purchased the item. So considering locking every vehicle and add-on that the player can discover or be awarded.

Starting Parameters

You can specify the player's starting rank (Starting Level) and savings (Starting Money). The initial values of -1 set the player's rank and money to the default values of 1 and 5000, respectively.

Custom Prices

The system for custom prices works similarly to the system for locked content (above), but instead of locking a vehicle or add-on, it sets its price. The minimum price is 0, and it applies to both buying and selling. (In hard mode, selling is a fraction of the buy price.)

Note that for trucks, the custom price is the **base** price. The garage store price includes the prices of all default add-ons, which can significantly increase the price above the base price.

Zone Types

Objective-related zones are not assigned a zone type. They are specified via a separate system, described beginning on page 180.

Garage Entrance

A zone with the property ZonePropertyGarageEntrance allows the player to enter a garage. The forward orientation of the zone only matters if the zone is also used as a garage exit.

Each map may contain only one garage, and Saber says that a map “can” contain only one garage entrance. However, my testing shows that a map can contain multiple garage entrances, although they all enter the same garage. Taking advantage of this may be risky, however. Also, while the player is in the garage, the building list next to the navigation map shows the player icon in both of the garage entrances.

If a separate garage exit is not specified, then the garage entrance also acts as the garage exit. In this case, the zone’s orientation is important, as described in the following section.

Garage Exit

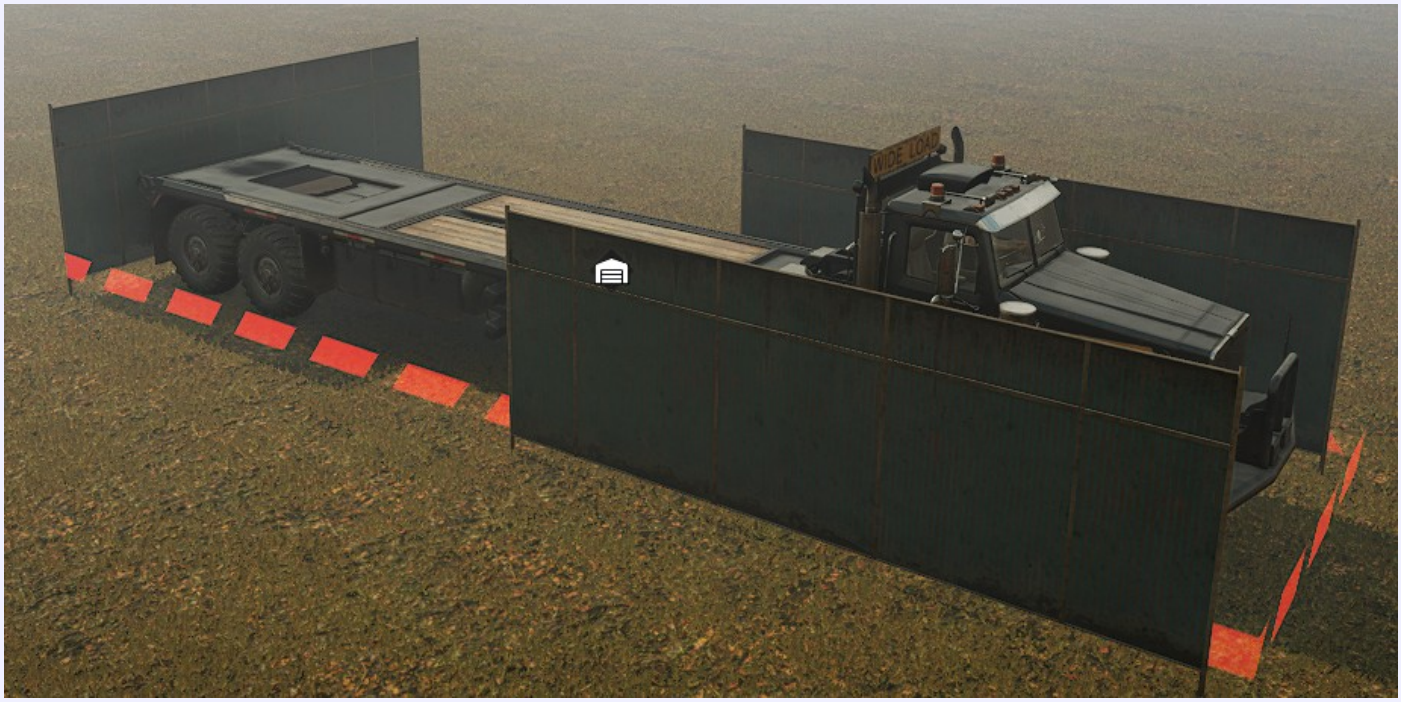
When the player exits the garage, the truck is placed in a zone with the property ZonePropertyGarageExit. If there is no such zone, then the truck is placed in a zone with the property ZonePropertyGarageEntrance instead.

A ZonePropertyGarageExit zone has a red boundary. It is automatically suppressed from appearing on the navigation map or building list, regardless of the Is Visible On MiniMap property value.

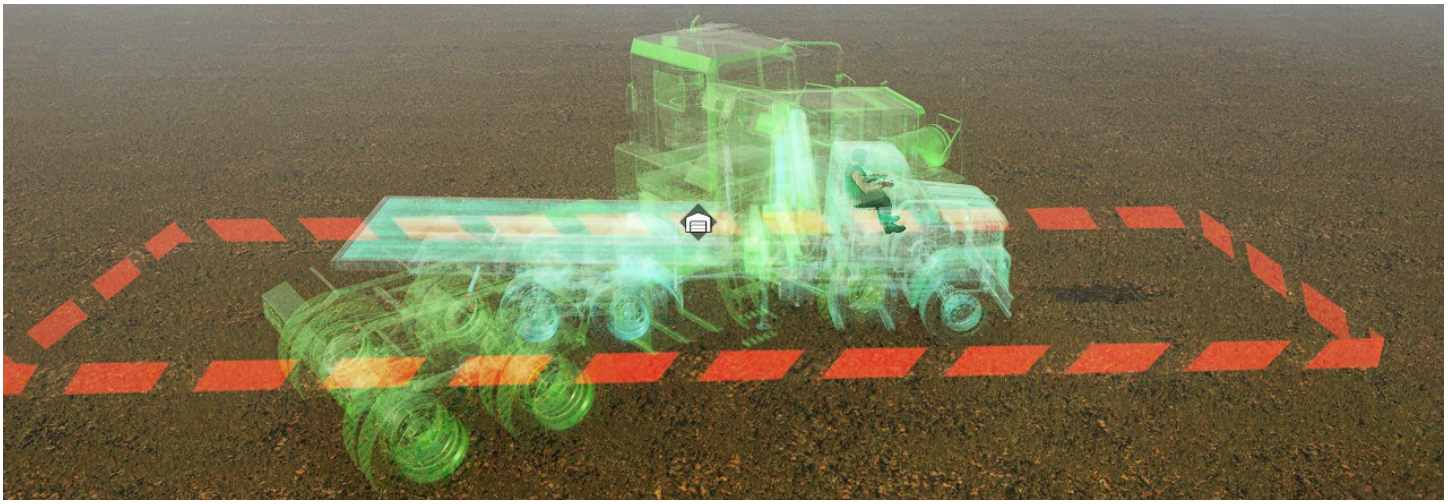
> **Tip:** Clear any icons from Icon 30x30 and Icon 40x40 to prevent a stray icon from appearing over the zone in the driving view.

The truck is centered in the exit zone, facing in the same direction as the zone’s orientation. To avoid a catastrophe, make sure that the largest truck can exit without intersecting an obstacle. If you make a zone of the corresponding dimensions, it will not only help you keep the area clear of stray models, but it will also help guide the player to keep the area clear of parked trucks.

Tip: My measurements indicate that a zone with a Length of 17 meters and a Width of 5 meters (pictured below) will accomodate the longest and widest trucks (including the year 1 DLC). However, extrapolating from Saber's recommendations, 21 meters by 9 meters may be safer for future DLC and/or player mods.



If another truck is in the garage exit zone, both trucks become "ghost" (virtual) trucks that ignore collisions with each other. As soon as the player drives clear of the blocking truck, both trucks become fully "real".



If an model or plant is in the garage exit zone, the player's truck may immediately collide with it and is likely to get stuck on it. Unlike a recovery zone (below), SnowRunner makes no attempt to avoid the collision, so it's up to you to keep the area clear of obstacles on the map.

Saber says that a map "can" contain only one garage exit. My testing shows that a map can contain multiple garage exits, but the game chooses one of them randomly for where to respawn the truck. The random choice is made when you pack the map, so you can't get the fun of randomizing the truck's position each time they exit the garage, either.

Recovery Zone

When the player chooses Recover from the Functions menu, the truck is recovered to the garage if one has been discovered. Otherwise, the truck is recovered to the zone with the ZonePropertyRecover property.

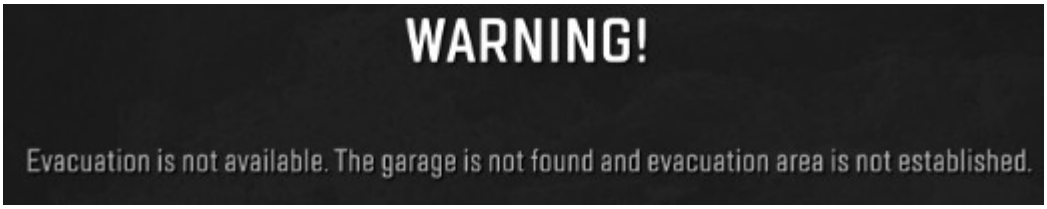
No zone boundary appears for a ZonePropertyRecover zone, and no icon appears in the driving view. The zone is also automatically suppressed from appearing on the navigation map or building list, regardless of the Is Visible On MiniMap property value.

The recovery zone should be placed to avoid obstacles the same as for the garage exit, above. However, the game uses a very different heuristic for placing the truck than it does for the garage exit. When the truck is recovered to the recovery zone, the game automatically shifts the truck's position to try to avoid collisions with obstacles, including other trucks. It never creates a ghost truck.

Tip: Place the recovery zone where the active truck starts so that the player returns to where she started rather than warping to somewhere new.

You can create multiple recovery zones, and the game uses the one with the highest value in the priority property. It never uses a recovery zone of a lower priority, so you might as well create only one recovery zone.

If the garage is not yet discovered, and there is no zone with the ZonePropertyRecover property, then recovery fails. It's not nice to do this to the player.



Trailer Store

A zone with the property ZonePropertyTrailerAttach allows the player to buy or sell a trailer attached to the truck. The forward orientation of the zone doesn't matter.

When the player buys a trailer, it doesn't need to fit within the zone, but the purchase fails if the trailer would intersect an obstacle. Make sure there is sufficient room in at least one direction for a very long trailer such as the superheavy semi-trailer (which adds about 20 meters to the truck length).

Tip: There should usually be plenty of room in the direction of the store entrance for a long trailer. However, the player would often prefer to drive forward out of the trailer store after purchasing a new trailer, so consider leaving enough room for the trailer in the direction opposite the store entrance.

Fuel Station

A zone with the property ZonePropertyFuelStation allows the player to refuel her truck. The forward orientation of the zone doesn't matter.

Service Hub

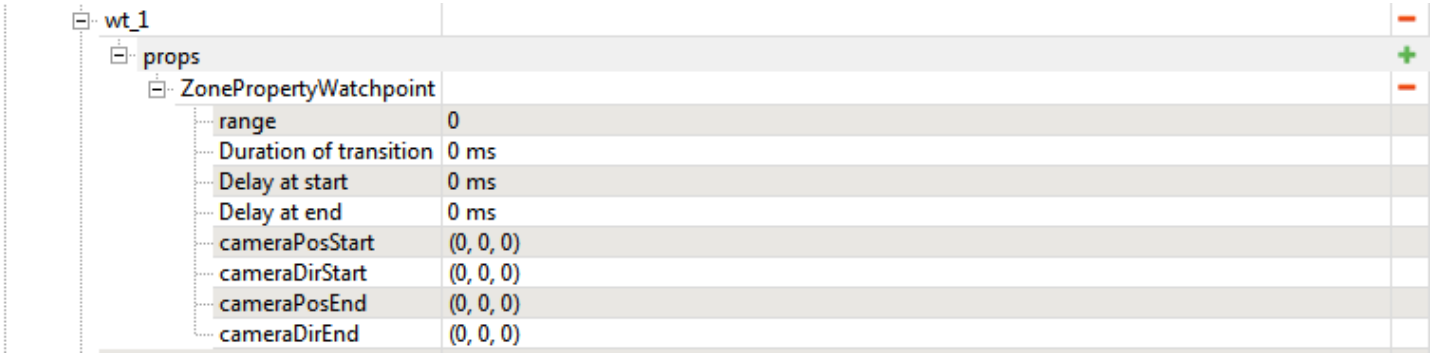
A zone with the property ZonePropertyFuelStation repairs the player's truck. It also restocks both repair parts **and** fuel in any add-ons and trailers, but it doesn't add fuel to the truck's internal tanks. The forward orientation of the zone doesn't matter.

Unlike the fuel station, the service hub doesn't cost money, so it happens automatically without confirmation.

Watchpoint

A zone with the property ZonePropertyWatchpoint reveals the sorrounding region on the map. The forward orientation of the zone doesn't matter.

Unlike the above zone types, extra properties are added below ZonePropertyWatchpoint as soon as it is added.



The range property specifies the radius that is revealed on the map, in meters. See page 225 for more information about revealing the map. Areas within 39 meters of the active truck are revealed without the aid of a watchpoint, so the watchpoint range should generally be significantly larger than that.

The remaining properties specify the camera behavior if the player chooses to Launch Observation:

- cameraPosStart specifies where the camera starts at the beginning of the observation animation.
- cameraDirStart specifies the camera orientation at the beginning of the observation animation.
- Delay at start specifies how long the camera pauses before starting to move.
- Duration of transition specifies how long the camera takes to move to the end position.
- cameraPosStart specifies where the camera stops at the end of the observation animation.
- cameraDirStart specifies the camera orientation at the end of the observation animation.
- Delay at end specifies how long the camera pauses after moving and before the observation ends.

When editing the times, the "ms" string is uneditable. The time is always in milliseconds (thousandths of a second), so e.g. 7000 ms is 7 seconds.

When editing the camera positions and orientations, the three values split into separate x, y, and z fields, each of which can be edited.

Tip: The easiest way get a good camera position and orientation is to position the camera in the Editor, then enable the Statistics toggle button in the toolbar. (Remember to close the Zone Settings Editor first.) The statistics list the camera position followed by the direction: (x; y; z) > (x; y; z).

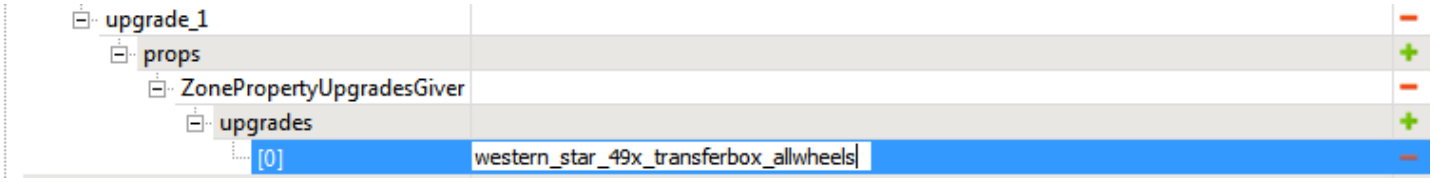
If the camera properties are not changed from the defaults, Launch Observation does nothing.

Upgrade Zone

A zone with the property ZonePropertyUpgradesGiver grants the player with a free part and the ability to buy more. The forward orientation of the zone doesn't matter. The zone disappears after being used.

ZonePropertyUpgradesGiver comes with an upgrades subcategory, to which any number of numbered properties can be added. Click the green + to the right of upgrades to add a numbered property below it.

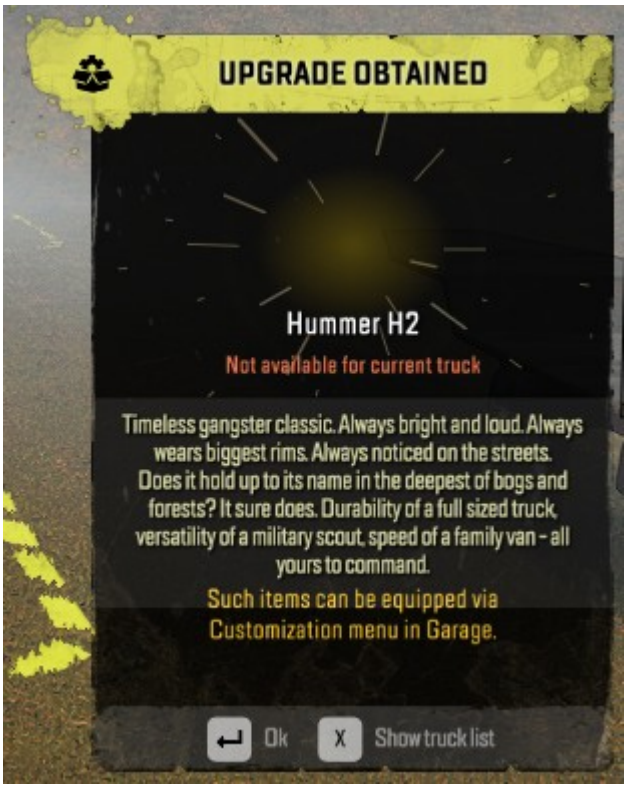
Enter an add-on name in the [0] property to cause the add-on to be unlocked when the player reaches the zone. The player is also gifted with a free part, which is then available to be installed the next time she visits a garage. If the add-on is already unlocked, then the upgrade still grants the player a free one, even if she already owns one or more.



Bug: ZonePropertyUpgradesGiver implies plural upgrades, and any number of numbered properties can be added within the upgrades subcategory, but the game grants only the first upgrade to the player. Everything after [0] is ignored.

Add-ons can also be awarded for completing objectives. See page 186.

You can name a truck type or trailer type in the upgrades property, and the game will unlock it (but not grant a free copy). However, this feature is not documented and displays incorrect messages to the player with a missing icon, so it clearly isn't intended to be used. The normal way to unlock a truck or trailer is to find it on the map.



The various saddle_low and saddle_high addons clearly also aren't intended to be locked or upgraded. A saddle_low simply cannot be locked at all. A saddle_high can be locked, but it is still supplied for free when a trailer that requires it is purchased from the trailer store. TBD: move this to the 'lock parts' section. Add similar text to the 'adjust prices' section.

