

RuneSword II Tome Building: Fundamentals

By VampiricDread

Edited by Argh 05/10/2002.

Introduction

This is a special tutorial. It should lead you, wanna-be Creator of adventures, to the understanding of what you must do. . . to create your own Tome.

There are a number of folders in the file you've(hopefully) downloaded along with this tutorial. You will need these files to reproduce everything this tutorial attempts to teach you. These files should be extracted and placed in the appropriate RuneSword II (hereafter referred as RS) folders, as per the instructions. I suggest you use the default folder structure when unzipping this, to keep things neat and organized.

Note: if all of the above looks like Greek to you, then you're going to need to brush up on your knowledge of Windows® and the use of .ZIP files before tackling the rest of this tutorial. This tutorial is intended for folks who're fairly familiar with Windows, and while we've tried to keep it as simple as possible, things will get rather confusing if you don't know what a "directory" is, or how to rename a file. Of course, we could just be writing a lousy manual- if you think so, shoot us an email, and we'll try to help. We want more people to write Tomes for RuneSword. Really.

The objectives of this tutorial are:

1. [How to plan a plot.](#)
2. [How to determine your needs.](#)
3. [How to work with basic RS objects.](#)
4. **How to use RuneScript**(at least the basics- for advanced lessons, see the URL [www.URL TO BE DETERMINED.com](http://www.URL_TO_BE_DETERMINED.com) for more information).

The materials included are authored and copyrighted by some of the hard-core fans of RS, or used with permission through CrossCut Games and Shrapnel Games. While most of them are visible in the credits of your licensed copy of RS, Count0 is not and special mention should be paid to him and his excellent program TileSet Cannibal.

Please note that visiting and supporting the following web pages with your contribution and opinions helps the total community of RS users.

Thank you and please enjoy.

J. P. Carvalho (also known as VampiricDread)

Greg Wolfe (also known as Argh)

Planning your plot [Home](#)

This is where a Creator really shines. The RS engine allows you to build just about any environment, from pre-historic adventuring to the wildest futuristic fantasies. You can use or forget about magic; you can create an urban or wilderness environment for the adventure, or go for a classic dungeon. You're not limited to a pre-built game, like so many other "adventure creation tools"- you can re-write the laws of physics, create entirely new Skills, Spells, Objects and Monsters. . . whole new worlds. Best of all, it's fairly easy! Read on. . .

This is a tutorial, so we will limit ourselves to building a small open environment and a small dungeon. Our plot is basically linear: the Party reaches the village of Tutorialville, plagued by nasty monsters that dwell in a nearby cave. The Player's objective is to kill the evil overlord of these monsters, thereby saving the village.

There are no other objectives in this tutorial - no side quests or additional plots to unravel. The Party should easily get in touch with a NPC (non-player character), who gives the quest. The town will have a building, where the Party can trade, buy, and sell weapons, healing potions, and the like. For the sake of simplicity, we will use the Eternia WorldRules, which are the default WorldRules that come with RS. Most Tomes use these WorldRules at this time - but some of the more elite Creators are working on WorldRules that are completely different from the defaults.

Deciding what you need [Home](#)

For our little adventure, we're going to make the following places, items, and monsters:

- Three maps: one for the open, one for the trading building, and one for the dungeon.
- One NPC trader for the trading building.
- The monsters inhabiting the dungeon.
- A monster-boss.

First, here's the plan for the maps:

1. A small map with a entrance point to the Tome. We'll add a few houses (to give the illusion of a village) and a trade building. We'll also need an entrance to the dungeon.
2. We also need the trading outpost inside the trade building, where the adventurer may conduct business. This is a separate map from the outside world map, because it depicts the inside of a building. It's not really necessary to make buildings work this way, but we've done it here to teach you about making doorways and such.
3. Last but not least, we'll need a map for the dungeon, with three rooms: entrance, monsters, and boss hall.

We require a Trader to sell weapons, armor, and healing potions to the Party. The Trader is an Encounter that we'll write so that players can get a hold of some items they may not have at the beginning of the adventure.

Last but not least, we'll use some monsters included with RS. The Boss monster will be a little different - but hey, that's why you're reading this tutorial, right? Read on. . .

Note: the full mini adventure is in the Simple Tome – 11 folder. If you'd like to see what the entire tutorial adventure is **supposed** to end up like, open up that Tome in RuneSword.

Working with RS objects [Home](#)

Just about anything you do with RS is an object, or part of one. These lessons are easier to learn by example, so you should print out this document now so that you can look at it whilst building your first Tome.

Map Drawing

First, there was the Map. Then there was the Word, and then there was Code. Let's talk about maps for a moment. RuneSword uses Maps to represent the physical location of a Party. Since Maps represent the physical world that your players are going to manipulate, much effort must be spent on making a Map convincing, useful, and logical to the setting. Making maps look really nice isn't easy - you have to make your own tilesets, build custom code, and think rather carefully about the world you want players to experiment. This tutorial will teach you the basics of making a Map - further refinements are up to you!

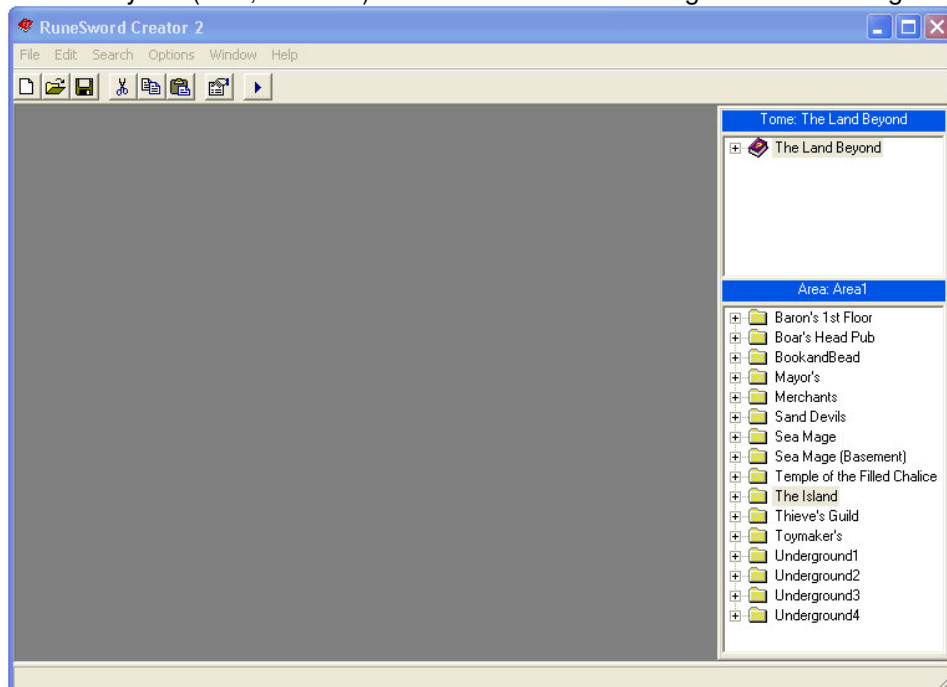
There are two ways to create your map: you can build it from scratch or use somebody else's work as a starting point (although outright plagiarism isn't encouraged). In this tutorial, we're going to start by using someone else's work - when you're starting out, it's easier to use the content that came with RuneSword.

The process of creating a map follows very simple rules:

1. Choose a tile set containing all the tiles you'll need for your map. **Maps can only have one tile set.**
2. Check tile properties. Even when copying a tile set from another map (we'll get into the details later on), not all the tiles you may need may have their properties set, so you should check the properties of the tiles you'll be using.
3. Define the map size (x tiles wide by y tiles across).
4. Paint the bottom layer, then the middle layer, then the top layer.
5. Define special properties on the map, like entry/exit points.
6. Test your map.