If upgrades→[0] has an invalid part name, then the game appears to grant the part to the player, although there is no way to buy or install the part.

If no upgrades are valid, the zone is displayed in the game, but the player cannot activate it.

Automatic Cargo Loading Zone

A zone with the property ZonePropertyCargoLoading allows the player to generate packed cargo directly onto a truck or attached trailer. The zone boundary is yellow. The forward orientation of the zone doesn't matter.

ZonePropertyCargoLoading comes with a subproperty platformId. Enter the Id of a manual loading zone to associate it with the automatic loading zone. See the next section for details.

ZonePropertyCargoLoading also comes with a subcategory cargosSettings. Click the green + to the right of cargosSettings to add a numbered category below it. Each numbered category adds a cargo type that can be loaded in the zone.

The name subproperty specifies a cargo type. When editing this property value, you can either type in a name or select a name from the dropdown menu. Note that the menu is missing some names, so you'll need to type those in by hand. Reference information for cargo types is on page 248. If a name is mistyped, that cargo type is ignored.

The Amount subproperty specifies how much of that cargo is available from that loading zone (or the associated manual loading zone). If the Amount is –1, an unlimited quantity is available. If the Amount is 0, the cargo is shown to the player but with the caveat "No more cargo left."

Tip: If there isn't another source for goods with a limited quantity, player mistakes can potentially ruin her game. E.g. if the player generates goods that aren't needed yet, her only option is to use a crane to set them aside for later. If a crane isn't available or the player chooses instead to destroy the goods, then a later objective may be undeliverable. On the other hand, if there is a limited source for goods nearby and an unlimited source further away, the punishment for a mistake is the usual one for SnowRunner: more driving.

The available cargo types are presented to the player in a random order. If the same cargo type is listed twice, only one of them is used; their amounts are not cumulated. If no cargo types are valid, the zone is not displayed in the game.

If the automatic loading zone is enabled on the navigation map, it helpfully shows which cargo types are available at the zone. However, the map can only list 5 cargo types (chosen at random). If there are more than 5 cargo types, the map tells you how many more are available.

Manual Cargo Loading Zone

A zone with the property ZonePropertyManualLoading allows the player to generate unpacked cargo onto the ground. The player can then use a crane to manually load the cargo onto a truck. The zone boundary is red. The forward orientation of the zone matters.

The manual cargo loading zone must have the ZonePropertyManualLoading property and must be associated with an automatic cargo loading zone. Enter the manual zone's Id in the platformId property of the automatic zone to associate them. The player can then enter the automatic loading zone and choose whether to generate goods directly onto the truck or into the manual loading zone.

Since the manual cargo loading zone is effectively controlled by the paired automatic cargo loading zone, it has no additional properties of its own.

If a cargo type has limited quantity in the automatic loading zone, then the available quantity is reduced whether the goods are generated in the automatic or manual loading zone.

The manual loading zone always has a "hook" icon above it in the driving view, regardless of any icons assigned to the zone properties. (Oddly, this hook is different than the ones available for named icons.) The zone is automatically suppressed from appearing on the navigation map or building list, regardless of the Is Visible On MiniMap property value.

It is not possible to have a manual cargo loading zone without an associated automatic cargo loading zone. However, if you'd like to force the player to use the manual loading zone, you can hide the automatic loading zone. E.g. make the automatic loading zone small and put it under a platform. It appears that the game always gives priority to the automatic loading zone's icon over the manual loading zone's hook icon. That's good because you need the automatic loading zone's name and icon for the navigation map. Make sure to recommend a truck with a crane for any associated objectives (page 186).

Cargo Generation Location

Caution: Cargo is generated with a different orientation than cargo that is spawned for an objective (page 203)

If we assign a "forward" direction for cargo based on how it is loaded on the truck, then the cargo is generated in the manual loading zone pointing 90° to the right of the zone orientation. I.e. long goods are oriented with the long end perpendicular to the zone's direction.

Tip: The zone's dimensions have no effect on where the cargo is generated. But setting the dimensions to the maximum size of potential cargo helps inform the player where to expect the cargo to appear and where to avoid parking a truck.

Cargo Generation Height

The game attempts to generate cargo just above ground level. If the ground is hilly, it tries to generate the cargo high enough to not be inside the ground. If it detects a model in the way, it generates cargo just above the model. Its heuristic isn't very good, though. It does best when the ground is very flat and any model underneath is well centered and also flat (such as the various platform models).

Tip: To perfectly center a model in the zone, simply copy the zone's Org X and Org Z values to the model's X and Z properties. Unfortunately, the zone's direction isn't even in the same format as the model's, so you're on your own there. The default orientation for zones (pointing along the X axis) happens to match the default orientation for plaform models, however, so you're all set if that orientation works for you.

The zone's Height property has no influence on the cargo generation height. However, if Is Wall is True, the Height property is useful to position the zone ribbon just above a platform model.

Logs

The height heuristic for generated cargo performs poorly with the standard log trestle (logs_scavange_02), but it compensates by generating a log 2 meters higher than it otherwise would. This gives the log room to drop into place.

Unlike other cargo types, logs can share space with other cargo in the manual loading zone (particularly with other logs). The game is weirdly inconsistent about when it declares that the zone is “full” or when it asks for confirmation to delete other cargo, but it does always consistenly allow at least three logs of the same type on a trestle. (A log crane has trouble picking up more than two logs at a time, anyway.)

Crafting Zone

- ZonePropertyStorehouseCraft: TBD.
- ZonePropertyEnergy: TBD.

Multiple Zone Types

Multiple types can be assigned to a single zone. For zone types that do something automatically, all automatic effects are invoked. For other zone types, the player can switch among the various interaction options.

Tip: The player might not notice that she can switch to another interaction option. The main campaign uses multiple interaction options only when the additional option is for a delivery goal, which adds an icon to the navigation map and driving view to help make the extra option more obvious.

Objectives

Because a SnowRunner objective usually involves visiting multiple zones, objectives are not specified within a zone. Instead, objectives get a separate objectiveSettings category in the Zone Settings Editor. The objective properties are also saved in a separate file, objective_settings.json.

An objective can take one of three forms:

- A contract is activated automatically when unlocked, without the player needing to visit a particular zone. Contracts are organized within the game based on employer offering the contract.
- A task is offered when the player visits a particular zone.
- A contest is also offered when the player visits a particular zone, and it offers different rewards depending on the speed of completion.

All three objective types are very similar. Any differences are discussed where appropriate in the sections below.

Objective Status Terminology

I use the following terms to describe the status of objectives:

- Locked: The player does not have the required rank to activate the objective, or it is blocked by another objective.
- Unlocked: The player meets all of the requirements for an objective to be offered or activated. An unlocked task or contest is offered when the player drives to the starting zone. An unlocked contract sometimes needs to be manually activated, but usually it automatically moves to the Activated state.
- Offered: The game offers an unlocked task or contest when the player visits its starting zone. If the player accepts the offer, the objective is activated.
- Activated: The player can perform the stages of the task. Multiple objectives can be activated at once with different levels of progress in each.
- Tracked: The goals of the objective's current stage are displayed in the upper right corner of the screen. Only one objective can be tracked at a time, and the player can get credit for completing a goal only when the objective is being tracked.
- Conpleted: The player has completed all stages of the task and received the reward.

Custom Employers

Contract objectives are associated with employers. The game has a number of built-in employers, described on page 240. Or you can create your own employer using one of the two method below.

New Employer

> objectiveSettings.employer+*Employer Name*
>
> Adds a new employer.

Within the *Employer Name* property are subproperties for Employer Logo and Employer Background. However, these are not yet supported. The new employer uses a default image identical to Husky Forwarding.

The standard employers use all capital letters in their names, but you can use lowercase if you want.

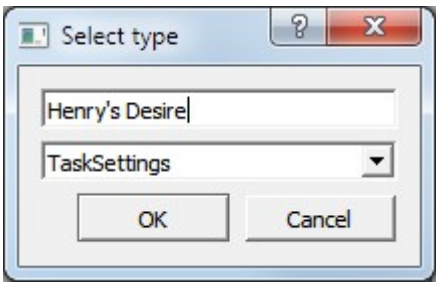
Modified Employer

If you like the icon of one of the built-in employers, you can use localization (page 147) to change its name while keeping its icon. The localization ID for each employer is listed on page 240. If you include this localization ID in your map's localization files, it will override the default name (in each language).

Add Objective

To add an objective, start the Zone Settings Editor and open the objectiveSettings category. Add a new objective by clicking the green + to the right of tasksSettings, contractsSettings, or contestSettings.

The resulting dialog box looks similar to the one used for ZoneProperties, but is used very differently. In the top field, enter the name for the objective.



Bug: The dialog box is called "Select Type", but you actually use it to name your objective.

The objective name is used for the game to internally track the status of the objective, and it is also the name shown to the player. After the objective is added, you can edit its name by double-clicking it. The objective name is UI text, described on page 146. Formatting tags are not supported.

> **Tip:** Localization potentially allows you to show the player the same name for objectives with different internal names, although this could be slightly confusing to the player.

The bottom field of the dialog box has a dropdown menu, but it only ever has one entry which matches the type of objective you are adding.

Requirements

An objective is not offered or activated until it meets all requirements below.

Required Rank

The Required rank property specifies the minimum rank required to unlock the objective. The default Required rank is 0, which is less than the minimum rank that the player can possibly have, so the objective is unlocked as soon as the player starts the map. (A Required rank of 1 works equally well.)

Blocker Objectives

Click the green + to the right of Blocker Objectives to add one or more blocker objectives. Each subproperty gives the name of one objective. An objective is not offered or activated if any of its blocker objectives are incomplete.

Offer Zone (Task or Contest)

A task or contest requires an offer zone. When the task or contest is unlocked and the player enters this zone, the game offers the objective to the player.

A task or contest can be activated as soon as the player enters the map (page TBD), but it still requires an offer zone where the player can restart the objective if desired.

The primary offer zone is specified in the taskGiverZone property. This property requires a global zone ID, described below.

Additional zones can be specified by adding properties under the additionalTaskGivers category. These properties require a regular zone ID.

When a task or contest is unlocked, its offer zone appears with a yellow perimeter. If the offer zone is shared with a zone type that normally has a red perimeter, the perimeter turns yellow until the objective is completed.

Bug: If two objectives are given by the same zone, accepting one may accept the other, **even if the other should not being offered** (e.g. because it is blocked or completed).

Tip: One workaround is to create duplicate zones with the same name and location, but different zone IDs.

Bug: If two objectives are simultaneously offered in the same location, the game only offers one to the player. The player cannot activate the second until she completes the first.

Tip: Avoid offering objectives in the same location at the same time. E.g. use Blocker objectives to offer them at separate times.

Bug: If the player's truck straddles the offer zones of objectives with adjacent locations (the same objective in both zones or two different objectives), the game sometimes gets confused and may offer neither objective until the player leaves both zones and then returns to just one of them.

Tip: Make sure that objectives are offered sufficiently separate in space to prevent straddling or use blocker objectives (in the previous section) to offer them at separate times.

Tip: It is also helpful to offer objectives separate from other interactive zones so that the user doesn't need to switch among interaction types. This separation doesn't need to be as far, however, since the player can more easily identify the problem and the fix.

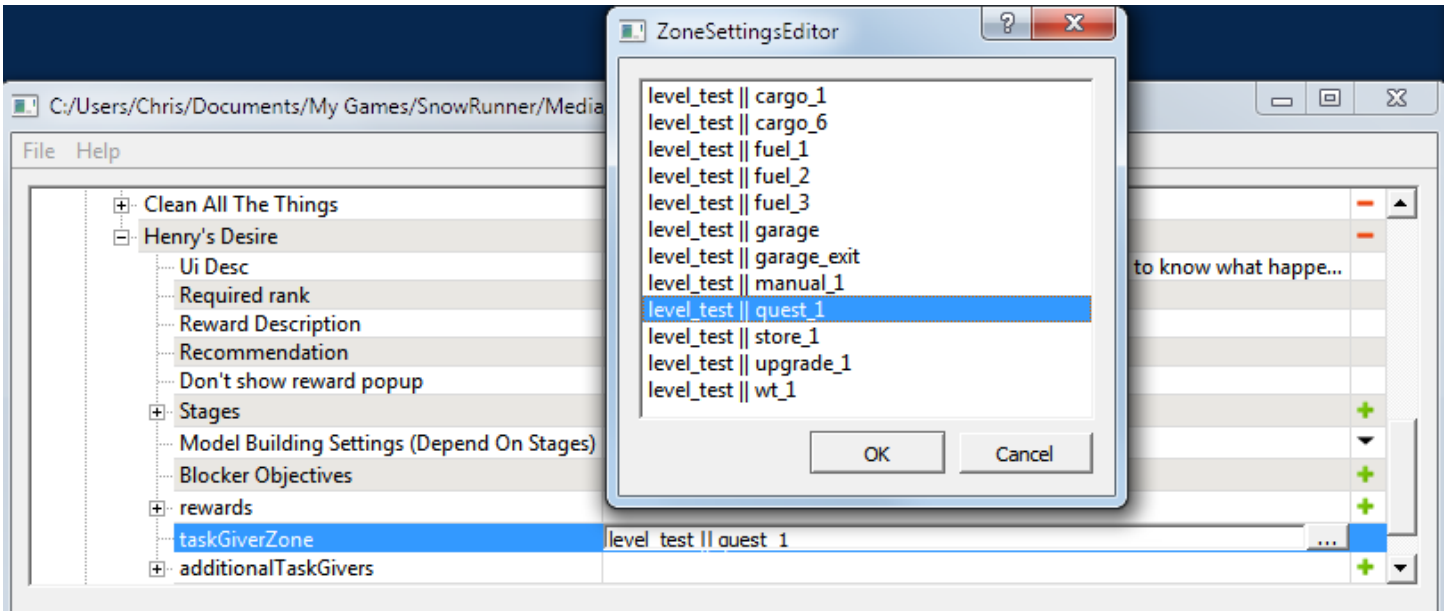
A contract doesn't have an offer zone. Instead, a contract is offered from the navigation map by an employer (page 184).

Global or Regular Zone ID

Objectives can cross maps within a region. As such, certain properties require a global zone ID in which both the map name and the zone ID are specified. The map name is separated from the following zone ID by two vertical pipe characters, e.g. <level_name> || <zone_id>.

If the map is implicit, only a regular zone ID is needed. In this case, the map name should **not** be attached to the zone ID.

For either case, when you select the zone ID value for editing, a small button labeled … appears to the right. Click the … button to open a dialog listing the known global or regular zone IDs, as appropriate. From this dialog, you can simply select the desired zone without worrying about its format.



Employer (Contract)

A contract requires an employer who offers the objective. If an employer is not specified, Husky is used as the default. But even if you want Husky, you should probably specify the employer explicitly to avoid trouble down the road.

The employer can be one of the built-in employers or a new employer. The names and other reference material for each employer is on page 240. Note that some built-in employers are missing from the property dropdown menu, but you can still type in their names manually.

A new employer requires additional configuration (page 181).

Bug: If a new employer is not properly configured or if an employer name is mistyped, the game **might** hang when the navigation map is opened. Or it might wait until you've edited the map a few more times, so it can be very mysterious what is causing the game to hang. Be very careful with your employer names.

Objective Description

The Ui Desc property specifies the objective's description that is shown to the player. The objective description is UI text, described on page 146. UI text is described on page 146. Formatting tags are not supported.

Bug: If the UI Desc value is blank, the objective details won't show a description, but the offer popup will show the description of the previous offered objective!

When the player views the objective offer, the game shows an overview of the objective, including its description. The description is also displayed when the player views the objective details from the navigation map.

The objective offer shows an icon for each goal type required by the objective. If an objective has multiple goal types, multiple icons are displayed.

Recommended Equipment

The Recommendation property allows you to choose a recommended piece of equipment for the player to have. A dropdown menu provides the following choices:

- NONE: The area for recommendations is left blank.
- CRANE: A truck with a crane is recommended.
- OFFROAD: An offroad truck is recommended.
- SCOUT: A scout truck is recommended.
- SEISMIC: A truck with a seismograph is recommended.
- METAL_DETECTOR: A truck with a metal detector is recommended.

Bug: A metal detector is used to detect loose cargo, so it implies that a crane is also needed. However, there is no way to tell the game to recommend both a metal detector and a crane.

Bug: The Zone Settings Editor doesn't have a menu item to allow recommendation of a truck with a log-loader crane. To recommend a log-loader crane, hand edit the objective_settings.json file (page TBD) and change the recommendedTruck value to a numeric (unquoted) 6. This will display as a blank value in the Zone Settings Editor (distinct from the default NONE string). As long as you don't edit the Recommendation value, the recommendation will be retained as you make other edits in the Zone Settings Editor.

Rewards (Task or Contract)

Click the green + to the right of rewards to add a reward. A dialog opens with a pulldown menu to select a reward type:

- ObjectiveRewardExperience: awards some amount of experience toward the next rank.
- ObjectiveRewardMoney: awards some amount of money.
- ObjectiveRewardItem: awards an item.
- ObjectiveRewardOpenZoneProperties: adds a new type to a zone.

The Experience and Money reward types have a subproperty for the amount.

The Item reward type has a subproperty for the Item name. The item can be a truck type, in which case it goes to the garage storage area. Note that a truck can also be awarded via a truck delivery goal (page 209). Alternatively, the item can be an add-on, in which case it goes into the player's garage inventory.

Bug: If you award a trailer type, then the next time that the player views her garage storage area, the game will crash, presumably because there is an unexpected trailer in the garage. The safe way to award a trailer is via a truck delivery goal (page 212).

Tip: Use the reward description (below) to remind the player to check the garage storage area for an awarded truck.

Any number of rewards can be awarded. However, the game can list no more than one Experience reward, one Money reward, and one Item reward, and which reward it lists of each type is not consistent throughout an objective. You should probably avoid multiple rewards of the same type. That means also avoiding an Item reward if a truck or trailer is awarded by a goal.

The OpenZone reward type is typically used to add a type to a zone that did not have one before. This makes the zone visible with the new type. Instead of specifying a global zone ID, separate subproperties are created to hold the map name and zone ID. The props subproperty allows you to specify the zone type (or types) and its properties (as on page 169).

Bug: A new watchpoint type permits the player to launch an observation, but it does not actually reveal any of the map, regardless of the range specified.

The game does not tell the player that the new zone type is available.

Tip: Use the reward description (below) to let the player know that the new zone is open.

Rewards (Contest)

Click the green + to the right of rewards to add a set of rewards for gold, silver, and bronze. Although you can have multiple reward sets, they suffer from the same complications of multiple rewards of the same type in the section above.

Bug: If you don't have any rewards for a contest (i.e. no ObjectiveRewardsByTime property), the game will hang when it tries to load your map.

Each trophy has an associated Time limit, but only the gold and silver time limits are used. The bronze time limit is ignored:

- If the player completes the contest within the gold time limit, the gold rewards are awarded.
- Otherwise, if the player completes the contest within the silver time limit, the silver rewards are awarded.
- Otherwise, if the player completes the contest without failing, the bronze rewards are awarded.

The silver time limit should be greater than the gold time limit. Otherwise, the silver trophy is unachievable, and the contest offer will look weird.

Experience, money, and item rewards are the same as in the section above. A contest cannot open a zone as a reward. A contest can act as a blocker objective, in which case it unblocks dependent objectives the first time it is completed with any trophy.

Bug: The game awards the listed rewards regardless of how many times the player has achieved any trophies. This is the only case in which the player can "grind" an objective to get multiple rewards. The game tries to limit the damage by restricting hard mode to a limited number of attempts, but that's a poor kludge. It would be much better if the game only awarded the **difference** between an improved result's rewards and the previous best rewards.

Bug: An item reward associated with trophy is not awarded if the player gets a better trophy. You're should probably have only one possible item reward, which is listed for each trophy at the desired level and above.

The contest offer dialog includes the required times, the experience and money rewards. Item rewards are not listed in the offer or the reward dialog, so you should include them in the contest description.

Fail Reasons (Contest)

There are three fail reasons that are always present and cannot be deleted:

- RecoveryFailReason: the player fails the contest if she recovers her truck to the garage or starting area.
- ChangeTruckFailReason: the player fails the contest if she changes trucks.
- GarageFailReason: the player fails the conest if she enters a garage.

Click the green + to the right of failReasons to add one or more additional fail reasons. When you add a fail reason, a dialog box presents a dropdown menu with four fail types. Although you can repeat a fail type more than once by giving it different names, there is never any useful reason to do so. The optional fail types are as follows:

- CockpitFailReason: the game automatically switches to cockpit view when the contest starts, and the player fails if she changes views.

- DamageFailReaon: the player fails if she accumulates more than maxDamage points of damage.
- HoursFailReason: the game automatically switches to time given by startHour, and the player fails if time reaches the finishHour. Hours are integers and should be from 0 – 23, inclusive. Values outside this range are treated as 0 (midnight). The game does the right thing if startHour is greater than finishHour: it ends at the finish hour in the next day. The game fails with a ModMapError if startHour is equal to finishHour. One hour of game time is equivalent to 150 seconds (2m30s) of real-world time.
- TimeFailReason: the player fails if the contest runs for more than timeLimitSec seconds. This is the real lower bound to achieve the bronze trophy, although the game never presents it as such.

The game lists up to three of the additional fail reasons in the contest offer dialog. If there are more than three, it does not list them all.

Bug: If HoursFailReason is given, the game always says “Complete during night”, even if the hours given are daytime hours. The game also doesn’t tell the player how many hours the contest runs.

Bug: If TimeFailReason is given, the game says only “Time limit present” without specifying the limit.

Reward Description

The Reward Description property specifies the text that is shown to the player when the objective is complete. UI text is described on page 146. Formatting tags are not supported.

Despite its property name, the reward description is probably used more often to comment on the job than on the reward.

The reward popup can be suppressed by ticking the checkbox in the Don’t show reward popup property. Music still plays when the player completes the objective, but no information appears onscreen. The Reward Description property isn’t used if Don’t show reward popup is checked.

Once an objective is completed, it is removed from list of objectives next to the navigation map. Any zones that were only used for a completed task or contract are hidden. However, the offer zone for a contest remains on the

Since a contest can be accepted again after being completed, it remains in the list of objectives next to the navigation map, and the offer zone remains visible in the driving view or on the navigation map when the contest is selected.

Multi-Stage Models

Multi-stage models (page 90) change state in response to completion of an objective stage (this section) or delivery of cargo (page 201). The change can be purely visual, or it can remove a barrier that was blocking

trucks from passing. Some models have a fancy animation to show a change, while others simply swap to the new shape with a sound effect and a cloud of dust.

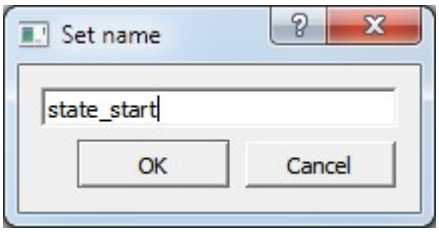
Most multi-stage models have the suffix _objective and so can be easily found in the model asset list. A reference to the available multi-stage models is on page 252.

Click the black arrow to the right of Model Building Settings (Depend On Stages) to connect a multi-stage model to the player's progress in the objective's stages.

The levelName property is hard-coded to the current map name. ?TBD

The modelTag functions as an ID for models. It should be set to the same value in the model (page TBD) as it is here in the objective. An objective can change only a single model in response to completion of the objective stages. Also, a model tag cannot be used by more than one objective, nor by more than one property within an objective.

To specify what state a model to change to at the end of each stage, click the green + to the right of stagesProgress. A dialog pops up where you can enter the name of the model stage that you want to tie to an objective stage.



Once a model stage name is entered, you can change it by double clicking it in the property list.

The easy way to get model stage names is by checking the reference on page 252. The hard way is to open the initial.pak file in the SnowRunner installation directory, find the model XML file under [media]/classes/models (or in one of the [media]/_dlc directories), and then find the name attributes specified in each <Subset> tag. But you shouldn't need to use the hard way unless you want to use a newer model that I haven't yet added to my reference.

Each subproperty under stagesProgress is arranged as a property name with a property value. The property name is a model stage name, and the property value is the objective stage number to associate with that model stage. Because property names must be unique, a model stage name can only be associated with a single objective stage number.

Double click a stage number to edit its value. Positive integers are associated with completion of the corresponding stage. E.g. when the player completes the first stage of an objective, the model switches to the model stage associated with objective stage 1.

Whaever model stage is associated with objective stage 0 is the initial state of the model before any objective stages are completed. If you don't associate any model stage with objective stage 0, then the game initializes the model to its default stage (the first stage listed in the reference).

When a model changes shape, there is a chance that its new shape could collide with a truck. Most of the premade models are safe from this, as they either don't change their collision boxes or they only remove collision boxes. But a few models add collision boxes, so you may want to place obstacles or use impassable terrain to keep the player from parking a truck there. And certainly keep the stage completion zone(s) away from where the model is about to move or appear.

You don’t have to use model stages in their default order. However, there is no animation or sound effect when a model makes a non-standard transition. Also, reversing model stages has an increased chance of causing a collision (e.g. if you reverse the stage order to add a barrier instead of removing a barrier).

If an objective removes a model barrier and then continues with additional stages, the player may choose to restart the objective. If the player has parked a truck where the barrier was, then this can also cause trouble. In this case, though, the player should be able to rescue her truck (perhaps with significant damage) by completing the stage to remove the barrier again.

Objective Stages

An objective is split across one or more stages which the player must complete sequentially. Add a stage by clicking the green + to the right of the Stages container.

Each stage is comprised of one or more goals which the player may complete in any order. The goals in a stage can all be the same type or a mix of types.

For each added stage, all of the available goal types are listed. The method of adding each type of goal and the details for each type are described below.

In addition to having goals, a stage can also spawn cargo as described on page 202.

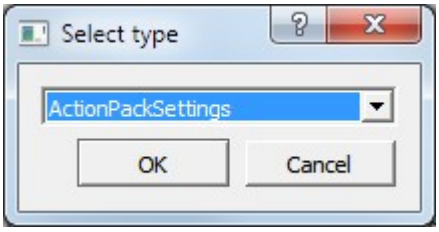
Conflicting Properties

Goals in one or more stages can occasionally have properties that apply to the same resource. E.g. each truck delivery goal specifies the name of a truck, so if multiple goals refer to the same truck, then only one of those names can be applied. The property that gets applied is often random, and in some cases the random result is not consistent. E.g. the truck can have one name on the map and another name in the objective details. In most cases, I haven't tried to determine if there is a pattern to how conficting properties are applied. Instead, I simply recommend that all properties **should** have the same value.

Cargo Delivery Goal

A stage can require the player to deliver certain cargo to one or more zones.

To create a cargo delivery goal, click the black arrow to the right of actions. This pops up a useless dialog. Click OK to confirm it.



The actions category now has a zoneToFill subcategory. Click the green + to the right of zoneToFill to add one or more cargo delivery goals.

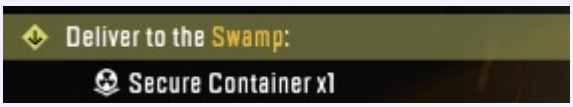
Stages	
[0]	
actions	ActionPackSettings
zoneToFill	
[0]	
Ui Desc	Deliver to the <y>Swamp</>:
Truck need to visit Uid:	
globalZoneId	level_test || store_1
Manual Unloading Settings	null
cargo	CargoPack
name	CargoRadioactive
Amount	1
Model Building Settings (Depend On Cargo)	null
additionalUnloadZones	
truckDelivery	
visitAllZones	null
changeTruck	null
repairTruck	
Seismograph Settings	null
spawnCargoOnStageActive	
livingAreaInfo	null
Model Building Settings (Depend On Stages)	null
Blocker Objectives	

If you decide that you don't want a cargo delivery goal after all, click the black arrow to the right of actions again. A useless dialog offers to change the property back to null, which removes all of the subproperties.

Goal Description

The UI Desc property describes the goal for the player. It supports formatting tags. UI text and formatting tags are described on page 146. If the Ui Desc value is empty, the game displays an error instead.

Tip: The main campaign uses the <y> tag to highlight named locations on the map in the manner shown above, and you may wish to do the same. Notice how the : sets up the cargo to deliver, which the game always writes immediately after the goal description.



Cargo to Deliver

To specify the cargo to deliver, click the black arrow to the right of cargo and confirm the useless dialog. Specify the cargo type and amount in the subproperties that appear.

The name subproperty specifies a cargo type. When editing this property value, you can either type in a name or select a name from the dropdown menu. Note that the menu is missing some names, so you'll need to type those in by hand. Reference information for cargo types is on page 248.

The Amount subproperty specifies how much cargo is required to complete the delivery. The player is allowed to deliver less than the full amount at once, so she can make multiple trips to fulfill a large objective.

Bug: The properties don't support multiple cargo types within a single goal. Instead you must add more goals under zoneToFill, each with their own description, etc.

Delivery Zones

The globalZoneId property specifies the global zone ID for the primary delivery zone. If the player is allowed to deliver the cargo to other zones instead, these are specified with regular zone IDs in additionalUnloadZones.

Tip: additionalUnloadZones is useful for making a pair of delivery zones on either side of a blocked passage.

If there are multiple delivery goals in the same stage, all of them get the same yellow deliver icon on the navigation map. The player can select each goal description in the objective details to center the navigation map on that goal's primary delivery zone. However, the player has no way to determine which additional delivery zones apply to which goals. Consider making all deliveries share the same set of additional delivery zones or otherwise keeping things simple for the player.

If multiple cargo delivery goals in a stage have the same primary delivery zone, the game combines them all under a single goal description (UI Desc). In this case, you should use the same goal description for all such goals.

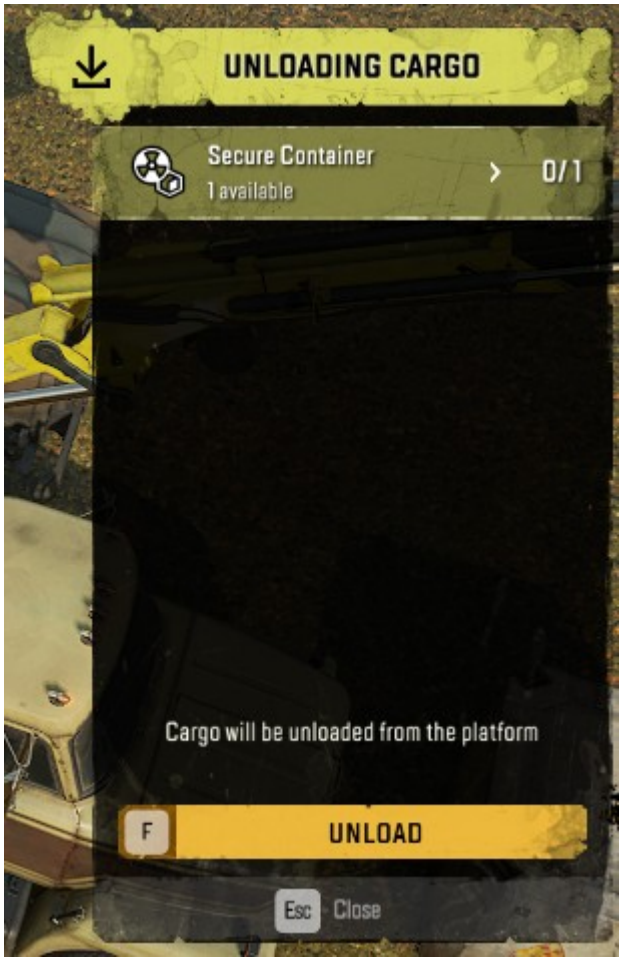
If Manual Unloading Settings is null, the player can only complete the delivery while the required cargo is packed on her truck or trailer.

Manual Unloading

To require manual unloading, click the black arrow to the right of Manual Unloading Settings and confirm the useless dialog. This adds a subproperty, Interaction zone Uid:. Fill this subproperty with a global zone ID. In order to complete the goal, the player's truck must be in this interaction zone (Interaction zone Uid:), **and** the cargo must be unpacked and within the primary delivery zone (globalZoneId).

Bug: When manual unloading is required, the zones in additionalUnloadZones are made visible as if they are additional delivery zones, but they offer the cargo management dialog as if they are additional interaction zones. However, these zones don't actually work for either function: cargo unloaded in them is not recognized as deliverable, and their cargo management dialog never recognizes when the cargo is correctly placed in the delivery zone. So keep additionalUnloadZones blank when using manual unloading.

The below screenshots show the cargo management dialog before and after the cargo is correctly unloaded. Your milage may vary, but I found the grammar of the first dialog extremely confusing until I noticed the tiny red period that ends the first sentence.



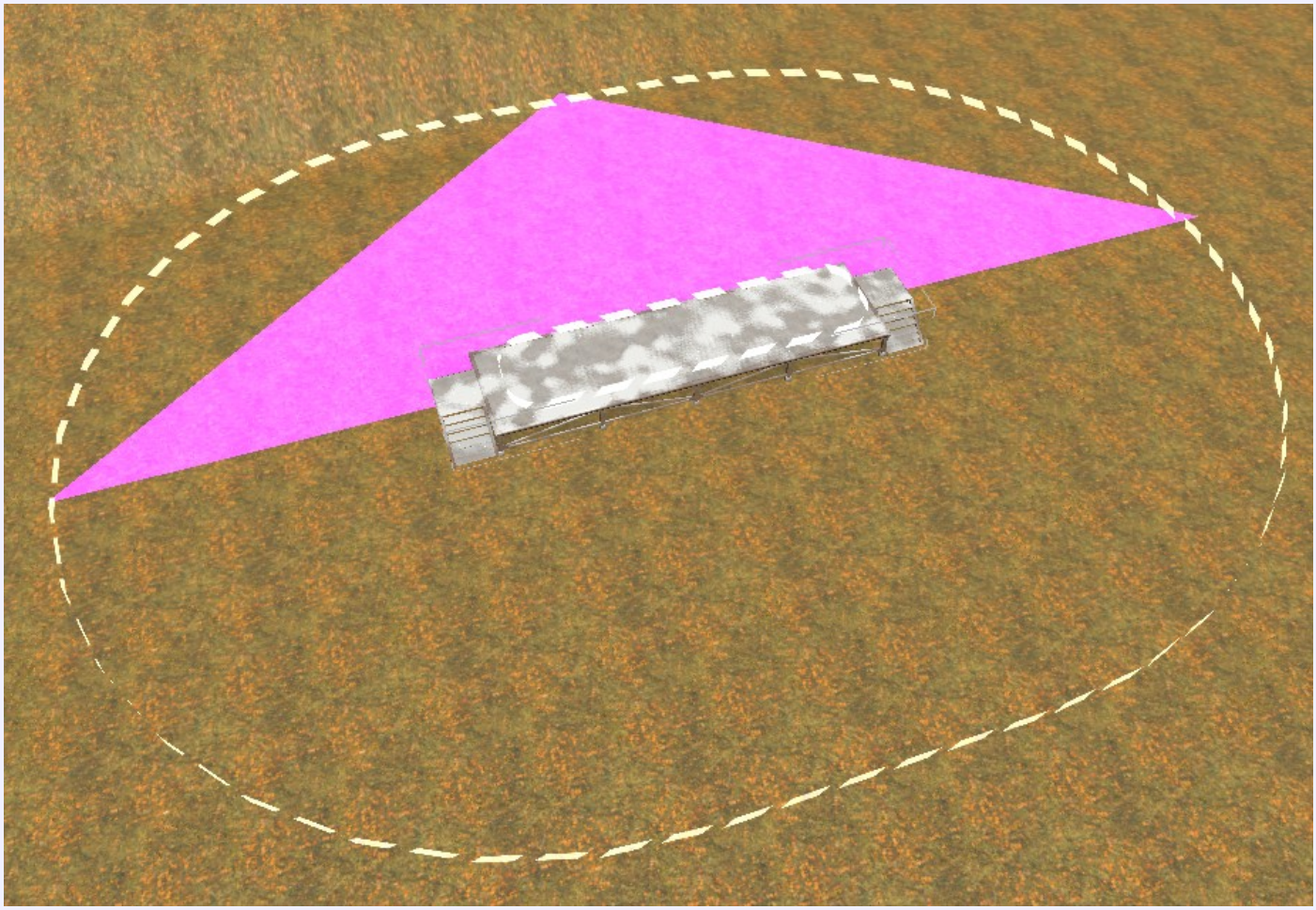
The delivery zone is made visible in the driving view while the objective is activated.

Warning: The interaction zone (Interaction zone Uid:) is not visible in the driving view (unless the zone has another purpose that makes it visible).

Saber recommends that the interaction zone entirely surrounds the delivery zone so that as soon as the player finishes unloading the cargo, she can make the delivery from the cargo management dialog offered by the invisible interaction zone.