We shall now do this for our project, starting with the dungeon. First create a folder in your Tomes directory. Let's call it "Simple". Drag-and-drop the tutorial files into this directory. Later on, we will move our work to more appropriate places, but for now Simple will hold whatever we build.

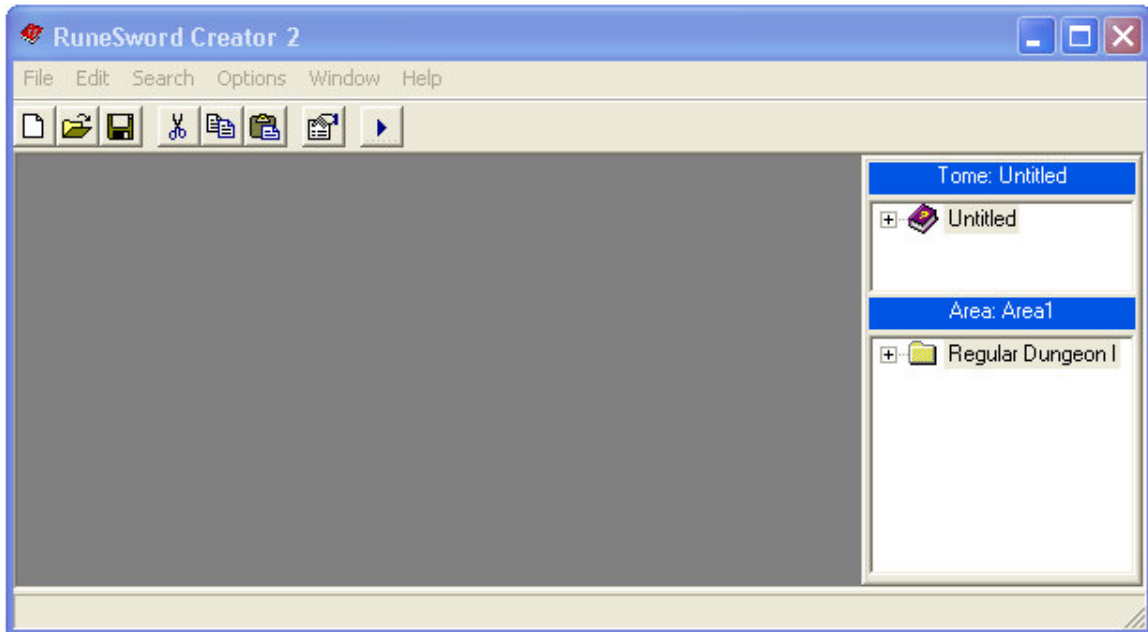
Open Creator. Accept the "Load an Existing Tome" option and click next. Locate and open The Land Beyond (TLB, for short). You should see something like the following:



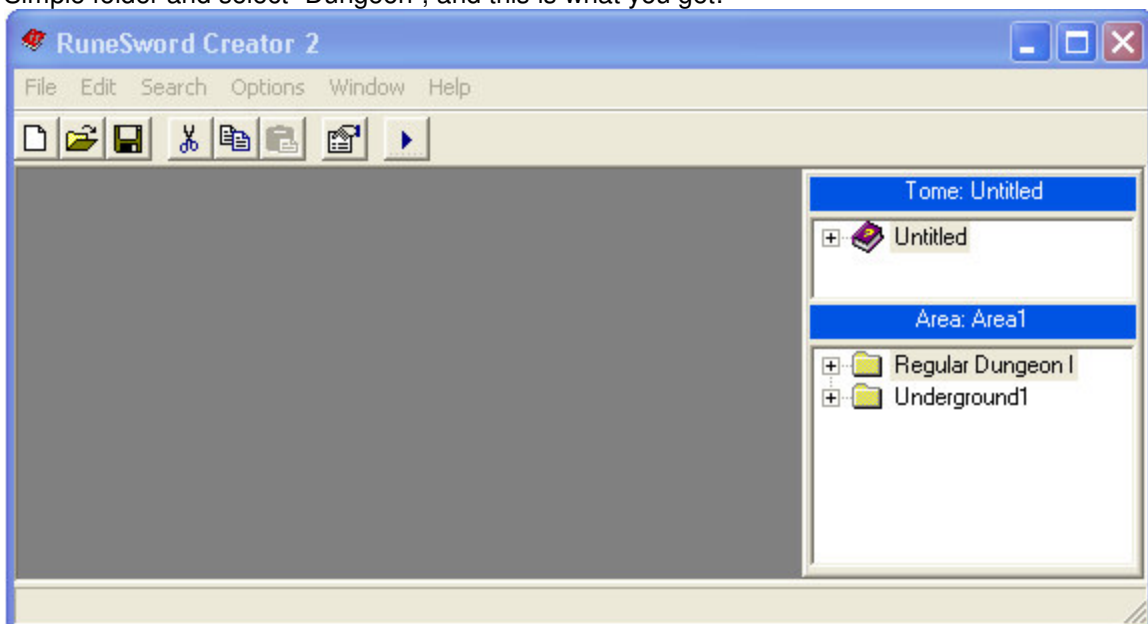
Right-click once the folder icon next to Underground1 to open the options box, and select “Save Map to Library...”. Make sure you save the map to the Simple folder, and that you rename it to Dungeon.

Now let’s go on to the preparation of our Tome.

First, close and re-open Creator. This will clear the memory Creator uses to store information about Tomes, giving you a “clean slate” to play with. This time, when Creator asks what you’d like to do, choose the option “Create a Tome by Hand”. Click Next and you’ll see the following:



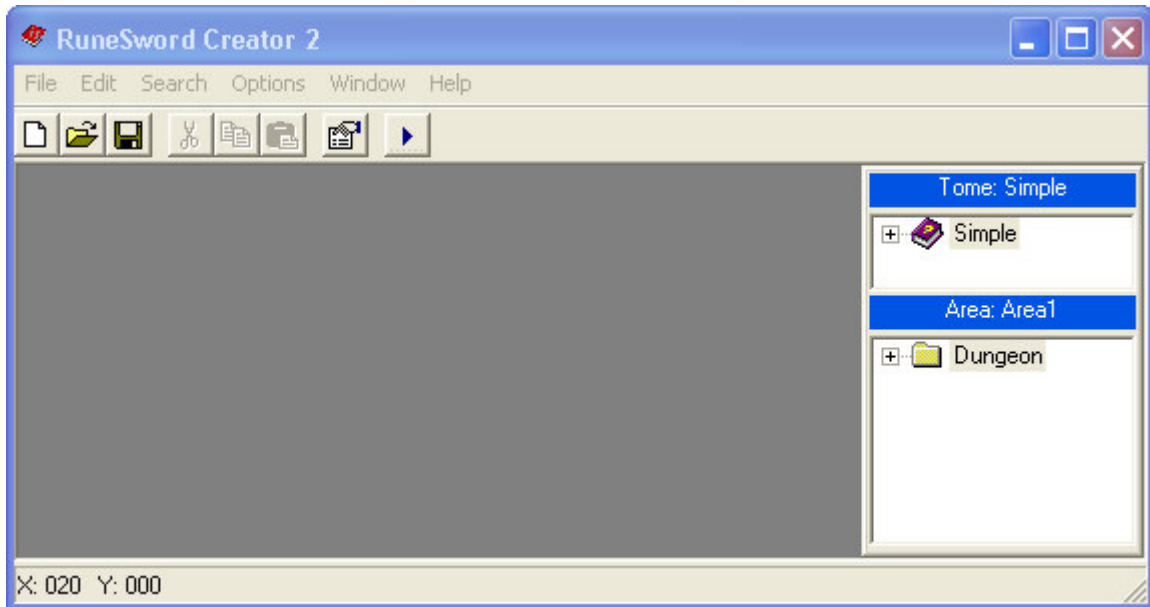
Right-click “Regular Dungeon I” and select “Insert Map from Library...”. Navigate to our Simple folder and select “Dungeon”, and this is what you get:



The complete Underground1 map is now at your disposal under a different Tome. Let's clear some things: Right-click "Regular Dungeon I" and select Cut. It's now gone.