Tip: The interaction zone should be large enough so that the delivery can happen even after a long stretch with a big crane, but small enough so that the delivery zone is obvious when the cargo manangement dialog is offered. I find that the longest crane stretch for both US and Russian is about 13.5 meters. Appropriate interaction zone dimensions can thus be calculated as follows:

- interaction zone Length = delivery zone Length + 27
- interaction zone Width = delivery zone Width + 27
- interaction zone Rounding radius = delivery zone Rounding radius + 13.5



If the player can drive into the delivery zone, she can potentially make the delivery by simply unpacking the cargo within her truck without actually unloading it.

Tip: Place the delivery zone on a raised platform (such as the various platform models) that the truck cannot climb onto.

The interaction zone and delivery zone are both shown on the navigation map when the objective is being tracked or when the objective details are viewed.

Tip: To avoid a pile-up of names and icons, set the interaction zone's Is Visible on Minimap property to False.

Manual unloading goes well with cargo that is spawned for the stage and thus requires manual loading. See page 202 for information about cargo spawning.

Required Truck

The Truck need to visit Uid property is optional. If filled, it should refer to a truck's unique ID (page 67), and the truck should not be reserved (page 69). The player is then allowed to make the cargo delivery only in this truck. Note that the ID refers to a specific truck, not a truck type. If a truck type is put in this field, then it won't match a truck ID, and so no truck can make the delivery.

The specified truck should not be reserved for another goal. Reserved trucks are described on page 69.

Tip: Use the Blocker Objectives property to ensure that a reserved truck is rewarded before offering an objective that requires the player to drive it.

Bug: The game never tells the player which truck is required, or even that a specific truck is required at all. Instead, when the player arrives at the delivery zone with the cargo in the wrong truck, the game either doesn't offer the cargo management dialog, or it allows the player to select the cargo to deliver and then ignores they keypress to deliver it. You should tell the player in the UI Desc which truck to use.

Bug: If the player ever visits a garage with that specific truck, its truck ID is lost. I.e. when the player leaves the garage, the game doesn't restore the original truck, but instead spawns a new truck of the same type. So to use Truck need to visit Uid with any safety, you need to make sure that no garage is unlocked. (Even if the truck is unable to travel to a garage, the player could still recover the truck to a garage.) Note that the truck ID is **not** lost if the player recovers to the start zone.

Bug: If the Truck need to visit Uid refers to a trailer, the delivery goal cannot be completed, even if the load is on the required trailer.

Multi-Stage Models

Multi-stage models (page 90) can change state in response to delivery of cargo. This is very similar to model state changes in response to completion of objective stages (page 190), where they are explained in detail. This section only describes the differences in how model stages are associated with cargo delivery.

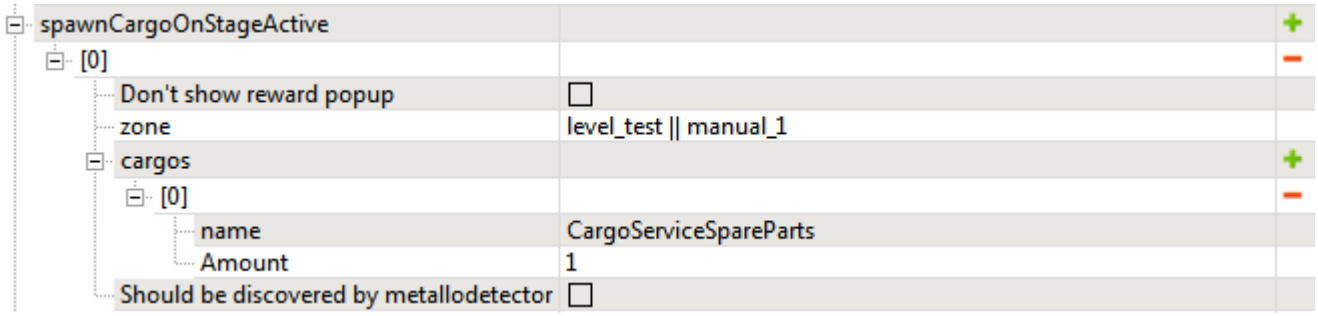
Click the black arrow to the right of Model Building Settings (Depend On Cargo) to connect a multi-stage model to the player's progress in the objective's stages. After setting the modelTag property, click the green + next to stagesProgress to add model stage names.

Cargo is either delivered or not delivered, so it can only switch a model between two states. The number next to a model stage name must therefore be either 0 or 1. A model stage name associated with a value of 0 sets the initial state of the model. A model stage name associated with a value of 1 specifies the state of the model as soon as cargo is delivered.

Spawn Cargo for Stage

Loose cargo can be spawned when a particular objective stage begins. This is the best mechanism for encouraging the player to manually load cargo for a cargo delivery goal in the same stage. Since the cargo is respawned if the stage is restarted, the player can never get stuck without sufficient cargo.

To spawn cargo when a stage is started, click the green + to the right of spawnCargoOnStageActive. You can add as many spawned cargo as you like.



Set the zone property to the global zone ID of the zone where the cargo should spawn.

The game attempts to spawn cargo just above ground level. If the ground is hilly, it tries to spawn the cargo high enough to not spawn inside the ground.

Bug: Unlike a manual cargo loading zone (page 177), spawned cargo is not placed above any conflicting model.

Unlike a manual cargo loading zone, there is no confirmation dialog to delete other cargo in the zone. Instead, cargo fails to spawn if other cargo is in the way.

Bug: The game does not indicate to the player that cargo failed to spawn, nor does it indicate **where** it failed to spawn, so the player has little hope of figuring out how to correct the situation. If the cargo spawns near where the stage starts (e.g. the task or contest offer zone), the player **might** figure out that the stray cargo she can see is obstructing the spawn of cargo she can't see, but then again she might not.

Interestingly, if a truck is in the spawn zone, it doesn't prevent cargo from spawning. If the cargo intersects a truck, it is displayed as a "ghost", and only becomes real when the truck is moved out of the way. The ghost cargo is shown on the map the same as if it is real.

Tip: Spawn cargo only in an area where it is highly unlikely that the player will drop loose cargo. Make the spawn zone's length and width very small or perhaps 0, although cargo can obstruct even a zero-size zone if the cargo is close enough.

Click the green + to the right of cargos to add exactly one entry to the category. Only one cargo item can be spawned in a zone, so there's no point in listing more. Set the name of the cargo type to spawn as usual. The Amount is hard-coded to 1 and cannot be edited.

The game works best when the spawned cargo types match the cargo delivery requirement for the same stage and when there is no other source for the same cargo type. In particular, nothing stops the player from ignoring the spawned cargo and instead delivering some other cargo of the same type. If spawned cargo is not needed, it remains on the map after the stage is completed.

If the checkbox next to Should be discovered by metallodetector is enabled, then the spawned cargo cannot be seen on the map until discovered by the metal detector. Details TBD.

Bug: There is a checkbox for Don't show reward popup, but it doesn't below here and doesn't do anything.

Cargo Spawn Location

Caution: Cargo is spawned with a different orientation than cargo that is generated in a manual loading zone (page 177)

If we assign a "forward" direction for cargo based on how it is loaded on the truck, then cargo is spawned in the target zone with the cargo pointing in the same direction as the zone orientation. I.e. long goods are oriented with the long end parallel to the zone's direction.

Cargo Spawn Height

The game attempts to generate cargo just above ground level. If the ground is hilly, it tries to generate the cargo high enough to not be inside the ground. The zone's Height property has no influence on the cargo spawn

height. **Caution**: Unlike in a manual loading zone (page 178), cargo is not spawned above an obstructing cargo model.

Logs

The height heuristic for generated cargo performs poorly with the standard log trestle (logs_scavange_02), but it compensates by generating a log 2 meters higher than it otherwise would. This gives the log room to drop into place.

Bug: If the player restarts the stage, there is not enough room to spawn a new log above the log trestle. Even worse, the game turns the **old** log on the trestle into a ghost image that cannot be touched, so the player now has no logs available to her.

If a log is instead spawned without a trestle, the old log has enough time to become "real" as falls. However, the old log(s) remain as ghosts that look stupid to the player. Therefore, I recommend against spawning logs as part of a stage.

Cargo Models

Instead of spawning cargo, you can place it on the map like any other model (page 84).

Bug: A cargo model isn't labeled on the navigation map, and it cannot be delivered. If it is lost, it cannot be respawned.

Because of the above bug, cargo models are poorly suited for cargo delivery goals. If you choose to place a cargo model on your map (e.g. as scenery), consider making it static with Freeze Physics (page 76) to eliminate any potential confusion for the player.

Cargo Sources on Map

Cargo and cargo loading zones are highlighted in various ways on the navigation map and in the objective details, depending on the tracked objective. Understanding this highlighting can help you decide how simple or complex you want to make the cargo situation on your map.

However, before we can talk about highlighting, let's review the normal visibility of cargo on the map:

- 'Recognized' cargo has a 3-D shape on the map with an icon and label. If unpacked cargo is in a revealed area of the map, it is recognized.

- ‘Unrecognized’ cargo has a 3-D shape on the map, but has no icon or label. Unpacked cargo that is in the gray, unrevealed part of the map is unrecognized. A cargo model (described in the section above) is also unrecognized.

- ‘Invisible’ cargo cannot be seen on the map at all. Cargo that is in a cloaked part of the map is invisible. Cargo that is packed on a truck or trailer is also invisbie.

A cargo loading zone never has a 3-D shape on the map, but it can be ‘recognized’ with an icon and label if it is revealed and Is Visible on Minimap is not False. Otherwise it is invisible.

OK, now we can discuss highlighting of cargo and cargo loading zones (cargo sources):

- A ‘highlighted’ cargo source is marked with a yellow ‘load’ icon on the navigation map when its objective is tracked or when the objective details are viewed.

- A ‘recommended’ cargo source is always highlighted as above. In addition, when the objective details are viewed, each recommended cargo source is listed under the delivery goal description to the left of the map. Selecting a recommended cargo source centers the navigation map on that source.

- An ‘unhighlighted’ cargo source is not marked with a yellow ‘load’ icon on the navigation map. An unhighlighted cargo source is never recommended.

Cargo sources when spawned cargo types differ from delivery cargo types

If any cargo is spawned that doesn’t need to be delivered in the same stage, then all bets are off. I can’t figure out the pattern, but recommendations and highlighting are definitely unreliable for this case. Don’t do it.

If the spawned cargo types are a subset of the delivery cargo types (e.g. one of the two delivery cargo types is spawned), then recommendations and highlighting mostly work, but when the player selects the unspawned cargo in the objectives list, the navigation map doesn’t center on any cargo source.

Cargo Sources when cargo was spawned and metal detector is not required

If the player views details for the objective while it is being offered and before it is activated, then cargo has not yet spawned, so cargo is highlighted as per page 209.

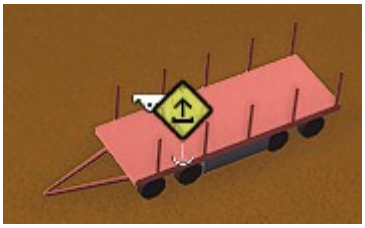
If cargo was spawned by the current stage of the tracked objective, then any unpacked cargo that matches one of the required delivery types is recommended. If there is more than one unpacked cargo of the necessary type(s), then each is listed below the delivery goal description in the objective details to the left of the navigation map. This may be a different number of cargo items than is listed under the tracked objective in the upper right of the screen.

Bug: If the same cargo type is required for more than one delivery goal, then each unpacked cargo of that type is listed once for each goal. E.g. two goals for metal rolls and three unpacked metal rolls on the map esults in six entries for metal rolls in the objective details. This is a bit confusing, but ultimately harmless.

A cargo source can be recommended even if it is invisible.

If a cargo type is required for delivery, but there is no unpacked cargo of that type to recommend, then the cargo type is listed once in the objective details, but selecting it does not center the map on anything. Other cargo sources may be highlighted (as below), but they are never recommended when any cargo was spawned.

Besides the recommended cargo sources above, other cargo sources are also highlighted. A cargo loading zone is highlighted if it includes a cargo type used in the current delivery goal and Is Visible on Minimap is not False for the zone. The cargo loading zone remains highlighted even if it has run out of inventory of the needed cargo type. Cargo packed on a detached trailer is also highlighted.



Bug: Unpacked cargo that was spawned by the current objective stage is highlighted, as is other unpacked cargo of the same type, even if it is not required by a delivery goal. It may also disrupt highlighting of unpacked cargo that **is** required, although at this point it gets so confusing that I'm not sure anymore.

That leaves cargo that isn't highlighted:

- Unpacked cargo that isn't a type required by a delivery goal and doesn't match the type of spawned cargo.
- Cargo packed on a truck or on an attached trailer.
- Any cargo model.

Tip: The recommended cargo sources make the most sense when exactly the right amount of cargo is spawned as is required for delivery, and there are no other sources for that cargo type.

Cargo Sources when cargo was spawned and metal detector is required

The Should be discovered by metallodetector property has independent effects for each cargo type:

- If Should be discovered by metallodetector is enabled for **any** spawned cargo item of a particular type, then its visibility and highlighting is handled as below. I abbreviate a cargo item that was spawned with Should be discovered by metallodetector enabled as "MD cargo", and its type is an "MD cargo type". There can also be non-MD cargo (spawned or unspawned) in an MD cargo type.
- If Should be discovered by metallodetector is disabled for **all** spawned cargo item of a particular type, then its visibility and highlighting is handled as in the section above.

When an objective stage begins, all unpacked cargo of an MD cargo type is hidden on the navigation map. This applies to all spawned and unspawned cargo of that type, including spawned cargo for which Should be discovered by metallodetector is disabled.

Hidden cargo is not just invisible on the map; it also is not highlighted with a yellow icon. Although it is still recommended and thus listed in the objective details, selecting one of the recommendations does not center the map on its location. The cargo is still visible and manipulable in the driving view as usual.

Bug: If the player completes the stage without using the metal detector to detect any MD cargo, then that cargo type remains hidden afterward. It can only be exposed by a later metal detector stage/objective with the same cargo type (which the player could again complete without using the metal detector).

Tip: As with regular spawned cargo, MD cargo makes the most sense when there are no other sources for that cargo type. In addition, that cargo type should not become available in the future, either, or it may end up hidden.

When the player is tracking the relevant objective and uses a truck's metal detector within about 50 meters of MD cargo, the metal detector beeps three times, and the MD cargo is detected. The metal detector does nothing if the objective is not being tracked or if it detects only non-MD cargo.

BTW, it can be difficult to decipher how to use the metal detector. When the metal detector is installed, you can operate it from the functions menu on the left. A useless confirmation then asks if you want to scan or not. The metal detector folds out and makes some noises. If it folds back up without further ceremony, your truck isn't close enough to the cargo. If it beeps three times and cargo icons show up in the driving view, you found something.

If the player has detected **any** MD cargo of a cargo type but has **not yet** detected **all** MD cargo of that type, then all cargo of that type changes visibility in the following ways:

- All unpacked cargo of that type is no longer hidden, but is now recognized (i.e. drawn on the navigation map, but not labeled).
- A yellow icon appears in the driving view pointing to each unpacked cargo item of that type. But no yellow icons appear on the map for those cargo items.
- Selecting a cargo source from the recommended list in the objective details centers the map on it (even thought it isn't highlighted with a yellow icon on the map).

If the player has detected **all** MD cargo of a cargo type, then all cargo of that type becomes fully recognized, highlighted, and recommended.

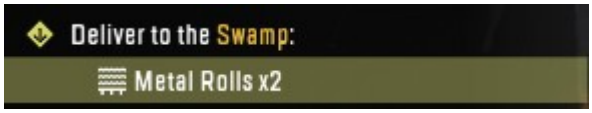
If the player completes the objective stage after detecting at least some MD cargo, but not all of it, then all cargo of that type also becomes fully recognized after the stage is complete.

If two stages are simultaneously active with the same MD cargo type, then any change to visibility or highlighting by one objective affects all cargo of that type as described above. You probably want to avoid doing this.

Should be discovered by metallodetector has no effect on the visibility or highlighting of cargo loading zones and packed cargo on unattached trailers.

Cargo Sources when cargo was not spawned

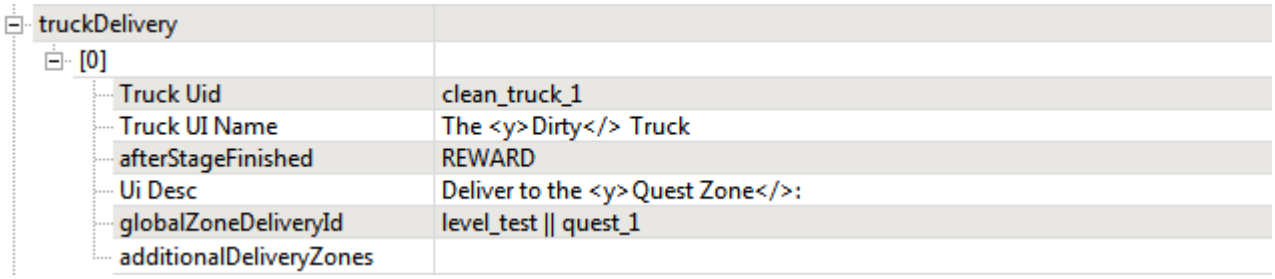
If no cargo was spawned by the current stage of the tracked objective, then only cargo loading zones and detached trailers with packed cargo are highlighted. The cargo details lists the quantity of each cargo type required for delivery on a single line. Selecting that line centers the map on a random highlighted cargo loading zone. The player has to zoom out and pan the map around to find any other highlighted cargo loading zones or detached trailers.



Truck Delivery Goal

A stage can require the player to deliver one or more particular trucks (and/or trailers) to specified zones.

To create a truck delivery goal, click the green + to the right of truckDelivery. This adds one truck delivery goal.



Truck ID

The truck Uid should specify a truck's unique ID (page 67). When a truck's ID is used in a truck delivery goal, that truck is reserved (page 69), and the player cannot drive it unless and until the objective awards it to the player. Since a player cannot drive the specified truck, the player can only complete the truck delivery goal by towing it with another.

If the player restarts an objective with a truck delivery goal, the truck respawns at its original location. If another vehicle is in the way, the truck remains "virtual" until space is cleared for it.

Bug: If the truck delivery goal is for a DLC truck that the player doesn't own, then the truck won't initially appear on the map. But if the player restarts the objective with the delivery goal, it **will** spawn at its designated location.

A particular truck ID can be reserved only by a single objective, although it can be reserved by multiple goals in one or more stages of that objective.

If the truck Uid doesn't match a truck on the map, the game will throw a ModMapError (page TBD).

Truck Name

The Truck UI Name provides a name for the truck to use on the navigation map, in tracking information, and in the objective details. It supports formatting tags. UI text and formatting tags are described on page 146.

If the Truck UI Name is not specified, the game refers to the truck by its type instead.