(Side note: A fresh Tome always requires at least one Area, with one Map.)

Double-click Underground1 to open its properties. Rename it Dungeon, and get rid of the comment "danger". Make the width 32 and the height 11, then click Apply, then OK, or just OK. Next, do the same in the Tome properties to rename it to Simple instead of Untitled. If you click on the minus box in the Tome, you get this:



Save your work by selecting File in the menu bar, then "Save Tome As...", rename it to Simple and make sure you save it to our Simple folder. It should now contain 3 files: Dungeon.rsm (for RuneSword Map), Simple1.rsa (for RuneSword Area), and Simple.tom (for Tome). Please delete Dungeon.rsm, because we no longer need it. Its information is already stored in the other 2 files.

Again, indulge us and close and re-open Creator, this time selecting our Simple Tome from "Load an Existing Tome" next, and after locating the Simple.tom file, by clicking OK. Let's do some cleaning before moving on.

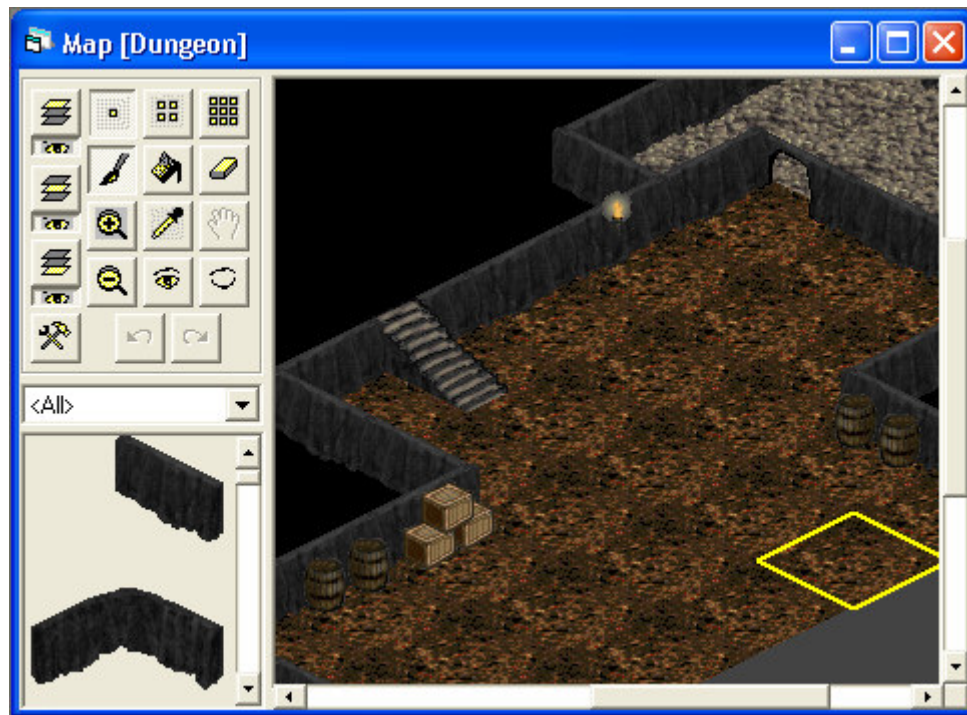
Click the plus sign next to Dungeon, then click the plus sign next to Entry Points. Get rid of the three defined entry points by using "Cut" (the same procedure we used above), or by selecting each entry point and pressing Ctrl+X (the Cut shortcut key).

Do the same for all the Encounters defined in the map. Save your work on top of the previous save.

TIP: always use "Save Tome As..." to make sure you're saving what you want!

We have now a Tome with a single map (called Dungeon), that has all the tiles and definitions from Underground1. Before we can use it, however, we'll need to clean it up. Open up Dungeon and double-click on Graphic. You'll be presented with a partial map, because we have re-defined its size. Now is a good time to introduce you to all the tools we have available for our map painting.

For the sake of simplicity, please refer to page 46 of the RuneSword II User's Manual. The following image shows all the elements which we will be using next:



Note that the “Paint Graphics” window (above, left) opens with 5 pressed buttons by default: the eyes buttons under each layer (the buttons on the left), the brush button (which, not surprisingly, looks like a paintbrush), and the brush size button (the button in the top row showing a single square).

Using the bars on the graphic window, move around to where you have some painted tiles (you can maximize the window for better ease of work). Try to center your view on the one above, so that the next instructions will make more sense ☺

Notes about the buttons:

The “eye” buttons reveal and hide the Bottom, Middle and Top Layer. As you can see (or not), the tile objects on each layer are shown or not according to the position of these buttons. They are also very useful in conjunction with the eraser button, because the eraser will only operate on the visible layers, which is a great way to erase just one particular tile in a certain layer, without affecting placed tiles in other layers.

The zoom in and zoom out buttons are self explanatory, and you can work with all the other functions, irrespective of the zoom level selected.

The brush is used to paint tiles in, the fill will use the currently selected tile to fill within tile boundaries, and the eraser deletes painted in tiles. The eye dropper used on the paint window picks the appropriate tile on the tile window (below the buttons), according to the active layer.

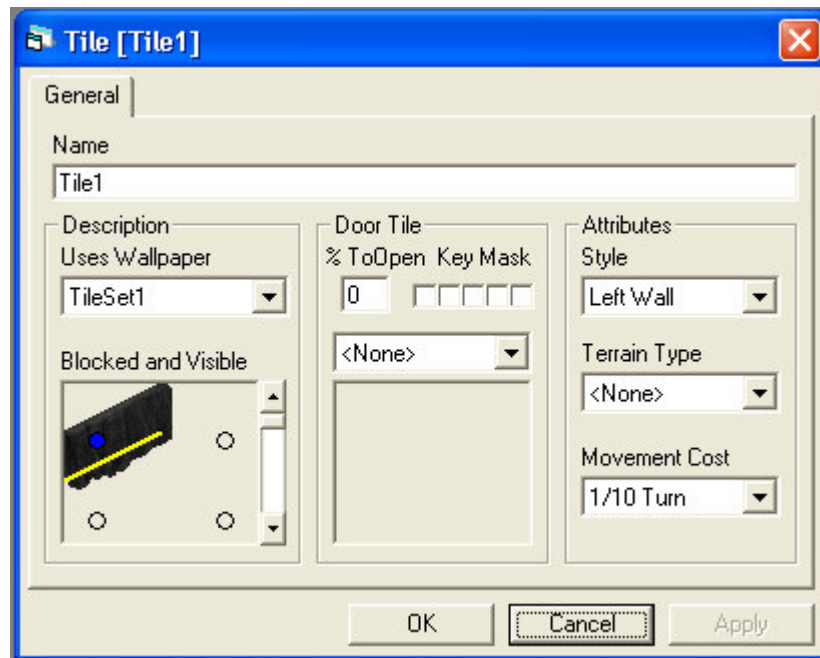
We'll leave the fog buttons and the paint Encounter buttons for later lessons.

The brush sizes cover one, four, or nine tiles in one go.

So, let's start. Select the biggest brush size and the eraser and erase the map. When you finish, save the Tome.

You can now check tile properties, and you can do it in two ways: either you right-click in a tile and then select “Tile Properties” on the tile pick list window, OR you can double-click a tile in the tile folder of the map in the Area window (bottom right). Try it out.