If a truck ID is used by multiple goals and/or multiple stages in an objective, it should have the same name for all of them. If different names are used, the objective details and tracking information may list the different names, but the truck gets only one of the names, which never changes.

Goal Description

The UI Desc property describes the goal for the player. UI text and formatting tags are described on page 146.

Delivery Zone

The globalZoneDeliveryId property specifies the global zone ID for the primary delivery zone.

Bug: If any IDs are specified in additionalDeliveryZones, then the goal cannot be completed (even if the truck is delivered to all zones simultaneously).

If multiple truck delivery goals in a stage have the same delivery zone, the game combines them all under a single goal description (UI Desc). In this case, you should use the same goal description for all such goals.

Stage Complete

The afterStageFinished property has a dropdown menu of options of what to do with the delivered truck when the stage is complete:

- DO_NOTHING: Nothing happens at the end of the stage.
- DETACH: The truck is detached from the winch at the end of the stage (if not already detached).
- DESTROY: The truck is destroyed at the end of the stage.
- REWARD: The truck is awarded to the player. Unlike the above actions, the truck is awarded as soon as the goal is complete, even if the stage is not yet complete. Although the truck is rewarded at the end of the goal, the player is not told about the award until the end of the objective. Once awarded, the truck is no longer reserved, and the player can drive it. The truck retains its cool Truck UI Name when selected on the navigation map, although it is simply listed by its type in the list of trucks on the left, and the name is lost the next time the player takes the truck to a garage.

If a truck is awarded when there are goals remaining that reserve the same truck, the award is effectively delayed until there are no more such goals left in the objective.

CAUTION: If the player changes into the rewarded truck while the objective is still active and then restarts the objective, **the player is left without a truck and is completely helpless**. The truck should be awarded only in a stage with no other goals, and that stage should be the last one in the objective (so that there is no opportunity to restart the objective), or that stage should be followed only by a stage with a single change truck goal (page 213) (so that the objective ends as soon as the player enters the truck).

If multiple goals reserve the same truck in a stage, only one of the afterStageFinished properties is applied to the truck at the end of a stage. You should set the afterStageFinished property the same for all such goals.

If an objective rewards multiple trucks, only one of the trucks is listed as the objective reward, but all of the trucks are awarded as their goals are completed.

There is never an experience bonus for “discovering” a reserved truck, even when that truck is awarded to the player.

Trailer Delivery

A truck delivery goal can equally be applied to a trailer by specifying the “truck ID” of a trailer. The player can complete the goal by towing the trailer by a winch as with a truck, or the player can instead attach the trailer to her truck in the normal manner. In fact, the player can use the trailer as much as she wants, transporting cargo, etc. The only thing she can’t do with it is sell it.

Bug: The player still cannot sell the goal trailer even after it is given to her as a reward.

Truck Repair Goal

A stage can require the player to fully repair **and refuel** one or more particular trucks. A truck repair goal can be combined in the same stage as a truck delivery goal so that the player must repair, refuel, and deliver the specified truck.

To create a truck repair goal, click the green + to the right of repairTruck. This adds one truck repair goal.

Truck ID

The truck Uid should specify a truck’s unique ID (page 67). When a truck’s ID is used in a truck repair goal, that truck is reserved (page 69), and the player cannot drive it unless and until the objective awards it to the player. Since a player cannot drive the specified truck, the player can only complete the truck repair goal by repairing and/or refueling it from another truck. If the specified truck is already fully repaired and refueled when the objective stage starts, the player must still perform a “repair” action on it to complete the goal. (“Refuel” is disabled if the target truck is fully refueled, but “repair” is still possible when it is fully repaired.)

If the player restarts an objective with a truck repair goal, the truck respawns at its original location. If another vehicle is in the way, the truck remains “virtual” until space is cleared for it.

A particular truck ID can be reserved only by a single objective, although it can be reserved by multiple goals in one or more stages of that objective.

If the truck Uid doesn't match a truck on the map, the game will throw a ModMapError (page TBD).

If the truck Uid is set to a trailer, then the goal is impossible to complete because a trailer can never be repaired or refueled.

Truck Name

The Truck UI Name provides a name for the truck to use on the navigation map. It supports formatting tags. UI text and formatting tags are described on page 146. Unlike for a truck delivery goal, the truck's name is never displayed by the objective. It is only shown when the player clicks on the truck in the navigation map.

If the Truck UI Name is not specified, the game refers to the truck by its type instead.

If a truck is reserved by multiple goals and/or multiple stages in an objective, it should have the same name for all of them. If different names are used, the objective details and tracking information will list the different names, but the truck gets only one of the names, which never changes.

Goal Description

The UI Desc property describes the goal for the player. UI text and formatting tags are described on page 146. Unlike for a truck delivery goal, the goal description is not followed by the truck name, so you might want to provide your own description of the target truck in the goal's UI Desc.

Stage Complete

The afterStageFinished property is exactly the same as described for the truck delivery goal (page 211).

Change Truck Goal

A stage can require the player to change into a specific truck.

To create a change truck goal, click the black arrow to the right of changeTruck. A stage can have only one change truck goal.

The truck Uid should specify a truck's unique ID (page 67). The goal is completed as soon as the player changes to the designated truck. The designated truck should not be reserved, or the player will not be able to enter the truck and complete the goal.

If the designated truck is reserved (page 69) by a different objective, then the game will throw a ModMapError (page TBD). The truck to change into can be reserved by another goal in the **same** objective, as long as it is rewarded to the player before the change truck goal's stage.

If the truck Uid is set to a trailer, then the goal is impossible to complete because the player can never change to driving a trailer.

If the player restarts an objective with a change truck goal, the truck is not respawned.

Bug: If the player ever visits a garage with the truck needed for the change truck goal, its truck ID is lost. I.e. when the player leaves the garage, the game doesn't restore the original truck, but instead spawns a new truck of the same type.

Bug: If the player is already in the designated truck when the goal is activated, the goal is **not** completed. The player must change to a different truck and then change back to complete the goal.

Because of the above bugs, and because the player is not required to **do** anything in the designated truck, the change truck goal really only makes sense as a way to add flavor in a final stage after awarding a truck to the player from a truck delivery goal (page 209) or truck repair goal (page 212). In this case, the truck is guaranteed to have never visited a garage, and the player is guaranteed to not be in the truck when the goal is activated.

Bug: The player cannot unlock a truck that is specified by a change truck goal. The truck must have its Locked property (page 68) set to False to allow the player to enter the truck and complete the goal. This bug applies whether the truck was reserved and rewarded or was never reserved at all.

The Truck UI Name property is exactly the same as described for the truck delivery goal (page 210) and **not** as described for the truck repair goal (page 213).

Bug: The afterStageFinished property is presented in the Zone Settings Editor, but it has no function for a change truck goal.

The Ui Desc property is exactly the same as described for the truck delivery goal (page 210).

Visit Goal

A stage can require the player to visit one or more zones.

To create a visit goal, click the black arrow to the right of visitAllZones. This pops up a useless dialog. Click OK to confirm it.

The visitAllZones category now has a zones subcategory. Click the green + to the right of zones to add one or more goals with associated destination zones.

The UI Desc property describes the goal for the player. It supports formatting tags. UI text and formatting tags are described on page 146.

The globalZoneId property specifies the global zone ID for the destination zone.

The Truck need to visit Uid property is optional. If filled, it should refer to a truck's unique ID (page 67), and the truck should not be reserved (page 69). The player is given credit for visiting the destination zone only in she is driving this truck. See the identical property for cargo delivery (page 201) for the long list of cautions and caveats.

Bug: If the Truck need to visit Uid refers to a trailer, the visit goal cannot be completed, even if the designated trailer is attached to the player's truck.

Seismic Vibrator Goal

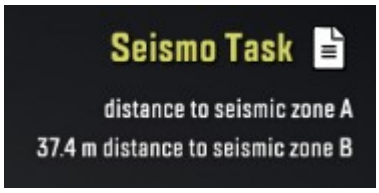
A stage can require the player to use a seismic vibrator module in one or more zones.

To create a seismic vibrator goal, click the black arrow to the right of Seismograph Settings. This pops up a useless dialog. Click OK to confirm it.

The properties of a seismic vibrator goal are identical to those for a visit goal, described above. But unlike a visit goal, the target zones are not made visible in either the driving view or the map. Instead, the distance to each zone is listed when the objective is tracked, and the player must find it by blind triangulation.

Tip: Don't make the seismic vibrator target zone too small, or the player may get very frustrated trying to maneuver her truck to find it. The maps from Saber appear to set the zone to 25 – 30 meters in diameter, which seems to work well. Putting the zone in an obvious clear area with congested areas around it also helps direct the player to the correct area.

When the player's truck is within a target zone, the goal description is listed in the objective tracking information without embellishment. But when the player's truck is outside a zone, the distance to the target zone is awkwardly prepended to the goal description. I haven't yet found a good way to phrase the goal description to deal with both possibilities.



BTW, it can be difficult to decipher how to use the seismic vibrator. When the seismic vibrator is installed, the objective tracking information in the upper right shows the location to each destination zone. If you are not in a destination zone, there is no way to operate the seimic vibrator. Once you reach a zone, open the functions menu and press the key indicated at the **bottom** of the screen (not with the other functions on the left).

Living Area Goal

TBD

Multiple Goals

Most goal types allow multiple goals of that type to be specified in a stage. In addition, a stage may have goals of multiple types. To complete the stage, the player must complete all goals of all types in any order.

As goals are completed, the game removes them from the objective tracking info. Note that this does not affect the objective details, which always lists all goals, complete or incomplete.

Zone Visibility

Understanding zone visibility can be helpful when debugging your zones. E.g. is a zone not visible because it isn't working, or is that normal? It is also useful to understand which zones can be easily found on the map vs. which ones require more searching.

Some zones change status based on how close the player brings a truck. In this case, the distance is always measured from the center of the truck to the center of the zone.

Driving View

If a zone is "available" for activity, its zone perimeter is generally visible in the driving view (except where noted). Otherwise, it is "unavailable", and it does not show up in the driving view.

If a zone is available in the driving view for multiple purposes with different colors of zone perimeter, then the pattern seems to be that blue beats yellow and red, or yellow beats red.

If a zone is available, the zone's icon is also typically displayed in the driving view above the zone's center, but only if the player's **camera** is closer than about 50 meters and the truck is **not** in the zone.

If a zone can display a white icon (based on its type) or a yellow icon (based on an objective), priority goes to a yellow icon. A yellow icon is based on the objetive details. A white icon is (usually) specified by the zone locator properties. If multiple icons of the same color are possible, the game picks one. I haven't tried to figure out the pattern.

> **Bug:** The game occasionally gets confused and stop's displaying a zone's icon in the driving view. This condition may continue until the map is reloaded.

Is Visible On Minimap has no effect on the driving view.

Navigation Map

An available zone can also be "visible" (with an icon and label) on the navigation map and in the list of zones on the left side of the navigation map. An unavailable zone is generally not visible on the navigation map, except where noted. Some zones become visible on the navigation map when they are available and in a revealed area of the map, but other zones differ. If a zone is not visible, it is "invisible".

If a zone is visible, it is usually listed in white in the zone list to the left of the navigation map. In certain cases, it is listed in gray, although that doesn't necessarily mean anything interesting.

A zone is marked on the navigation map with the highest visibility among all of its purposes. If multiple highlighting icons could be applied, the game picks one.

All of these conditions are determined separately based on the zone type and any objectives it is part of as described in the sections below. Regardless of anything else said in these sections, if Is Visible On Minimap is False for a zone, it is never visible on the navigation map.

Visibility Based on Zone Type

A garage entrance is always available in the driving view with a yellow perimeter. It is invisible on the map until revealed. Even when revealed, it is still listed in gray until the player “discovers” it by driving within 20 meters. The player can only use “recover to garage” after discovering the entrance. Observation from a watchtower does not count toward discovering a garage.

If the garage entrance zone is large enough, the player may be able to enter the garage before it is even revealed. In this case, it counts as discovered (so the player can recover to it), but it still isn’t revealed on the map until the player drives closer.

A garage exit is always available in the driving view with a red perimieter. It is never visible on the map.

A recovery zone is never available in the driving view or visible on the map. If a recovery zone is made visible in the driving view because the zone is shared with another function, the zone perimeter may have an exotic color. (Pink?)

A trailer store, fuel station, or automatic cargo loading zone is always available in the driving view with a yellow perimeter. It is invisible on the map until revealed.

A manual cargo loading zone is available only when it is linked to an automatic cargo loading zone. If it is, it is visible in the driving view with a red perimeter. Its icon on the driving map is a loading hook (ignoring whatever icon was specified for the zone). Its icon can be overridden by a yellow icon for an objective, however. A manual cago loading zone is never made visible on the map.

A service hub is always available in the driving view with a yellow perimeter, and it is always visible on the map. It is listed in gray until it is in a revealed area of the map, but that makes no difference to anything else.

A watchtower is initially available in the driving view with a blue perimeter, and it is always visible on the map. Unlike other zones, the map usually shows only the watchtower’s icon. Its name is shown only when it is selected on the map. The watchtower is listed in gray until it is in a revealed area of the map, but that makes no difference to anything else. When a watchtower is visited, it is removed from the driving view and the navigation map even though the “launch observation” function is still available in the region.

An upgrade zone is initially available in the driving view with a yellow perimeter. However, it only becomes visible on the map when the player drives within 10 meters of it, or when the upgrade zone is revealed by a watchtower.

Crafting zone, etc. TBD.

Visibility Based on Objective and Goals

The offer zone for an unlocked **task or contest** is available in the driving view with a yellow perimeter. It is "discovered" when the player drives within 10 meters of it, or when the zone is revealed by a watchtower. When a task or contest is discovered, its offer zone is listed under Task Giver to the left of the navigation map.

From the time it is discovered, a **contest** is listed under Contests to the left of the navigation map. Its details can be examine even if its offer zone has never been visited. It remains listed there until the contest is accepted and completed. Even after completion, however, its offer zone remains visible and listed under Task Giver, and the player can visit it again to retry the contest.

A **task** is listed under Tasks to the left of the navigation map only once the player has visited its offer zone and accepted the task. The offer zone remains available and visible under Task Giver until the task is completed, at which time it is removed. Likewise, the task itself is listed under Tasks until the task is completed.

A **contract** does not have an offer zone, but it may have other zones which are visible as described below.

When an objective is active, is being tracked, or the player is viewing its details, its goals make their zones visible as described below. If the player views an objective's details before accepting it, it makes zones visible and highlighted as if its first stage were active.

If a cargo delivery goal or truck delivery goal is in the current stage of any active objective, its delivery zones and interaction zones for manual cargo delivery are available in the driving view with a yellow perimeter. If the objective is being tracked or its details are viewed, then its delivery and interaction zones in the current stage are visible on the navigation map and are highlighted with an "unload" icon in a yellow diamond. Cargo sources are visible as described on page 204.

The interaction zones of a manual cargo delivery goal in the current stage is also available in the driving view with a yellow perimeter. If the objective is being tracked or its details are viewed, then the interaction zones are the current stage are visible on the navigation map and are highlighted with an "unload" icon in a yellow diamond.

If a visit goal is in the current stage of any active objective, its destination zones are available in the driving view with a blue perimeter. If the objective is being tracked or its details are viewed, then its destination zones in the current stage are visible on the navigation map and are highlighted with a magnifying glass in a yellow diamond.

A seismic vibrator goal does not make its target zones visible in either the driving view or the navigation map. However, if a zone is already visible in the driving view for some other purpose, its zone perimeter changes to blue. In either case, selecting a zone from the objective details list does not center the navigation map over it or otherwise give a clue to its location.

More goal types TBD.

When the objective details are viewed, zones used in later stages are sometimes listed. This listing does not make those zones visible on the map. Selecting a zone from a future stage centers the map on where it would be if visible.

Playing Your Map

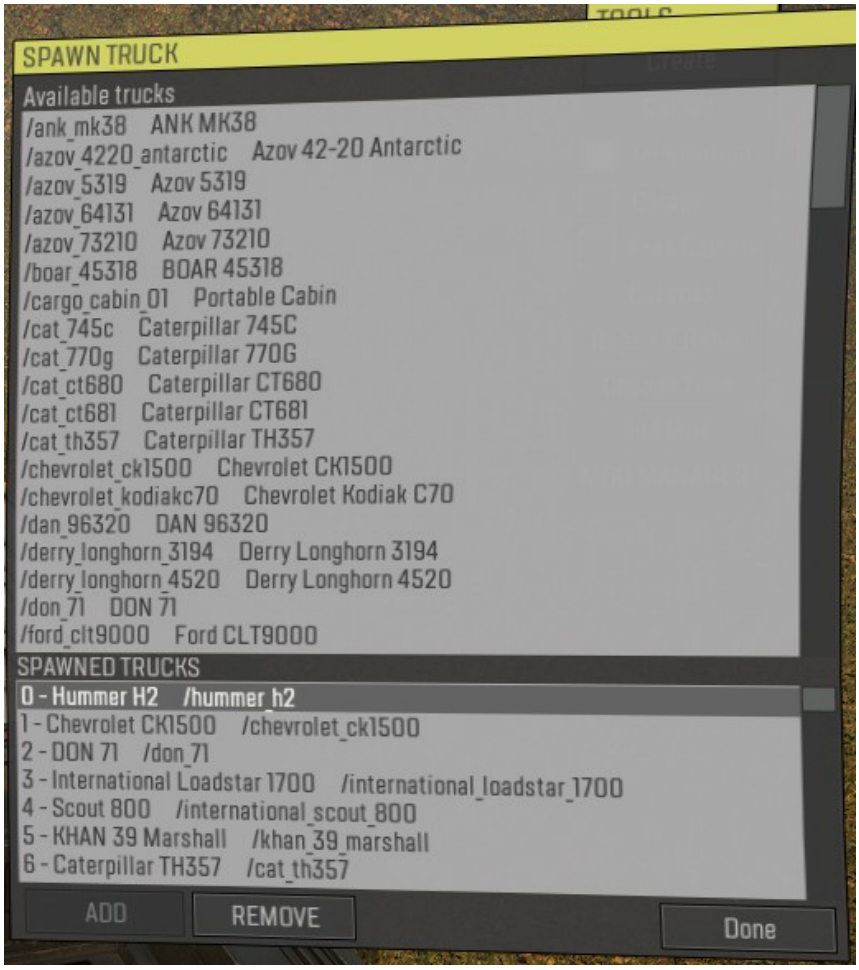
Dev Tools

While running the game in a proving grounds map or a local custom map, a Tools menu is shown. This menu of development tools (abbreviated as “Dev Tools”) gives you the ability to quickly try different combinations of trucks, trailers, and add-ons.