Let's see the tile properties now. Open the tile properties of the first tile, and you'll get the following window:



Page 51 of the RuneSword II User's Manual explains all the properties of this window. What it doesn't tell you is that if you select some of the particular styles in Attributes, it automatically updates the “Blocked and Visible” properties, which is very useful if you're making a standard wall, doorway, etc.

Pick the last defined graphic (tile 72, not 73, which is blank). See the difference? Should you use this tile, the wall would show on your map, but it would also be transparent to the adventurer's vision and passable, like it wasn't there. The blue dots signal blocked sight from that angle, and the yellow lines blocked movement.

In case your experiments have messed up your Tome, you'll find a copy of the work up to now in the Simple Tome - 1 folder. Let's start painting the rooms and the connecting corridors.

The Simple Tome – 2 has the Dungeon map bottom layer partially painted, but the rooms do not have all the tiles of the floor (tile 8 and tile 67). If you copy its contents to your Simple Tome folder under Tomes and open it in Creator, you may see it, and making sure you've selected (shown as depressed buttons) all the eyes under the layer buttons, the bottom layer button, the smaller brush size and the paint brush. Now each left-click of the mouse will paint your selected tile on the position of the yellow diamond on the paint window, so you must select the appropriate tile first before completing the floor of the rooms. Alternate with use of the eraser to clear mistakes and repaint whatever is necessary.

Having done this practical exercise, you may save your work if you wish, or close Creator and replace the files in your Tome folder with those in Simple Tome – 3.

Now we're going to paint the walls. It's the same as before, using tile 1 and tile 2, only, BUT they are painted on the middle layer – check it out using the eyes for the layers. If you make a mistake, be careful on the usage of the eraser not to delete tiles on several layers! Notice that we've left at least one tile space around the borders, and notice why it's needed to place some of the wall tiles.

Finally we get to the top layer. It is used to place things that you want, but conflict in space with the previous two layers, like doors, decoration on the walls, stools under tables, fireplaces, torches or lights, and so forth. The Simple Tome - 4 folder shows the finished dungeon, which I advise you to examine, namely the door, and the objects placed, using the eyes or tile properties. The next two bits of the tutorial cover the setting up of the Tome and the placement of Encounters, and use the files in this folder.

To use the Tome (and test what was made), we need to make a few changes to it: set some properties of the Tome, and change two things in the map.

Start Creator and load the Tome. First open the graphic of the Dungeon, maximize its window and zoom out to see the whole map. Then click on the add fog button (the closed eye, to the right of the remove fog – the open eye). Paint anything that is visible on the map, if required; when you want to add visibility, instead of letting the surroundings be revealed as your party progresses, you use the remove fog button.

Next check the coordinates of the entry point we will use. Move around the cursor (effectively dragging the yellow diamond) until you reach the entry room, where one of the tiles has coordinates X : 001 and Y : 008, presented on the bottom left of your Creator window. Those are the coordinates we need to define the entry/exit point. Please restore down your graphic window, or close it.

In the Area window, right-click EntryPoints and select New EntryPoint. You'll get the following window (please refer to page 40 of the User's Manual):

The screenshot shows a Windows-style dialog box titled "EntryPoint [EntryPoint1]". It has a "General" tab selected. The dialog is divided into several sections:

- Description:**
 - Name:** A text box containing "EntryPoint1".
 - Hide Exit Sign?:** An unchecked checkbox.
 - Description when enter/exit:** A large empty text area.
 - Style:** A dropdown menu set to "Exit".
 - To Town:** A dropdown menu set to "To Town".
- Location on Map:**
 - At X:** A text box containing "0".
 - At Y:** A text box containing "0".
- Links To:**
 - Area:** A dropdown menu set to "[Main Menu]".
 - Map in Area:** An empty dropdown menu.
 - EntryPoint on Map:** An empty dropdown menu.

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

Edit this window as shown below, and click OK.

The screenshot shows the 'EntryPoint [EntryPoint1]' dialog box with the 'General' tab selected. The 'Description' section includes a 'Name' field with 'Dungeon', a 'Hide Exit Sign?' checkbox, and a 'Description when enter/exit' text area containing 'This is the entrance to the dungeon. Do you wish to exit?'. The 'Style' section has 'Exit' and 'To Wilderness' dropdowns. The 'Location on Map' section shows 'At X: 1' and 'At Y: 8'. The 'Links To' section has 'Area' set to '[Main Menu]', 'Map in Area' as an empty dropdown, and 'EntryPoint on Map' as an empty dropdown. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

Notice that not covered in the User's Manual is a check box named "Hide Exit Sign?". With RuneSword II this box, if checked, allows you to hide a visual sign marking entrances and exits. Since this is (for now) the only map, with a single entry and exit point, we've defined it to link directly to the Main Menu of RuneSword II, thus providing you with an exit point (more about this later).

Now, double-click the Tome window on Simple, the name we gave it, and it opens the Tome properties window (page 53). We've already filled it in, so make sure you put the proper info as shown here, before clicking OK.

The screenshot shows the 'Tome Properties' dialog box with the 'General' tab selected. The 'Description' section includes a 'Name' field with 'Simple', an 'On Adventure' checkbox, and a 'Comments' text area containing 'Tutorial on building tomes'. Below this is a table with columns: Party, AvgLvl, Turn, Day, Moon, Year. The values are: Party 5, AvgLvl 1, Turn 100, Day 25, Moon 5, Year 5000. The 'Starting Location' section has 'Area' set to 'Area1', 'Map in Area' set to 'Dungeon', and 'EntryPoint on Map' set to 'Dungeon (1.8)'. The 'Current Location' section shows 'At X: 0' and 'At Y: 0'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

Save your Tome. You are ready for the first exploration of an empty dungeon. You can go from Creator, by clicking the “Run the Tome” button below the menu bar, or by going to RuneSword II and play it from there. To exit, go to the exit point we have defined. When you’re happy with all your experiments return here to start populating the dungeon.

Populating the Dungeon

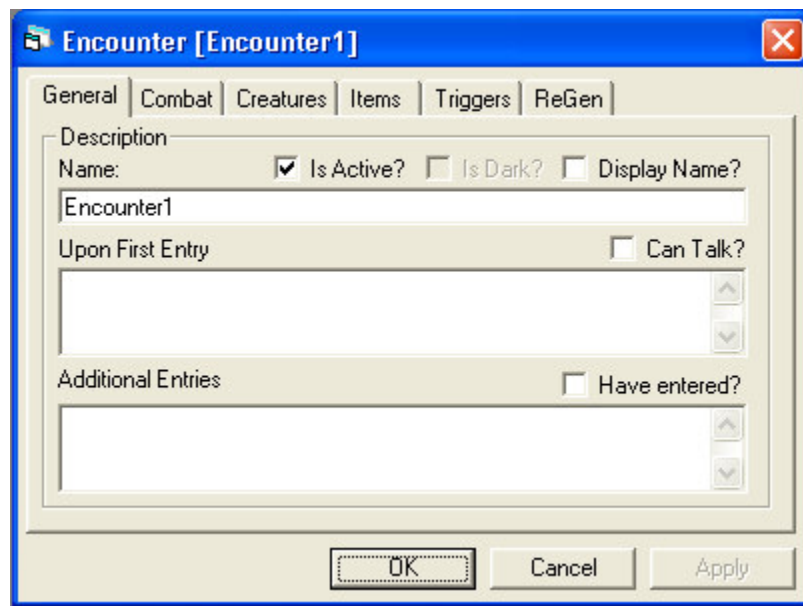
Now is the time to add the creatures. We’ll put in a couple skeletons, a giant scorpion and create a boss.

To add whatever item or creature we want the player to interact with we must have Encounters. In fact, we can also have Encounters without items or creatures, placed to interact with the player and enrich the gaming experience we’re offering, using dialogs or special descriptions, resorting or not to triggers.