The various options in the Tools menu are described below.

Dev Tools: Create

Clicking Create pops up a dialog that allows you to create a truck and/or warp to any existing truck.



To add a truck, click on any truck under Available trucks, then click somewhere on the visible terrain. A ghost image of the selected truck will appear. Rotate the truck with left click and drag. When you are satisfied with its location, click ADD.

Warning: the created truck may intersect hazards on the map such as models or other trucks. As soon as you get close enough to enable its physics, the truck may take (or cause) significant damage.

To remove the truck currently being driven, click REMOVE.

Warning: if you close the Create dialog while not in control of a truck, you won't be able to drive any truck or even move the camera. However, you can still use the Tools menu to jump into another truck.

To jump into another truck from the Create dialog, click any truck listed under SPAWNED TRUCKS.

Bug: While stranded outside of a truck, the map won't list any trucks, so you can't jump to a truck that way.

Dev Tools: Reload

The Reload option is not particularly useful to map development. It assists truck modders by reloading truck assets without leaving the map. It is not possible to change a packed map while it is being used by the game.

Dev Tools: Information

The Information checkbox puts some velocity information in the center of the HUD. Again, this is useful only to truck modders.

Dev Tools: Virtual Garage

The Garage checkbox pops up additional menus that allow you to customize the current truck with add-ons and trailers. This virtual garage allows any compatible part to be installed regardless of whether you unlocked the part or reached the necessary rank.

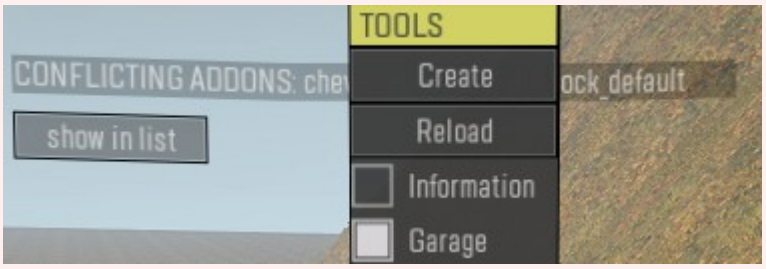
Bug: You can open the map while in the virtual garage, but it will not give you the usual options to view objectives or change trucks. Close Garage mode and re-open the map to get the usual map options.

All parts in the virtual garage are listed by the lowercase name used by the Editor. Thus, the Garage option is extremely useful for getting the names of compatible parts to add to trucks in the Editor.

To install or uninstall an item, double-click it.

If there is an installation conflict, a message will appear on the screen along with a button, show in list. Clicking show in list moves the part selection to the conflicting part (or one of the conflicting parts if there are multiple). The conflicting part may need to be uninstalled (to make room) or installed (to provide an attachment point, such as for a semi-trailer). You can then double-click the conflicting part to install/uninstall it before re-attempting your original change.

Bug: The conflict message is obscured by the Tools menu. Therefore, you'll probably need to click the show in list button to find the (first) conflicting item.



Bug: The virtual garage doesn't always detect when a combination of add-ons conflicts with an existing trailer. But it does detect when a new trailer conflicts with existing add-ons. So always add the trailer last.

If your truck is too close to another object, the installation may instead fail because there isn't enough room to install it without causing a collision. In this case, the part will not be installed. This can be distinguished from a conflict with another part by the lack of a conflict message.

Tires, Rims, and Suspension

Tires and rims must match. To make this easier, installing any tire will automatically install a compatible rim if necessary. This helping hand only goes one way, though. If you attempt to install a rim that is incompatible with the current tire, the install silently fails.

The tire and rim size must also be compatible with the suspension. If you attempt to install a tire size that is incompatible with the suspension, the install silently fails.

Bug: After installing a raised suspension and a large tire and rim, the virtual garage will allow you to install a shorter suspension while keeping the oversized tire and rim. Do not attempt to use this incompatible combination in the map Editor.

Dev Tools: Free Camera

Clicking the Free Camera checkbox separates the camera from the truck. There is no immediate change to the camera view, but you can now drive away from the camera. Or you can move the camera freely using the following controls:

- right click and drag: rotate the camera up, down, left, or right.
- I or NumPad 8: move the camera forward.
- K or NumPad 2: move the camera backward.

- J or NumPad 4: move the camera left.
- L or NumPad 6: move the camera right.
- O, NumPad 3, or Shift: move the camera up.
- U, NumPad 1, or Z: move the camera down.

The free camera is useful for looking at an angle not normally allowed, for getting a closer look at something, or for arranging an artistic screenshot. More tips for screenshots are on page TBD.

Dev Tools: Cargoes

Click Cargoes in the Tools menu adds some information to the upper left showing the usage of cargo points on the truck.

Unfortunately, there is no way to artificially spawn cargo using the Tools menu.

Dev Tools: Repair & Refuel

Click Repair & Refuel to fully repair and refuel your current truck. The parts and fuel are not taken from any truck; they are gifted to you by the Tools menu.

Dev Tools: Change Time

Click Change Time to cycle through the lighting conditions at different times of day. This is useful for checking how your map looks when streetlights are on or off, at dusk, etc.

Dev Tools: Mod Tools

The Add Mod and MOD MANAGER buttons allow modded trucks to be managed and spawned.

Revealing the Map

The map is initially an unexplored land. As the player drives around and visits watchpoints, more details are revealed in the map view, and vehicles and zones are discovered. By default, the map view is accessed by pressing the M key in the game.

To fully reveal an area, the player must drive a truck near it. Areas within 39 meters of the active truck are drawn in color in the map view. The player can also visit a watchpoint, which reveals the map within the radius set by the watchpoint's property. Vehicles and zones are usually labeled on the map if they are within a fully revealed area.

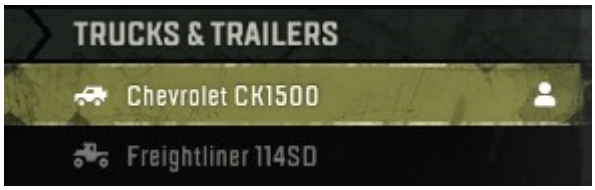
Viewing an objective's details can also label unrevealed vehicles and zones used by the objective. Zone visibility is described in more detail on page 218. If an objective is being tracked or the objective's details are viewed, and a vehicle is needed for the current stage, that vehicle is always labeled on the map, even if it hasn't yet been revealed.

If the map is set to be uncloaked (page 166), then the portions of the map that have not been fully revealed are partially revealed. These areas of the map are drawn in gray shades. The shapes of vehicles can be seen in these partially revealed areas, but they are not usually labeled on the map. Zones are also not usually labeled on the map if they are only partially revealed.

If the map is set to be cloaked, then most of the map is initially black. As the player drives around, areas of the map within about 140 meters of the active truck are partially revealed. Visiting a watchtower fully reveals some portion of the map, but it does not partially reveal any additional area.

Discovery of Vehicles

If a truck or trailer is set to be locked (page 68), then even after it is revealed, its label is initially gray in the list of TRUCKS & TRAILERS on the left side of the map.



A player can "discover" the vehicle by driving within 15 meters of it (as measured between the truck centers). This discovery rewards the player some experience points, and its label changes to white on the left side of the map. If a locked truck has not yet been discovered, the player can't jump into it, even if the target truck is clearly visible. Once the player discovers the truck, she can jump into it by using either the map view or the CHANGE TRUCK function.

If a truck or trailer is set to be unlocked, then its label is white on the left side of the map as soon as it is revealed. There is no separate “discovery”, and nothing new happens when the player drives near it. As soon as an unlocked truck is revealed, the player can jump into it from the map view. Alternatively, the player can use the CHANGE TRUCK function to jump into an unlocked truck. In this case, the target truck does not necessarily need to be revealed as long as it is currently within about 50 meters from the camera. (I.e. moving the camera affects whether you can jump into a truck.)

Once a locked truck or trailer is discovered, it can be purchased in the garage or trailer store even if its rank requirement hasn’t been met. On the other hand, an unlocked vehicle can never be discovered, so it isn’t offered for sale until its rank requirement is met, even if the player has driven it.

Take a Screenshot

To take a screenshot, simply press the PrtScn key on your keyboard. This makes a copy of the framebuffer than you can then paste into an image editor of your choice, such as IrfanView. Alternatively, if you’re running SnowRunner in a window, press Alt+PrtScn to copy only the currently selected window.

If you’re running the Steam version, you can also press Steam’s screenshot key, which by default is F12. In this case, the screenshot is saved to a folder managed by Steam, and you can review and manipulate it from the Steam library.

If you’re trying to set up a pleasing view for the screenshot, a few options are available.

To hide the main HUD and certain other HUD features, open the game settings and turn off the various options under HUD. Or you can leave the HUD enabled and then use an image editor to crop the screenshot to a central part of the screen where the main HUD elements aren’t in the way. Potentially you can adjust the Third-Person

Camera FOV under Video settings so that the central part of the screen shows approximately the field of view that you want.

The Free Camera option in the Dev Tools is useful for getting a different angle.

Bug: Hiding the HUD disables the free camera. So you can't get precisely the angle you want **and** get a pleasing full-screen view.

Alternatively, you can open the game settings, and under Video, turn on the Legacy Camera. With the legacy camera enabled, click the left mouse button to enter camera mode, then push the mouse up to cause the camera to move over the truck so that only the roof of the truck at the bottom of the screen. In this mode, you can hide the HUD and get a good view across most of the window. If you want to cut out the truck, you only need cut off a small portion of the screen shot.

Appendix A: Reference Information

The following sections contain various lists and tables that are handy for reference but would have cluttered up the main text. Once you’re familiar with the editor, you may find yourself using this appendix the most often.

Truck Reference

The table below provides overview information for all trucks so that you don't have to try them all individually. To get detailed information about a particular truck, use the dev tools to create the truck and view it in the virtual garage to discover all allowed add-ons for the truck.

Another source of vehicle information is the SpinTires fandom wiki.

Table Column Meanings

Truck is the name of the truck in the editor and the dev tools. The name of the truck in the game is similar. (In fact, the truck names are virtually identical across all languages, even Russian.)

Reg. indicates the region that the truck is associated with.

- US is the United States/North America.
- Rus. is the Russian Federation.

AWD indicates the truck's all-wheel-drive capability.

- no: AWD cannot be enabled
- upgrade: AWD can be installed after the appropriate upgrade is obtained
- yes: AWD can be enabled by the player
- always: AWD is always enabled

Diff Lock indicates the truck's diff-lock capability.

- no: diff lock cannot be enabled
- upgrade: a locking differential can be installed after the appropriate upgrade is obtained
- yes: diff lock can be enabled without an upgrade, but only in a low gear
- yes+no: yes by default, but can be removed with an "upgrade"
- always: the differential is always locked

Tires indicate what types of tires the truck can use.

- H: highway
- A: all-terrain
- O: offroad
- M: mud
- C: chains

Trailer indicates what kinds of trailers the truck can pull.

- S: scout trailers
- T: regular trailers
- L: low-saddle semi-trailers
- H: high-saddle semi-trailers
- *: excludes semitrailer_oiltank and semitrailer_heavy_oiltank
- †: excludes semitrailer_m747
- ‡: excludes semitrailer_stepdeck_5 and semitrailer_gooseneck_4
- R: special rock semi-trailer

Bed indicates what kind of non-log cargo the truck can carry in its bed (or on the roof).

- B: big crane
- R: repair parts; if followed by a *, can be combined with a saddle
- r: a small number of repair parts; sometimes can be combined other bed items. Spare wheels are not covered in this reference.
- F: fuel tank
- f: a small amount of fuel; sometimes can be combined other bed items
- M: maintenance parts (repair + fuel). Blocks use of a trailer.
- m: a small amount of maintenance parts (repair + fuel); sometimes can be combined other bed items
- D: metal detector. Can be combined with some bed items on some trucks, but that looks like be a bug.
- V: seismic vibrator. Blocks use of a trailer.
- C: mini-crane. If followed by a '+', can be combined with any following options (usually only cargo units). The combination may prevent attachment of trailers. The mini-crane itself sometimes conflicts with semi-trailer saddles or specific fuel semi-trailers.
- 1–4: 1 to 4 generic cargo units
- G: can carry a cargo container, but not other cargo units

Logs indicates what kind of logs the truck can transport.

- C+: log crane; can be combined with short, medium, or long logs if available; conflicts with other bed items
- C/: log crane; can be combined with medium log trailer if available, but not S or L or other bed items
- S: short logs in the bed; can be combined with medium log trailer if available
- M: medium logs in a trailer
- L: long legs stretching from the bed to a trailer

DLC indicates the DLC that must be owned to use the truck.

- S1: Season 1: Search & Recover
- S2: Season 2: Explore & Expand
- S3: Season 3: Locate & Deliver
- S4: Season 4: New Frontiers
- AD is the Anniversary DLC (free)
- TD* is the Tatra Dual Pack
- CL* is the Classico Pack
- GB* is the GMC Brigadier DLC
- N5* is the Navistar 5000-MV DLC (limited availability)
- WS* is the Western Star 49X DLC
- * indicates that I don't own these DLCs, so my data might not be perfect.

Some truck add-ons were also released as part of a DLC package, but you don't need to own the DLC to get the add-ons. DLC is required only for the truck itself.

Scout Trucks

Reg.	Truck	AWD	Diff Lock	Tires	Trailer	Bed	Logs	DLC
US	cat_th357	always	always	M		C		S2
	chevrolet_ck1500	yes	upgrade	HAOMC	S	rfm		
	chevy_apache	yes	always	AOMC		m		CL*
	ford_f750	upgrade	always	HOC	T	C+1	M	S1
	hummer_h2	yes	upgrade	HAOMC	S	mm		
	international_loadstar_1700	always	always	HAOC	S	CRm		
	international_scout_800	yes	always	HAOMC	S	m		
Rus.	don_71	yes	yes	HAOMC	S	fm		
	khan_317_sentinel	yes	yes	HAOMC	S	f		S4
	khan_39_marshall	always	yes	M		m		
	khan_lo4f	always	always	HAOMC	S	m		
	tuz_166	always	always	HAOMC	S			
	tuz_420_tatarin	always	always	O				
	yar_87	yes	always	AOMC	S	m		

Highway Trucks

Reg.	Truck	AWD	Diff Lock	Tires	Trailer	Bed	Logs	DLC
US	ford_clt9000	no	no	HAOC	TLH	BRFMDC+2	C/SML	
	gmc_9500	upgrade	upgrade	HAOC	TLH	BRFMDVC+2	C/SML	
	international_transtar_4070a	no	no	HAOC	TLH	BRFDC+2	C/SML	

Heavy Duty Trucks

Reg.	Truck	AWD	Diff Lock	Tires	Trailer	Bed	Logs	DLC
US	cat_ct680	upgrade	yes+no	HAOC	TLH	BRFMDC+2	C/SML	AD
	cat_ct681	upgrade	upgrade	HAOC	TLH	BRFMC+2	C/SML	
	chevrolet_kodiakc70	upgrade	yes	HAOC	TLH	BRFDC+2	C/SML	
	gmc_8000	upgrade	no	HAOC	TLH	BRFMDVC+2	C/SML	GB*
	international_fleetstar_f2070a	upgrade	yes	HAOC	TLH‡	BRFMDC+2	C/SML	
	international_hx_520	upgrade	no	HAOC	TLH	BRFMC+2	C/SML	AD
	ws_4964_white	upgrade	upgrade	HAOC	TLH	BRFMC+2	C/SML	

Offroad Trucks

Reg.	Truck	AWD	Diff Lock	Tires	Trailer	Bed	Logs	DLC
US	ank_mk38	always	always	HAOMC	T	2	M	
	freightliner_114sd	upgrade	yes	HAOC	TLH	BRFMDVC+2	C/SML	
	freightliner_m916a1	yes	always	HAOC	TLH*	C	M	
	international_paystar_5070	yes	yes	HAOMC	TLH	BRFMDVC+2	C/SML	
	royal_bm17	yes	no	HAOC	TLH	BRFMDVC+2	C/SML	
Rus.	azov_5319	always	always	AOMC	TLH	BRFMVC+2	C/SML	
	azov_64131	always	always	AOMC	TLH	BRFMVC+2	C/ML	
	krs_58_bandit	always	always	HAOMC	TLH	FMVC+Rr2	C/SML	S2
	step_310e	yes	yes	HAOC	TLH	BRFMDC+2	C/SML	
	tatra_805	yes	upgrade	HAOMC	T	m2	M	TD
	tayga_6436	always	always	HAOMC	TLH	BRFVC+2	SML	
	tuz_16_actaeon	yes	yes	HAOMC	T	RFC+1	ML	S1
	tuz_108_warthog	yes	yes	HAOMC	TL	RFC1	M	
	voron_ae4380	always	always	HAOMC	TLH	BRFDC+2	SML	
	voron_d53233	always	always	HAOMC	TLH	BRFMDC+2	C/SML	
	zikz_5368	yes	upgrade	HAOMC	TL	RFMD2C+1	C/ML	

Heavy Trucks

Reg.	Truck	AWD	Diff Lock	Tires	Trailer	Bed	Logs	DLC
US	cat_745c	yes	yes	M		FG	SM	
	cat_770g	no	always	M		F	R	S2
	derry_longhorn_3194	always	always	HAOC	TLH		M	
	derry_longhorn_4520	yes	yes	HAOC	TLH		ML	
	navistar_5000mv	always	yes	HAOC	TLH	C	ML	N5*
	pacific_p12w	yes	always	AOMC	TH	BRMD	C/SML	
	pacific_p16	no	always	O	TH†		ML	
	pacific_p512	no	upgrade	O	TLH	BFMDVC+R2	C/SML	S3
	paystar_5600ts	upgrade	upgrade	HAOMC	TL	BR*FC+3	C+SML	
	western_star_49x	upgrade	yes	HAOC	TLH	RFMVC+2	ML	WS*
	ws_6900xd_twin	upgrade	yes	HAOMC		4		
Rus.	azov_4220_antarctic	always	yes	M	T	2	C/ML	
	azov_73210	always	always	AOMC	TLH	BRFMV3C+2	C/SML	
	boar_45318	yes	upgrade	HAOC	TH	RV	ML	S3
	dan_96320	always	always	AOMC	TLH	BRF2	ML	
	kolob_74760	always	always	M	TH		M	
	kolob_74941	yes	yes	M	TH		M	
	tatra_t813	yes	yes	AOMC	TH	m2	M	TD*
	zikz_605r	always	always	M	TH	BRrFV2	C/SM	S4

Bug: The cat_745c_log_bunk add-on can hold either short or medium logs, so for example a CAT 745 can be initialized with cargo_logs_cat_745c_log_bunk (short logs). However, due to a bug, the same short logs cannot be loaded onto the CAT 745 in the game.