Encounters are described briefly on page 38 and 39 of the User’s Manual. Just think of them as black boxes where you store surprises for the party of adventurers.

We will require three Encounters: one for the skeletons, one for the scorpion, and one for the boss. Upon completing this section, you will be able to use the basic functions of Encounters.

Start Creator2 and load our Simple Tome (version 5). Right-click on Encounters under our Dungeon map in the Area window, and select New Encounter. You’ll be rewarded with the following window:



The image shows a screenshot of the "Encounter [Encounter1]" dialog box in the Creator2 software. The dialog has a blue title bar with a close button (X) in the top right corner. Below the title bar are several tabs: "General", "Combat", "Creatures", "Items", "Triggers", and "ReGen". The "General" tab is currently selected. Inside the "General" tab, there is a "Description" section with a "Name:" label and a text box containing "Encounter1". To the right of the text box are three checkboxes: "Is Active?" (checked), "Is Dark?" (unchecked), and "Display Name?" (unchecked). Below this is a section labeled "Upon First Entry" with a text box and a "Can Talk?" checkbox to its right. At the bottom of the dialog, there is a section labeled "Additional Entries" with a text box and a "Have entered?" checkbox to its right. At the very bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

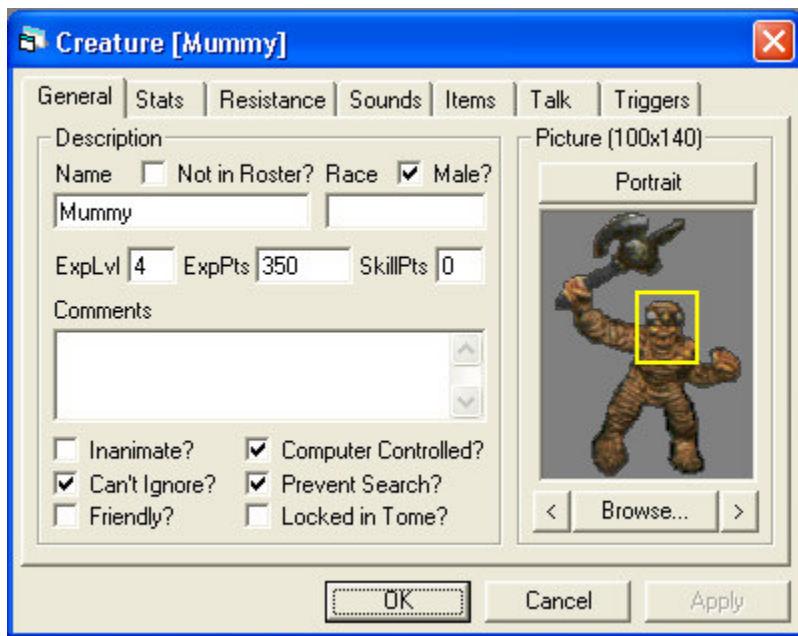
Notice that there’s an option not described in the User’s Manual, the flag “Is Active?”; its purpose is to make the Encounter active, that is, once the party sets foot in the Encounter area, the Encounter will take place. You can have inactive Encounters that become active through the use of triggers that change this property, like creating a group of monsters on a certain location that was initially vacant when you first crossed it. It does give ideas, doesn’t it?

Note that the property “Is Dark?” is not implemented at the time of this writing. Moving on.

Click OK, then open up Encounters by clicking on the plus sign, and you'll find our Encounter1 (we did not change the name) there. Now right-click it and select "Insert Creature from Library...". Go to BasicSet, then Creatures. A whole bunch of creatures are already there, made by the people that brought to you RuneSword II. Pick the Skeleton Swordsman, then repeat the process from Encounter1 and pick the Skeleton Warrior. Encounter1 now has the two skeletons and believe us it is ready! It's this simple! Do the entire procedure again to create Encounter2 with one Giant Scorpion.

Let's create Encounter3 and build our boss from scratch. Well, not from scratch, because we are lazy, but we will tweak it a bit, and illustrate a lot of creature properties. Place the Mummy in Encounter3. If you had any problems, or are just plain stuck for whatever reason, Simple Tome – 6 has it all up to now, so you already know what needs to be done.

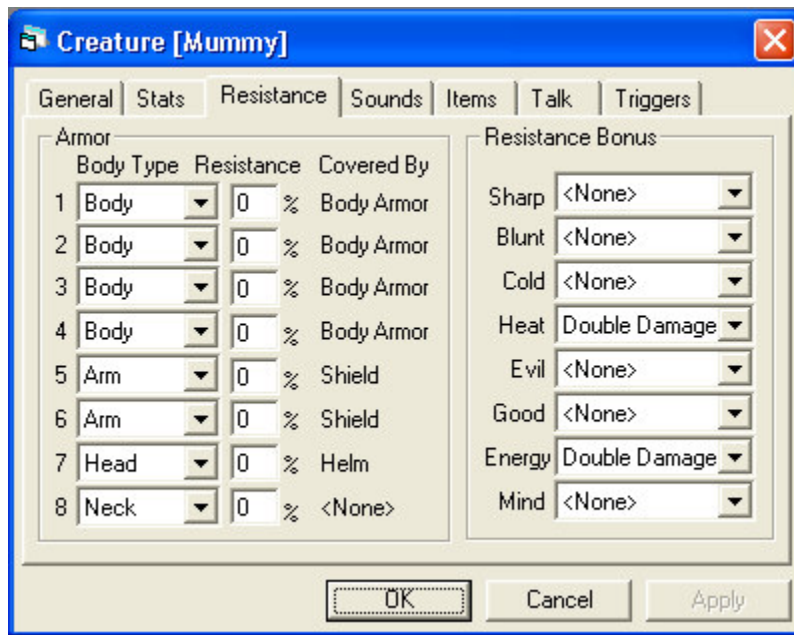
Familiarize yourself with pages 41, 42, and 43 of RuneSword II User's Manual, after you double-click on the Mummy, under Encounter3, under Encounters, under Dungeon in the Area Window (you see, it all flows naturally), and the following window will be available:



Notice the differences: no Help button below, and two new check boxes under "Comments". Well, really one, because "Computer Controlled?" moved down to a better placing (thanks, Adam). The "Inanimate?" check box renders the creature... inanimate. Think about Sleeping Beauty, or the monster of Frankenstein and you'll see some of the uses this property may have, much like the "Is Active?" property of Encounters, above.

We will "soften up" the Mummy a bit, so reduce its ExpLvl to 3 and ExpPts to 300, and click the Stats tab (you can click Apply as you make changes to see them reflected, but only OK will save it all and close the window).

This tab also shows two differences: Eternium is gone and there's no Help button. We will introduce the changes here like in the picture below to double the damage, if attacked and hit by Heat or Energy (spell casters realms, or magic weapons):



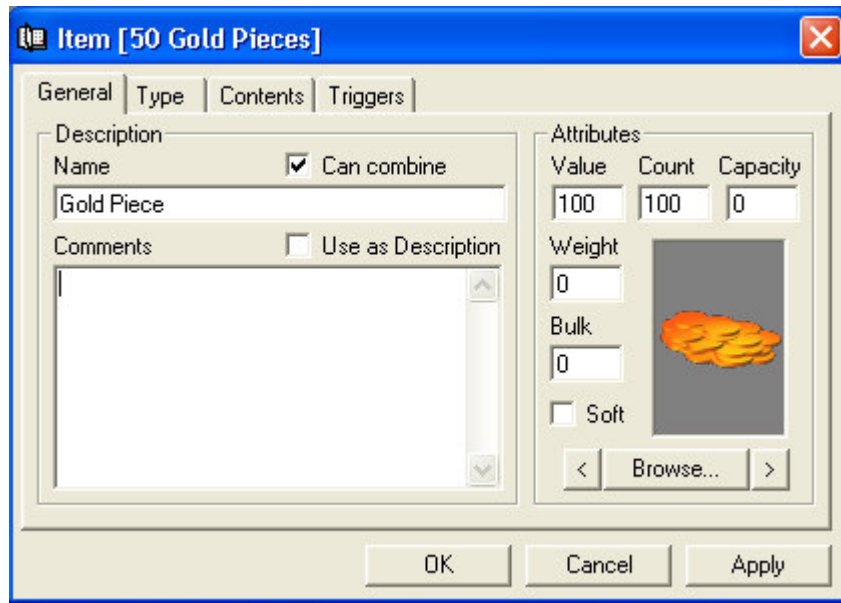
No changes in the next two tabs, Resistance and Sounds, are to be applied.

In the tab Items, you'll be able to see that our mummy carries a Hard Axe. You can check it out through the "Edit..." button, if you wish. The next tab, "Talk", is where you place conversations to interact with the characters, but you can also do it through triggers. It's blank, which seems appropriate for a fully bandaged creature.

Finally, there's the "Triggers" tab, where two triggers have been defined. The mummy brains decides if it will use the axe or use the other trigger, that summons skeletons to come into battle the party, if runes are available and if there are less than 5 characters (both players and non-players) in the Encounter.

The triggers use a simple language called RuneScript, which is out of the scope of this tutorial. Click OK, to close the editing of the mummy.

Finally, we want to add some treasure to the Encounter. Right-click Encounter3 and select (Insert Item from Library..., go to "BasicSet", then "Items", and pick "50 Gold Pieces". Double-click this item to open it, and use this opportunity to check pages 44 and 45 of the User's Manual. Make the following changes, click apply to see them taking effect, then OK.



Now it is time to save your adventure, and test it. You'll find this stage complete in the Simple Tome – 7 folder.

This concludes the first part of this tutorial.

Linking maps

In this second part we will link the maps that I have created. You'll need to copy the contents of Simple Tome – 8 to your Simple folder, under Tomes.

Start Creator and load the Tome. As you can see, there are three maps shown in Area1. Do check them out, before continuing, to get a grasp of the layers used in their painting.

As we've seen before, maps link through the use of Entry Points, which can double as exit points, but while up to now we required a single entry/exit to link the Dungeon to the RuneSword II menu, we shall need a total of five Entry Points to link everything: on the Village surroundings map we need one to link to the menu, one to link to the Trading Outpost, and one to link to the Dungeon; On the Trading Outpost we need one to link to the Village surroundings, and finally one in the Dungeon to also link to the Village surroundings.

Remember this: Entry Points work in pairs, with the exception of the one linking to the menu, and even that needs to be defined also at the Tome properties level, so it's debatable (tele-transportation is implemented as a trigger in an Encounter with the MoveParty statement).

Now we shall link the different maps:

1. Open the Village surroundings map, and place entry points at x= 0, y=9, x=11, y=8, and x=2, y=8, naming them as Main, Dungeon and Outpost, respectively. A proper description should

be included into each like “Exit to the Menu?”, “Enter the Dungeon?”, “Enter the Trading Outpost?”, also in the same order.

2. Open up the Dungeon Map and edit the entry point already defined like the picture below:

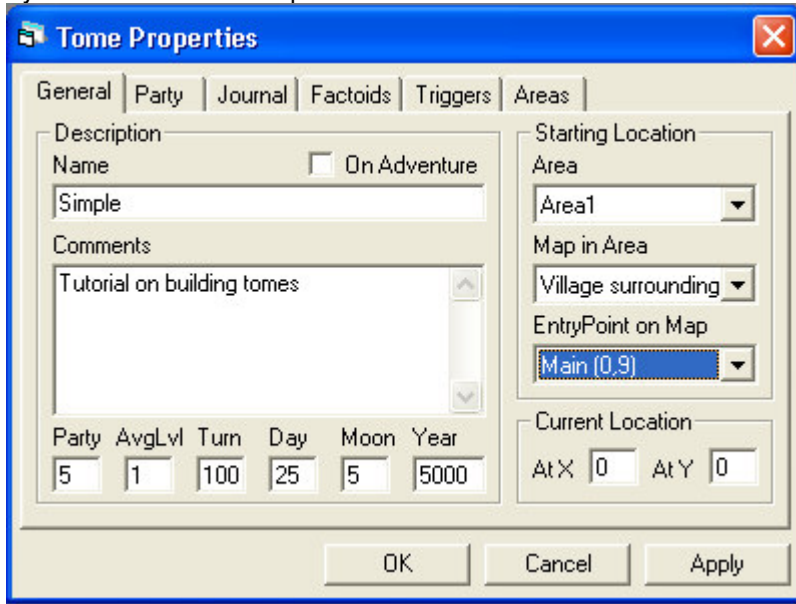
The screenshot shows a dialog box titled "EntryPoint [Dungeon]". It has a "General" tab. The "Description" section includes a "Name" field with "Dungeon", a "Hide Exit Sign?" checkbox (unchecked), and a "Description when enter/exit" text area containing "This is the entrance to the dungeon. Do you wish to exit?". The "Style" section has two dropdowns: "Exit" and "To Wilderness". The "Location on Map" section has "At X:" set to 1 and "At Y:" set to 8. The "Links To" section has "Area" set to "Area1", "Map in Area" set to "Village surrounding", and "EntryPoint on Map" set to "Dungeon". At the bottom are "OK", "Cancel", and "Apply" buttons.

Click OK.

3. Open the Trading Outpost map and edit the EntryPoint like the picture shown below, and click OK:

The screenshot shows a dialog box titled "EntryPoint [EntryPoint1]". It has a "General" tab. The "Description" section includes a "Name" field with "Outside", a "Hide Exit Sign?" checkbox (checked), and a "Description when enter/exit" text area containing "Do you want to get outside?". The "Style" section has two dropdowns: "Exit" and "To Town". The "Location on Map" section has "At X:" set to 7 and "At Y:" set to 10. The "Links To" section has "Area" set to "Area1", "Map in Area" set to "Village surrounding", and "EntryPoint on Map" set to "Outpost". At the bottom are "OK", "Cancel", and "Apply" buttons.

4. Finally, go to the Tome properties (double-click on Simple in the Tome window, above right), and finalize your work like this last picture:



After clicking OK, save your Tome on top of Simple, and test it out. You should be able to take a grand tour around the whole setting, but not fight out the Dungeon, but we still need one Encounter to be in place: the trader. This comprises the last part of this tutorial.

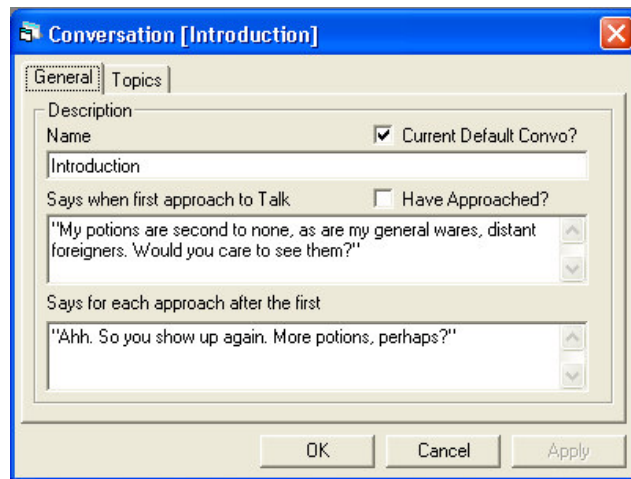
Final Encounter

If you got anything wrong, don't despair, because it's in Simple Tome - 9, which we'll be using to add our final Encounter.