Trailer Reference

Table Column Meanings

Trailer is the name of the trailer in the editor and the dev tools.

Obj. indicates whether the trailer is intended only for objectives.

- no: the trailer can be bought and sold at a trailer store
- yes: the trailer cannot be bought or sold at a trailer store

Function indicates what functions and/or cargo capacity a trailer provides.

The player does not need to own any DLC to use all trailers.

Scout Trailers

Obj.	Trailer	Function
no	scout_trailer_flatbed_1	1 cargo unit
	scout_trailer_flatbed_2	2 cargo units
	scout_trailer_oiltank	900 fuel
	scout_trailer_radar	350m radar, 120 fuel

Regular Trailers

no	trailer_addon_maintainer	350 repair parts, 2000 fuel
	trailer_flatbed_2	2 cargo units
	trailer_sideboard_2	2 cargo units
	trailer_flatbed_ramps_4	4 cargo units
	trailer_oiltank	2000 fuel
	trailer_service_2	1500 repair parts
	trailer_generator	generator, 1500 fuel
	trailer_log	medium logs
yes	trailer_flatbed_special_2	none (curtainside trailer)

Low-Saddle Semi-Trailers

no	semitrailer_gooseneck_4	4 cargo units
	semitrailer_flatbed_5	5 cargo units
	semitrailer_sideboard_5	5 cargo units
	semitrailer_stepdeck_5	5 cargo units
	semitrailer_oiltank	3700 fuel
yes	semitrailer_special_w_cat_770	none

High-Saddle Semi-Trailers

no	semitrailer_m747	3 cargo units
	semitrailer_gooseneck_8	8 cargo units
yes	semitrailer_stepdeck_plane_01	5 cargo units
	smitrailer_heavy_oiltank	5000 fuel
	semitrailer_coiled_tubing	none
	semiltrailer_heavy_construction_equipment	none
	semitrailer_oil_rig	none
	semitrailer_oil_refinery	none
	semitrailer_for_rocket	none
	semitrailer_rocket	none

Other Trailers

no	trailer_log_pole	requires bunk_log_addon
yes	semitrailer_cat770g	can only be attached to cat_770g
	train	cannot be attached
	trailer_train_rocket	cannot be attached
	cargo_cabin_01	cannot be attached

Texture Layer Reference

Below is reference information for the various texture layers available in the Editor. Since the texture layer dialog box shows a preview of the ground texture, I won't cover that. But many texture layers share the same ground texture while having different 3-D models and/or different properties. I've highlighed the important differences below.

The recommended range for the Tiling scale property is derived from the values used in Saber's reference maps. A few maps use values outside of the ranges I've listed; feel free to experiment and use a value that looks natural in your setting.

Some layers have weird suffixes like _test and _out that look like they were released by mistake. But most of these are used extensively in Saber's reference maps, so it's more likely that these materials simply got used in too many places and so weren't worth the trouble of renaming. I think you can feel comfortable using them in your own maps.

TBD: review special properties in XML files

Grass

Texture Layer	Tiling scale	Notes
default	4 – 7	the default 3-D grass models
grass_out	4 – 7	clumpier 3-D grass models
grass_test	4 – 7	thick, tall 3-D grass, and a few flowers and weeds
grass_spring_01	4 – 7	brown, dry-looking 3-D grass and a few flowers
grass_rus_01	5	lush green 3-D grass and tall flowers and weeds
grass_rus_02	5	the same, but without flowers

Dirt

Texture Layer	Tiling scale	Notes
ground_usa_01	3 – 5	with 3-D dirt clumps
ground_rus_01	4 – 5	the same 3-D dirt clumps as ground_usa_01
ground_spring_01	5	with different 3-D dirt clumps; allows much more hidden mud
ground_test	5	with 3-D stones
ground_snowy	5	with 3-D dirt clumps; wetness is ice; supports snow?

Sand

Texture Layer	Tiling scale	Notes
sand_pbr_02	2.5 – 5	with 3-D dirt clumps
sand_pbr_03	2.5 – 5	only the ground texture differs
sand_test	2.5 – 5	only the ground texture differs; looks more like dunes

Mud

mud	1.5	no 3-D objects
mud_flat	1.5	only the ground texture differs; is a bit less wet looking

Gravel

Texture Layer	Tiling scale	Notes
gravel_road	3 – 5	with 3-D stones
gravel_road_rus	3 – 5	no 3-D objects
gravel_rus_01	2.5 – 20	no 3-D objects
gravel_pbr_test	3 – 5	no 3-D objects

Rocks

Texture Layer	Tiling scale	Notes
rocks_rus_01	1 – 3	no 3-D stones
rocks_rus_01_snowy	1 – 3	no 3-D objects; wetness is ice; supports snow?

Stone

Texture Layer	Tiling scale	Notes
stone_spring	1	no 3-D objects; wetness is ice
stone_snowy	1	no 3-D objects; supports snow?
stone_test	1	no 3-D objects

Concrete

Texture Layer	Tiling scale	Notes
us_concrete	3 – 5	no 3-D objects
us_concrete_old	3 – 5	no 3-D objects
us_concrete_tile	3 – 5	no 3-D objects

Asphalt

Texture Layer	Tiling scale	Notes
asphalt_rus_01	3 – 5	no 3-D objects
asphalt_rus_01_old	1 – 5	no 3-D objects
asphalt_rus_01_snow	1 – 5	no 3-D objects; wetness is ice
us_snow_asphalt	3 – 4	no 3-D objects; wetness is ice

Ice

Texture Layer	Tiling scale	Notes
ice_01	3	no 3-D objects; wetness is ice; can see through to a second layer underneath
ice_02	3 – 4	wetness is ice; is breakable; see TBD

Snow

Texture Layer	Tiling scale	Notes
snow_layer	2.2	supports soft_snow and crust_snow; supports snow depth
snow_spring	?	looks like snow_layer, but behaves like a normal terrain layer

Employers

The following built-in employers are available for contracts (page 184). Alternatively, you can create a custom employer (page 181).

Icon	Editor Name	Localization ID	English Name
	AmurFuelCompany	COMPANY_AMUR_FUEL	AMUR PETROLEUM COMPANY
	AmurRegionalGov	AMUR_REGGOV	AMUR REGIONAL ADMINIST.
	AttecaTownship	COMPANY_ATTECA	STEEL RIVER TOWNSHIP
	BlackBird	COMPANY_BLACK_BIRD	BLACK BIRD
	DysonDiesel	COMPANY_DYSON	DYSON DIESEL
	GarlicOil	COMPANY_GARLIC	THE COMPANY
	GoldhorseMining	COMPANY_GOLD	GOLDHORSE MINING
	GreenLake	COMPANY_GREENLAKE	GREENLAKE PAPERMILL
	GREnterprise	COMPANY_GRE	GR ENTERPRISE
	Husky	COMPANY_HUSKY	<no localization>
	HuskyForwarding	COMPANY_HUSKY_FORWARDING	HUSKY FORWARDING
	KolaExpedition	KOLA_EXPEDITION	KOLA EXPEDITION
	KolskyRegionalGov	KOLSKY_REGGOV	KOLSKY ADMINISTRATION

Icon	Editor Name	Localization ID	English Name
	Maneuver29Agency	COMPANY_MANEUVER_29	MANEUVER-29 AGENCY
	MorrisonMining	COMPANY_MORRISON	MORRISON MINING
	Nonagon	COMPANY_NONAGON	NONAGON INFRASTRUCTURES
	SeverMotorRepairs	SEVMOT_REPAIRS	SEVMOT REPAIRS
	Stonecreek	COMPANY_STONECREEK	STONECREEK LUMBER
	TaigaOil	COMPANY_TAIGA	TAIGA OIL
	TransTaymir	COMPANY_TRANS_TAYMIR	TRANSTAYMYR
	Voronoe12	COMPANY_VORONOE	VORONOE 12
	WoodyWoodpecker	COMPANY_WOODY	<no localization>

Zone Icons

The following icons can be used to mark zones in the game. In the table below, only the 40×40 icon image is shown. The 30×30 icon is similar.

Use the 40×40 icon name in the zone's Icon 40x40 property value, and use the 30×30 icon name in the zone's Icon 30x30 property value.

Travel Services

Icon Image	Purpose	30×30 Icon Name	40×40 Icon Name
	fuel station	fuelStationImg30	fuelStationImg
	garage	garageImg30	garageImg
	gateway (tunnel)	gatewayImg30	gatewayImg
	locked gateway	gatewayLockedImg30	gatewayLockedImg
	service hub	serviceHubImg30	serviceHubImg
	trailer shop	trailerShopImg30	trailerShopImg40
	upgrade station	upgradeCreateImg30	upgradeCreateImg40
	watchtower	watchTowerImg30	watchTowerImg

Objectives

Icon Image	Purpose	30×30 Icon Name	40×40 Icon Name
	contest	contestImg30	contestImg
	new task	taskNewImg30	taskNewImg40
	task	taskImg30	taskImg40
	subtask	taskSubImg30	taskSubImg40
	crafting area	craftCrafting30	craftCraftinging40
	lost cargo	constructionCargoImg30	constructionCargoImg
	lost truck	troubleAltVehicleImg30	troubleAltVehicleImg

Cargo Areas

Icon Image	Purpose	30×30 Icon Name	40×40 Icon Name
	airport	airportImg30	airportImg
	bunker	bunkerImg30	bunkerImg
	drilling site	drillingSiteImg30	drillingSiteImg
	factory	factoryImg30	factoryImg
	farm	farmImg30	farmImg
	lumber station	sawmillImg30	sawmillImg
	storage	townStorage30	townStorage
	timber station	lumberjackImg30	lumberjackImg
	warehouse	constructionWarehouseImg30	constructionWarehouseImg
	manual loading	manualLoading30	manualLoadingMarker40
	manual unloading	manualUploading30	manualUploadingMarker40

Vehicles

Icon Image	Purpose	30×30 Icon Name	40×40 Icon Name
	trucks	allVehicleImg30	allVehicleImg
	scout truck	scoutVehicleImg30	scoutVehicleImg
	highway truck	highwayVehicleImg30	highwayVehicleImg
	heavy-duty truck	heavyDutyVehicleImg30	heavyDutyVehicleImg
	offroad truck	offroadVehicleImg30	offroadVehicleImg
	heavy truck	heavyVehicleImg30	heavyVehicleImg

Ruins

Icon Image	Purpose	30×30 Icon Name	40×40 Icon Name
	brick ruins	ruins30Bricks1	ruins40Bricks1
	brick ruins	ruins30Bricks2	ruins40Bricks2
	brick ruins	ruins30Bricks3	ruins40Bricks3
	metal ruins	ruins30Metals1	ruins40Metals1
	metal ruins	ruins30Metals2	ruins40Metals2
	metal ruins	ruins30Metals3	ruins40Metals3
	wood ruins	ruins30Woods1	ruins40Woods1
	wood ruins	ruins30Woods2	ruins40Woods2
	wood ruins	ruins30Woods3	ruins40Woods3

Miscellaneous

Icon Image	Purpose	30×30 Icon Name	40×40 Icon Name
	GPS marker	markerImg30	markerGpsImg
	living area	craftLivingArea30	craftLivingAreaMarker40
	construction blockage	constructionBlockageImg30	constructionBlockageImg
	border booth	constructionBorderBoothImg30	constructionBorderBoothImg
	hard hat	constructionHardHatImg30	constructionHardHatImg
	offroad 1	offroadAltImg30	offroadAltImg
	offroad 2	offroadAltAltImg30	offroadAltAltImg

Additional Icons

The Saber guide includes a few more icons for which only a 30×30 or 40×40 icon are available. Since none of these looks particularly useful, I didn’t bother to document them.

Saber also documents a few alternative 40×40 icon names for which the icon is in an unoutlined style. These look worse than the other icons, so I see no reason to use them.

There are more icons available in the game, but without documentation from Saber, there’s no way to discover their icon names.

There is no known way to create new icons and integrate them into the game.

Cargo Types

The following tables give the internal name used by the Zone Settings Editor for each of the cargo types. The number of cargo slots used by each is also listed.

Pallets

Cargo Type	Editor Name	Cargo Slots
cement	CargoBags	1
packaged sand	CargoBags2	1
fuel	CargoBarrels	1
oil barrels	CargoBarrelsOil	1
bricks	CargoBricks	1
cellulose bricks	CargoCellulose	1

Crates and Containers

Cargo Type	Editor Name	Cargo Slots
consumables	CargoCrateLarge	1
vehicle spare parts	CargoVehiclesSpareParts	1
service spare parts	CargoServiceSpareParts	1
drilling spare parts	CargoServiceSparePartsSpecial	1
secure radioactive container	CargoRadioactive	1
cargo container	CargoContainerSmall	2
special cargo	CargoContainerSmallSpecial	2
oversized cargo container	CargoContainerLarge	4
drilling equipment	CargoContainerLargeDrilling	4

Frames and Straps

Cargo Type	Editor Name	Cargo Slots
wooden planks	CargoWoodenPlanks	1
concrete blocks	CargoConcreteBlocks	1
concrete slabs	CargoConcreteSlab	2
metal beams	CargoMetalPlanks	2
metal rolls	CargoMetalRoll	1
small pipes	CargoPipesSmall	2
medium pipes	CargoPipesMedium	2
large pipe	CargoPipeLarge	4
small cabin	CargoForkliftCaravanContainer2	2
large cabin	CargoForcklift	4 (requires flat bed)
sequoia trunk	CargoSequoia	5

Vehicle Sections

Cargo Type	Editor Name	Cargo Slots
BA-20 armored car (rusted)	CargoBA20	2
BA-20 armored car (wrecked)	CargoBA20Add	2
rail section	CargoRailway	5
rocket engine assembly	CargoRocketEngine	1
stage 2 rocket fuel tank	CargoRocketPart2	3
stage 3 rocket fuel tank	CargoRocketPart1	3
airplane fuselage part	CargoPlane	5 (requires wide bed)
wing wreckage with engine	CargoWing1	5 (requires wide bed)
wing wreckage	CargoWing2	4 (requires flat bed)

Logs

Logs require a special logs carrier of the appropriate size, and each load fills the carrier.

Cargo Type	Editor Name	Cargo Slots
short logs	CargoLogsShort	*
medium logs	CargoLogsMedium	*
long logs	CargoLogsLong	*

Rock Model Reference

Rocks tend to look good at any scale. The much stronger constraint on rock scale is the accuracy of the collision mesh when the player is rock crawling. A lot of small, overly detailed rocks are bad for performance. On the other hand, scaling up a simplified collision mesh causes the truck's tires to visually float above or sink into the rock surface.

Thus, each rock model has a recommended range of scales that it works best at. Some of these values are derived from Saber's recommendations, and some are my own. I've checked them all, but with the caveat that I'm driving blind when checking the shape of collision meshes. You might find that some orientations work better for rock crawling than others.

* represents any continuation of the model name.

Models suitable for rock crawling:

- rock_03, rock_03_rus scale = 3 – 5
- rock_03_pile* scale = 2 – 4
- rock_03_*_objective scale = 2 – 4
- rock_04_rus scale = 6 – 10
- rock_04a* scale = 3 – 5
- rock_05 scale = 4 – 7
- rock_06* scale = 6 – 10
- rock_rus_ter_01 scale = 6 – 10

Models suitable for blocking a truck's progress:

- rock_01 scale = 1
- rock_02* scale = 1
- rock_07* scale = 1 – 2

Models unsuitable for any purpose:

- rock_rus_ter_01_objective

In case you're curious, the rock_06* models are related to each other as follows:

- rock_06a_cap is the top of rock_06a
- rock_06_cap is one end of rock_06a_cap
- rock_06_cap is also the **bottom** of rock_06
- rock_06a_rus is a recolor of rock_06a
- rock_06a_rus_inst is identical to rock_06a_rus, but has some useful XML properties that Saber forgot to include in the rock_06a_rus model.

Multi-Stage Model Reference

Multi-stage models change state in response to completion of an objective stage (page 190) or delivery of cargo (page 201). The placement of the model itself is described on page 90.

The below table lists all of the multi-stage models through the year 1 DLC. The model asset list provides an overview of the model appearance, but this reference provides the specific names used for each model state and summarizes its appearance in each state.

For each model, the stages are listed in this format:

- <stage_name>: <description of stage>

The first stage listed is the default stage if no objective is associated with the model

Model stage transitions are animated only if marked with "(animated)". By default, each animation is followed by an animated camera, although the player can interrupt it to return to the driving view as the animation completes. If an animation has "no eagle-eye view", then the animation can only be watched from the driving view.

If the description of a stage is followed by an asterisk (*), then the new model stage could easily collide with a parked truck. There may be additional such cases that I didn't notice.

There is also often a chance of a collision if you use model stages in a different order than the default (e.g. to add a barrier instead of remove a barrier with the models below). Also, animations will play only in the default stage order.

A couple of models cause the Editor to throw a warning when they are loaded and/or packed, but I lost my notes about which ones. The warnings seem to be harmless.

Remove Barrier

These models aren't animated and are simply visible in one state and hidden in the other, thus opening a path where the model used to be. If these are used as a source of cargo with a desination elsewhere, no one will notice the lack of animations.