As you surely noticed by now, Creators are lazy and do not wish to re-invent the wheel, so we will use the third file in Simple Tome – 9 as the template for our trader, suitably wrapped up in an Encounter for us to paint, after editing.

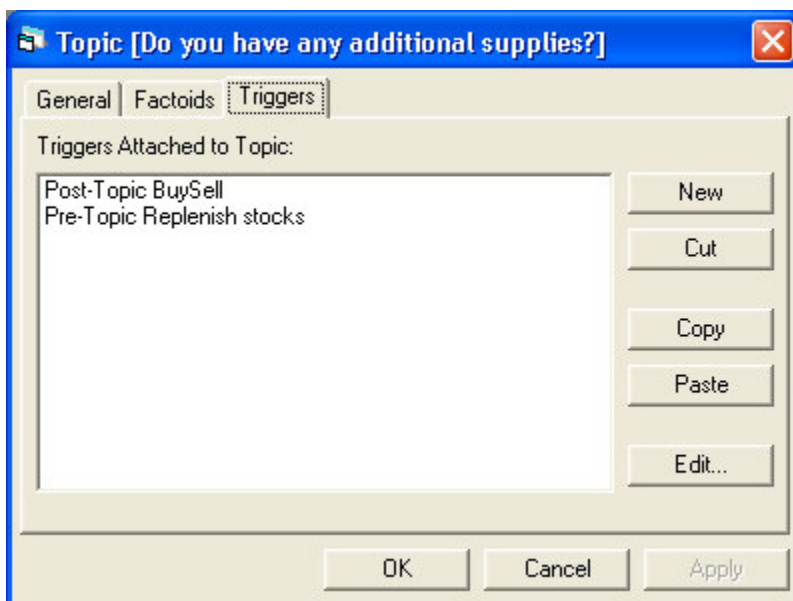
Open up Creator, load the Tome version 9, right-click Encounters, under "Trading Outpost", under the "Area1" window and load "The Trader". Double click on it (the Encounter, not the creature), and notice the check on the "Can Talk?" flag, as well as the two dialogues. Since this is self-explanatory, let's move on and either open the creature named "The Trader" by selecting the tab "Creatures", then the trader and finally "Edit..." or via the Area1 window. By now, you should be proficient in these navigations, so we'll leave them be.

The only noticeable thing is under the “Talk” tab, where one default conversation, named “Introduction” is. Go ahead and edit it to see the following picture:

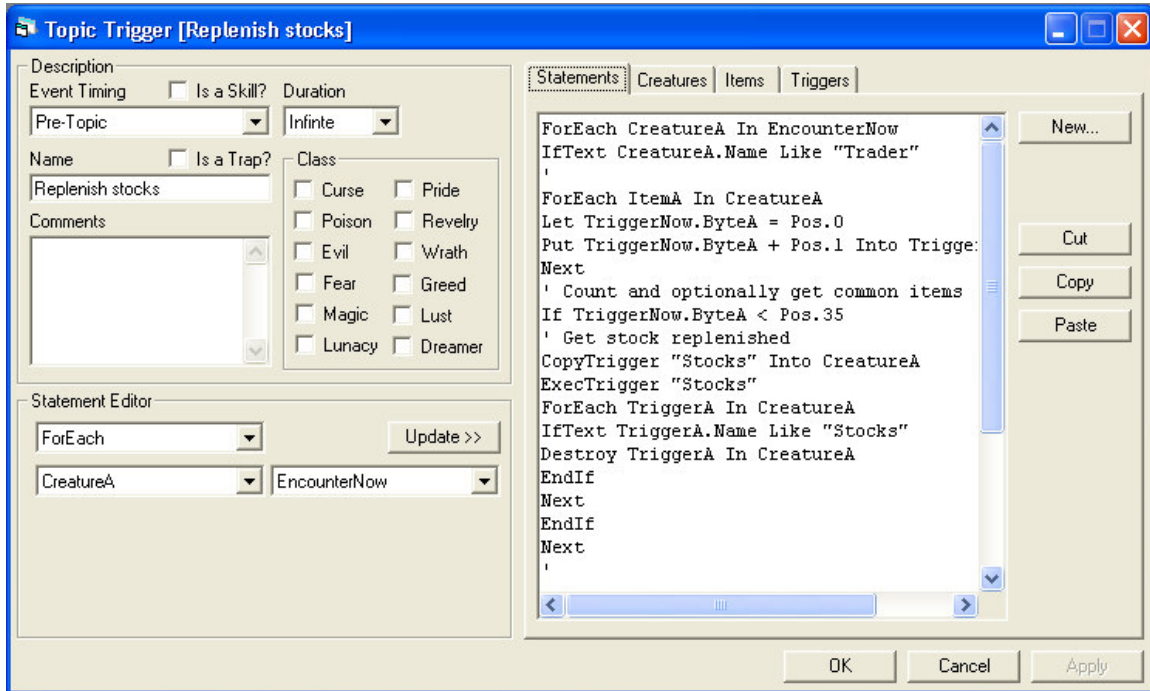


It's obvious we've used a potion merchant from “The Wild”, a Tome we've posted some time ago, suitable for generating random adventures, but in this one we want him to trade whatever goods may be required. Go on and check the “Topics” tab to see how the conversation goes. Note the “Triggers” tab contents, reproduced below for your convenience.

There are two triggers, one “Pre-Topic” and one “Post-Topic”. The first fires up and executes before this topic is used, and its purpose is to handle the trading process, while the second fires only after the topic is shown in play and its objective is to replenish the stocks our trader has to offer. If this is clear to you, let's check the trigger called “Replenish stocks”:



For your convenience, the window of the trigger is reproduced below:



On the left side of the window, you get all the important properties of a trigger, but some details are worth mentioning, besides what you can see from page 54 from the RuneSword II User's Manual.

Notice that one of the things not readily apparent is that you have different types of triggers: the one above is a "topic" trigger, the one in page 54 is a "creature" trigger. Trigger types restrict the RuneScript statements you may have available for that trigger. Also, the whole window is not resizable (yet, we hope), so you need to scroll up and down your RuneScript window, and ensure your cursor is properly positioned on the line you're working. Here's the full trigger, indented at each structure, which we will comment(by putting a ' symbol in front of the line) to give you an idea of how things work:

ForEach CreatureA In EncounterNow
IfText CreatureA.Name Like "Trader"

- ' This searches for "Trader" in this Encounter. The ForEach Statement
- ' Means that it'll look for this text string in every Creature.
- ' This includes characters in the Party, btw-
- ' In "The Land Beyond" many such strings look for the characters
- ' That are part of the Party, but who also serve a plot purpose.

```

ForEach ItemA In CreatureA
    Let TriggerNow.ByteA = Pos.0
    Put TriggerNow.ByteA + Pos.1 Into TriggerNow.ByteA
Next

```

' The code above counts the number of objects that the Trader
' Possesses, and then adds objects until the conditions below are met.

```

If TriggerNow.ByteA < Pos.35

```

' If the number of objects the Trader holds is < 35, then the script above activates
' The script that follows below:

```

    CopyTrigger "Stocks" Into CreatureA
    ExecTrigger "Stocks"
    ForEach TriggerA In CreatureA
        IfText TriggerA.Name Like "Stocks"

```

' The "Stocks" trigger puts semi-randomized sets of common
' Items into the Trader's inventory, so that he has something to sell your Party.

```

        Destroy TriggerA In CreatureA

```

' Destroy sounds ominous- but what it does is not. Destroy clears out a variable.
' This means that the variable TriggerA, which many RuneScripts make use of,
' Is no longer being used, and has a value of 0. If you have a variable that needs to
' Last longer than one Encounter, you'd be wise to use a Factoid or GlobalTrue statement
' Which can be read by Triggers throughout your world. More information on the uses of
' Factoids can be found in the more advanced Runescript tutorials.

```

        EndIf