The first set of models all use the same pair of stage names:

concrete_block_03_objective
rock_03_objective
rock_06_objective
rock_rus_ter_01_objective
ghostcity_debris_01_objective
rus_broken_pumptower_01_objective
rus_high_tower_01_objective
wooden_blockage_01_objective

- stage_visible: visible model blocks passage
- stage_hidden: model is hidden and does not block passage

container_03_objective

- build_complete: model visible
- build_stage_0: model hidden

rocket_broken_01_objective

- state_start: model visible
- state_end: model hidden

Open Barrier

These models open a path, similar to the above, but by moving things around in the model, rather than simply hiding them.

barrier_01_objective (animated)

- stage_closed: barrier closed
- stage_opened: barrier open

pipe_01_objective

- state_start: barriers in front of a broken pipe
- state_end: barriers removed and pipe fixed

rus_factory_rocket_01_01_objective (animated)

- stage_0: barriers in front of closed doors
- stage_1: barriers pulled aside and doors open

rus_launchpad_hangar_01_doors (animated)

- stage_0: doors closed
- stage_1: doors open

Deconstruct Structure

These models remove part of a model, in most cases leaving only the foundation. These would presumably be a source of cargo rather than a destination.

abadonned_oil_derrek_01_objective (animated)

- build_stage_0: tower erected
- build_complete: only base remains

rus_constr_bricks_01_objective

- stage_0: partially constructed building with construction material ready to be used
- stage_1: building unchanged; construction material removed

constr_brick_01_objective
us_constr_brick_01_old_objective

- state_start: derelict building on foundation pad
- state_end: foundation pad only

constr_metal_01_objective
us_constr_metall_01_old_objective

- state_start: building frame on foundation pad
- state_end: foundation pad only

constr_wood_01_objective

- state_start: building frame on concrete foundation
- state_end: concrete foundation only

Repair Structure

These models repair a structure. This might make a path a bit easier to get through, but the broken model isn't really designed as an explicit barrier to passage.

pole_a_objective

- state_start: pole lying down
- state_end: pole standing up with wires attached

power_lines_01_objective

- state_start: distribution tower collapsed
- state_end: distribution tower repaired (no wires)

pipe_01_rus_objective

- build_stage_0: broken supports with missing pipe section (not a barrier to passage)
- build_complete: fixed supports with high section of pipe

Build Bridge

These bridge models start with just barriers or with a non-supportive framework and finish with a complete bridge.

bridge_road_01_a_objective (animated)

- build_stage_0: bridge supports and steel underframe only; concrete barriers blocking road access
- build_stage_1: + steel topframe
- build_complete: + wood roadbed and side rails; concrete barriers removed

bridge_road_01_sn_objective (animated)

- build_stage_0: bridge supports and steel underframe only; concrete barriers blocking road access
- build_complete: + wood roadbed and side rails, steel topframe; concrete barriers removed

rus_bridge_road_01_objective (animated)

- build_stage_0: bridge supports and steel roadbed frame only; concrete barriers blocking road access
- build_stage_1: + partial steel topframe and roadbed tension cables
- build_complete: + remaining topframe and roadbed; concrete barriers removed

rus_bridge_road_01snow_objective (animated)

- build_stage_0: bridge supports and steel roadbed frame only; concrete barriers blocking road access
- build_stage_1: + partial steel topframe and roadbed tension cables
- build_complete: + remaining topframe and roadbed; concrete barriers removed

bridge_road_01_b_objective (animated)

- build_stage_0: bridge supports and steel underframe only; concrete barriers blocking road access
- build_stage_1: + steel siderails and roadbed frame, wood roadbed; concrete barriers removed

bridge_wooden_big_02_objective (animated)

- build_stage_0: bridge supports and wood underframe only; wooden barriers blocking road access
- build_stage_1: + wood roadbed; wooden barriers removed

bridge_logs_01_objective

- build_stage_0: wooden side rails only
- build_complete: + log roadbed

Build Other Structure

These models build a structure from a foundation or from nothing.

Drilling Sites

oil_derrek_01_objective (animated)

- build_stage_0: base only
- build_stage_1: + lower structure and conveyer
- build_stage_2: + tower
- build_complete: + equipment

rus_drilling_rig_01_objective (animated)

- build_stage_0: base only
- build_stage_1: + lower structure
- build_stage_2: + mid structure
- build_complete: + tower and conveyer

rus_drilling_rig_02_objective (animated)

- build_stage_0: frame and lower walls
- build_stage_1: + upper walls
- build_stage_2: + minor embellishments
- build_complete: + roof

rus_drilling_rig_03_objective (animated, but no eagle-eye view)

- build_stage_0: frame and end walls
- build_stage_1: + remaining walls
- build_stage_2: + roof
- build_complete: + embellishments

Paper Factory

Each section of the paper factory is split into two models: a static model, and an animated model that should have the same location and orientation. My guess is that Saber did this to reduce the size of each model. Unfortunately, the Editor will throw warnings about the duplicate model locations. Use the ‘Do not show this warning again’ checkbox to disable them.

The animation camera movement is also strange for paper_factory03_objective. Presumably there should be a nearby building that the camera wants to include in the view.

paper_factory01_objective (animated) (should be placed with paper_factory01_stage_0)

- build_stage_0: nothing
- build_complete: everything

paper_factory02_objective (animated, but no eagle-eye view) (should be placed with paper_factory02_stage_0)

- build_stage_0: nothing
- build_complete: everything

paper_factory03_objective (animated) (should be placed with paper_factory03_stage_0)

- build_stage_0: nothing
- build_complete: everything

paper_factory04_objective (animated) (should be placed with paper_factory04_stage_0)

- build_stage_0: nothing
- build_complete: everything

paper_factory05_objective (animated, but no eagle-eye view) (should be placed with paper_factory05_stage_0)

- build_stage_0: nothing
- build_complete: everything*

paper_factory06_objective (animated, but no eagle-eye view) (should be placed with paper_factory06_stage_0)

- build_stage_0: nothing
- build_complete: everything

US Three-Part Factory

us_3part_factory_1_part_objective (animated)

- build_stage_0: building and boilers
- build_stage_1: + rooftop and boilertop additions
- build_complete: + smokestakes and other details

us_3part_factory_2_part_objective (animated)

- build_stage_0: main structures and pipe foundations
- build_stage_1: + connecting pipes and silotop additions
- build_complete: + more pipes and embellishments

us_3part_factory_3_part_objective (animated)

- build_stage_0: main structures and foundations for silos and pipes
- build_stage_1: + silos and taller buildings
- build_complete: + pipes and rooftop stuff

Russian Fuel Factory

rus_fuel_factory_05 is split into two models similar to the paper factory above. (In fact, these models essentially duplicate paper_factory03 except for the textures.)

rus_fuel_factory_03_objective (animated)

- build_stage_0: building and boilers
- build_stage_1: + rooftop and boilertop additions
- build_complete: + smokestakes and other details

rus_fuel_factory_04_objective (animated)

- build_stage_0: main structures and pipe foundations
- build_stage_1: + connecting pipes
- build_complete: + more pipes and embellishments

rus_fuel_factory_05_objective (animated) (should be placed with rus_fuel_factory_05_stage_0)

- build_stage_0: nothing
- build_complete: everything

Russian Base

rus_base_01_building_objective (animated)

- build_stage_0: metal building frame
- build_complete: building complete

rus_base_02_building_objective (animated)

- build_stage_0: metal building frame
- build_complete: building complete

Russian Control Bunker

rus_controlbunker_02_1_part_objectives (animation broken)

- build_start: building foundation
- build_stage_1: + walls, poles, and wires (after a delay caused by the broken animation)*
- build_end: + no change

rus_controlbunker_02_2_part_objectives (animated)

- build_start: building foundation and derelict helicopter
- build_stage_1: building foundation (helicopter gone)
- build_end: fixed foundation and new helicopter

rus_controlbunker_02_3_part_objectives (animation broken)

- build_start: derelict building
- build_stage_1: new building (after a delay caused by the broken animation)
- build_end: + no change

rus_controlbunker_02_4_part_objectives (animation broken) (works best on top of a building to avoid rotating through a truck)

- build_start: broken dish
- build_stage_1: fixed dish, stationary (after a delay caused by the broken animation)
- build_end: fixed dish, rotating (after a delay caused by the broken animation)

Other Buildings

conveyor_objective_01

- state_start: building frames and supplies
- state_end: complete buildings and connecting conveyer*

radio_station_rus_objective

- state_start: bare tower and concrete foundation for equipment
- state_end: complete tower and equipment*

us_factory_sorting_01_objective (animated)

- build_stage_0: buildings
- build_stage_1: + rooftop structure and silos
- build_complete: + more rooftop stuff and connecting covered conveyer

us_mine_01_objective (animated)

- build_stage_0: buildings
- build_stage_1: + more building height and some roofs
- build_complete: + final roofs, connecting conveyers, and embellishments

rus_pier_01_objective (animated)

- build_stage_0: pilings only
- build_stage_1: pier complete

Specialty Models

I'm not sure if trailer_train_rocket_01_collision was a test model that got released accidentally, or whether it's intended to keep trucks out of trailer_train_rocket_01_objective prior to build_stage_1. If the latter, then you still need to advance trailer_train_rocket_01_collision to build_stage_1 somehow, e.g. by specifying it as the only model stage used by one of the initial contracts.

trailer_train_rocket_01_collision

- build_stage_0: no visible model; no collision box
- build_stage_1: no visible model; collision box for train only
- build_stage_2: no visible model; collision box for train only

trailer_train_rocket_01_objective (animated)

- build_stage_0: idle gantry only
- build_stage_1: train pulls in and erects rocket in gantry*
- build_stage_2: rocket launches, leaving idle gantry and empty train

bridge_and_train_objective (animated) (triggers Editor warnings about the model, but it seems to work OK)

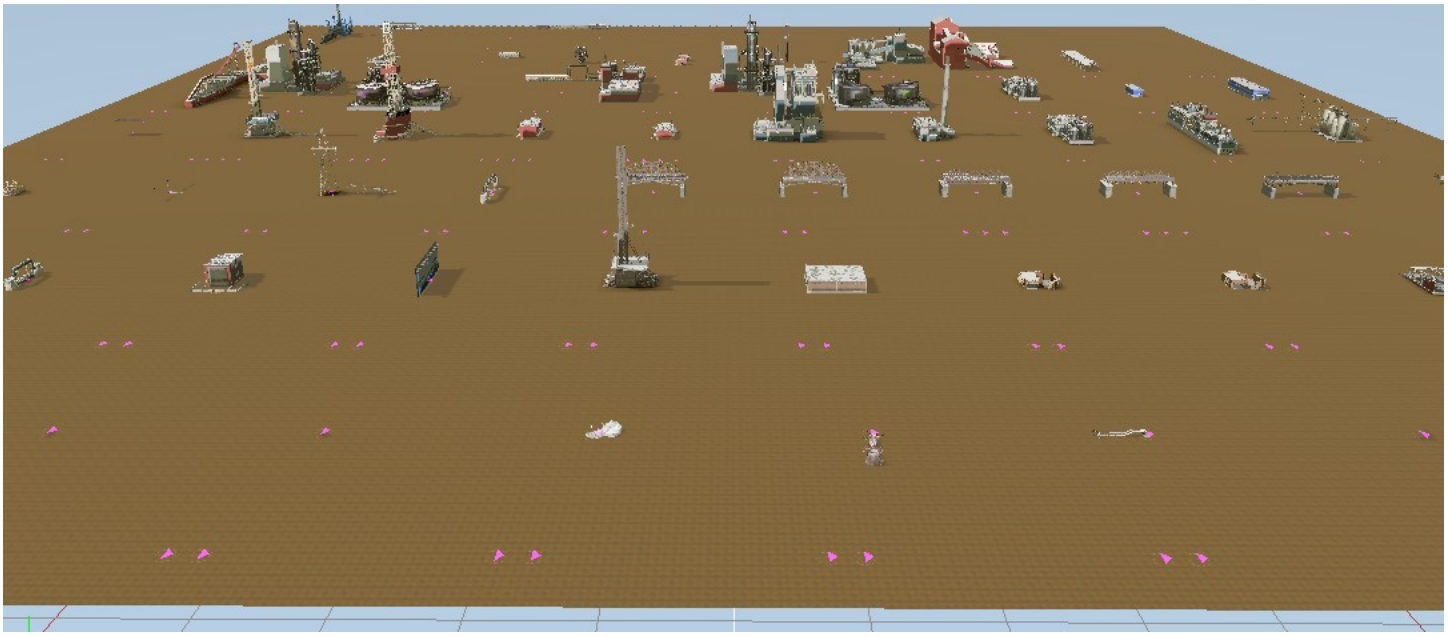
- build_stage_0: rail frame with steel upperframe at one end only
- build_complete: + finished frame and train rails; trail rolls through backwards and disappears in the distance
- train_hide: + no chage (train is already hidden)

rail_blocker_objective_01 (triggers Editor warnings about the model, but it seems to work OK) (requires a fair amount of straight, level track on either end)

- build_stage_0: train rail section with missing portion; barriers at each end of section
- build_complete: complete train rail section; no barriers

train_carriage_objective

- empty_carriage: empty train carriage
- cargo_carriage: train carriage carrying sequoia log
- hide_cargo_carriage: model hidden



Appendix B: Hand Edit the Map Files

The SnowRunner Editor makes it easy to edit your map and immediately see your changes. However, you sometimes need to do something that requires more power than what the editor offers or that is simply easier done outside the editor. The following sections describe what you need to know in order to edit files outside the SnowRunner editor, using either a text editor or an image editor.

Material Layout Bitmap

The material layout bitmap is named mtrl_layout.tga. It has one pixel for each terrain block, where the pixel value maps to terrain features as follows:

- a non-zero value in the red channel maps to the first material.
- a non-zero value in the green channel maps to the second material.
- a non-zero value in the blue channel maps to the third material.
- a non-zero value in the alpha channel maps to the fourth material.

If multiple channels are non-zero, an earlier material has priority over a later material. If all channels are zero, the Editor defaults to the first material.

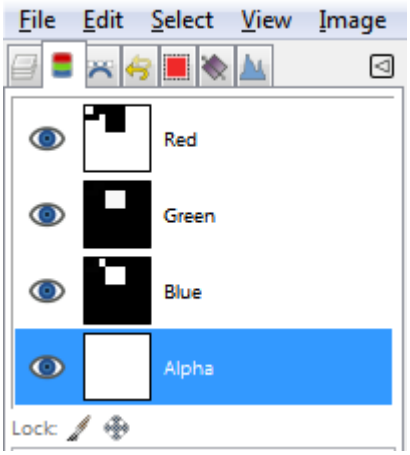
When painting material opacity in the Editor, the tint that the Editor applies to the terrain is taken directly from the R/G/B channels, so low values and/or conflicting values could be deceptive. The Editor ignores the alpha channel when tinting.

The editor has a bug in that it fails to put a value in the alpha channel when you paint opacity for the fourth material. Instead, it sets all channels to zero. This can be fixed in an image editor by assigning a non-zero channel to the alpha channel for all pixels. Since the fourth material has lowest priority, the Editor will correctly use the fourth material only when no other material is used.

Fix the Material Layout Alpha Channel in GIMP

It can be difficult to figure out how to use an image editor to edit the alpha channel while leaving the other channels alone. Here is the procedure that I use in GIMP.

- Open mtrl_layout.tga in GIMP. Because the alpha channel is zero everywhere, the image is transparent.
- Click the tab above layers to switch to individual Channels.
- Click each of the R, G, and B channels so that they are disabled.



- Click the paint bucket to fill the alpha channel with white. The entire image is now opaque.
- Select File->Overwrite mtrl_layout.tga

In the SnowRunner Editor, rebuild the terrain. Since the alpha channel is now non-zero, the fourth material is now correctly used where the R, G, and B channels are all zero.

Ambient Music

Your map can have ambient music that plays in the background. Different tracks can play at different times, plus a unique track in the garage.

To set up your ambient music, create a file: %USERPROFILE%\Documents\My Games\SnowRunner\Media\classes\sounds\music_presets.xml. You may need to create the classes\sounds directory. Note that this is **not** the same directory as where sound files are stored.

This XML file contains a section for each of your maps. Below is an example with music for only one map:

```
<MusicPresets>
  <LevelMusic LevelName="level_test">
    <Sound
      Name="bass solo"
      StartTime="0"
      EndTime="2"
      FadeInTime="15"
      FadeOutTime="7.5"
    />
    <Sound
      Name="drums"
      StartTime="19"
      EndTime="24"
      FadeInTime="15"
      FadeOutTime="7.5"
    />
    <GarageMusic
      SoundName="music/ipanema"
      FadeInTime="15"
      FadeOutTime="7.5"
      Volume="0.8"
    />
  </LevelMusic>
</MusicPresets>
```

Level Definition

```
<MusicPresets>
  <LevelMusic LevelName="map_name">
    ...
  </LevelMusic>
</MusicPresets>
```

> *map_name*: string
>
> Specifies the name of the map for which music is being defined.

If you make multiple maps with music, they will all be listed in the same music_presets.xml file, even if the maps are not related to each other.

Driving Music

Different music can be defined for different times of day.

```
<MusicPresets>
  <LevelMusic LevelName="map_name">
    <Sound
      Name="sound_file"
      StartTime="start_hour"
      EndTime="finish_hour"
      FadeInTime="fade_in"
      FadeOutTime="fade_out"
    />
    ...
  </LevelMusic>
</MusicPresets>
```

Sound File

> *sound_file*: string
>
> Specifies the relative path and base filename of the sound file(s) to play.
>
> Default: unspecified; no music plays for this Sound definition.

Choose a sound file or set of sound files in the same way as for a point sound (page 99). The built-in music files are in the [sound]/music directory. The Editor will display a warning dialog if the sound file is specified but not found, but not if it is left unspecified.

Since it doesn't have an aural direction, stereo sound is officially permitted for an ambient music.

In order to match the volume of the built-in campaign music (so that the player doesn't have to reach for the audio settings), Saber recommends that the music files have an internal volume of –7 dB.

Hour Range

> *start_hour*: numeric, in hours after midnight, in the range 0.0 – 24.0, inclusive
>
> Specifies the hour when this Sound definition starts playing music.
>
> Default: 0.0 (midnight).

> *finish_hour*: numeric, in hours after midnight, in the range 0.0 – 24.0, inclusive
>
> Specifies the hour when this Sound definition finishes playing music.
>
> Default: 0.0 (midnight).

The **_hour* values can be fractional to start or finish in the middle of an hour. Values out of range are treated as 0 (midnight). The game does the right thing if *start_hour* is greater than *finish_hour*: it ends at the finish hour in the next day.

If you are using music that is tied to the time of day, refer to page TBD for when the light conditions change at the start and end of each part of the day.

If *start_hour* equals *finish_hour*, no music plays for this Sound definition. To play one track (or sequence) all day, set *finish_hour* to just under *start_hour*, e.g. 0.0 and 23.999. The music will awkwardly fade out and fade back in at the start of the track at midnight, but that's the best we can do.

Fade In and Out

> *fade_in*: numeric, in seconds
>
> Specifies how long it takes this Sound definition to fade in when it start playing.
>
> Default: ~20.

> *fade_out*: numeric, in seconds
>
> Specifies how long it takes this Sound definition to fade out when it stops playing.
>
> Default: ~15.

In many cases, one Sound definition ends as another starts, e.g. at a particular hour, or when the player enters or leaves the garage. In this case, the fade-out of the previous music overlaps with the fade-in of the new music, so make sure they are arranged accordingly.

Bug: While not in the garage, the game fades in every track as it begins playing, even if it's still within the same the Sound definition. This bug does not apply to fade out, and it doesn't apply within the garage. If the sound files are short, the garage can even spread its fade in across multiple sound files.

Even with *fade_in* of 0, the music tracks are not played cleanly back to back, so don't try to make a clean loop from the end of a track to its beginning. Every track should have a built-in fade in and fade out at the beginning and end to avoid an abrupt transition.