```

```

    Next

```

```

EndIf

```

```

Next

```

```

EndIf

```

```

Next

```

```

ForEach ItemA In CreatureA
    If Random.5 = Pos.1

```

' "Random.5" means "a number between 1 and 5". So, if a random number between
' 1 and 5 = 1(the "Pos.1" in the script) then the script below is activated:

```

        Destroy ItemA In CreatureA

```

' Once again, we see the Destroy argument being used, but in a different context.
' As you can see, Destroy is used whenever we need to get rid of something-
' In this case, we're using Destroy to erase an ItemA(which would be the first item out of
' The random list of items picked out by the Stocks Trigger referred to up above.

```

        EndIf

```

```

    Next

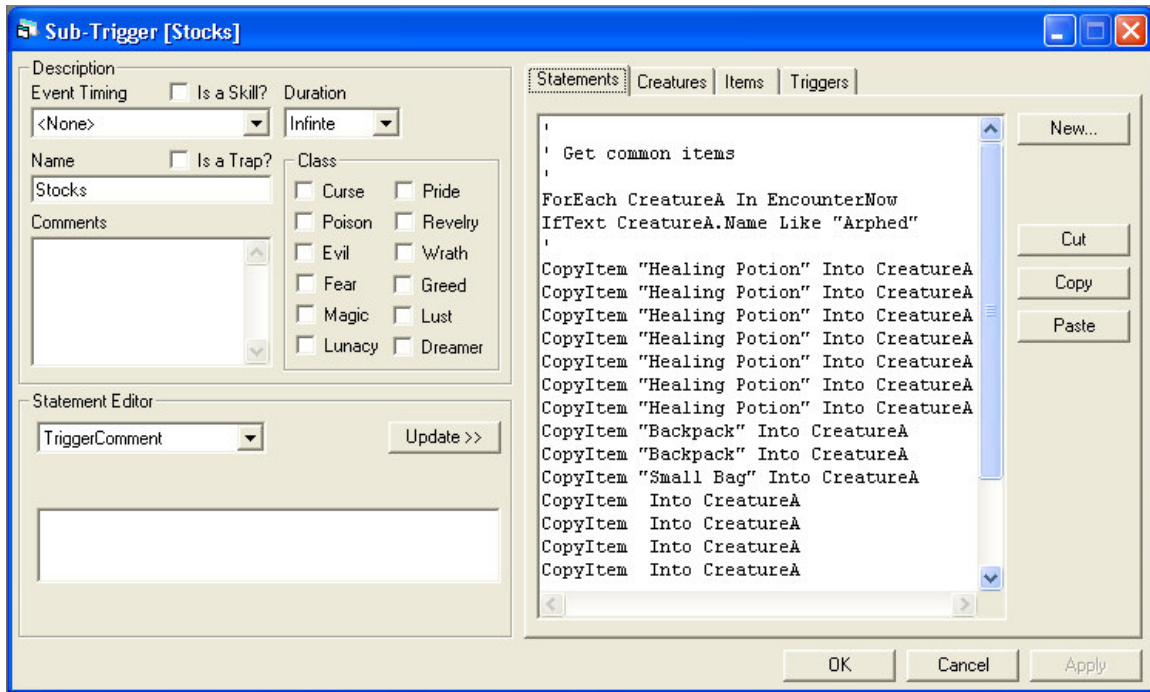
```

There are 2 blocks in the trigger, delimited by the ForEach...Next statements: the first one, the biggest, looks into each creature in the current Encounter for one named "Trader" (the second statement – IfText). When CreatureA.Name is "Trader", that is, when the loop finds our trader by the name, it goes into the next ForEach...Next loop, and counts all the items the trader has into ByteA of the current trigger; if the trader is currently carrying less than 35 items, it copies and executes a sub-trigger that replenishes the stock of items, which is afterwards destroyed. Lastly, there's another loop that destroys each item the trader has 20% of the time, allowing for some randomization of the goods carried.

If you got this easily digested, congratulations! You've probably noticed also that, if there's a guy in your party also named "Trader", you're bound to have problems, because the trigger will run for that guy as well, but how likely is it?

Notice I mentioned something called a sub-trigger. They are discussed in pages 126-129 of the User's Manual. The only thing I want you to understand at this point is that for a trigger to call upon a sub-trigger, this sub-trigger must already exist, otherwise the Statement Editor cannot find it, much like the EntryPoints way of things discussed earlier.

Let us check the sub-trigger, because it needs editing.



As you can see, there's no Event Timing. Also, who is this Arphed bloke, and those CopyItem statements without an item to be copied? Arphed was a leftover I forgot to edit when I posted the Wild (the Encounter was based on a NPC from Eoria, another large adventure I'm creating).

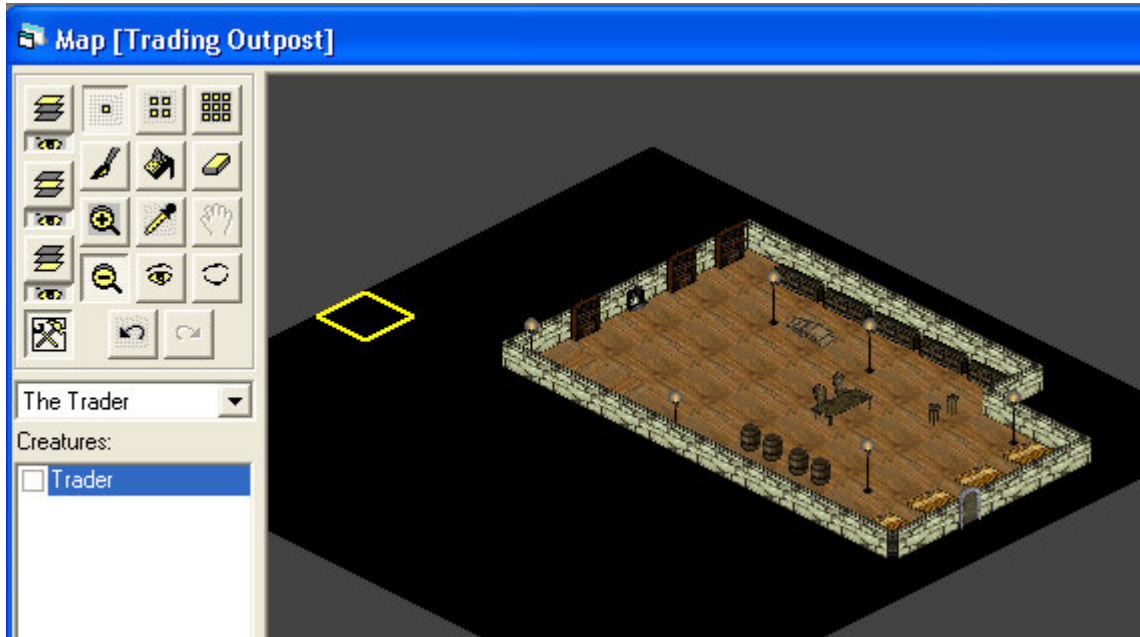
Edit "Arphed" to "Trader" and change the CopyItem statements to whatever items you want to add to our trader's display of goods. Note that the trigger must contain one copy of each object you want to place in your CopyItem statement, therefore you need to use the "Insert Item from Library..." option first.

To save you time and trouble, you'll find a copy of the RuneSword II Library we have put together in the Simple Tome – 9 folder. This has a whole bunch of RS II stuff you will want to use and explore for your Tomes, structured in an easily referenced hierarchical structure.

You can save your work after this task or use the Simple Folder – 10 for the painting of the Encounters.

To paint our Encounters on the map we shall use the facilities explained on page 47 of the RuneSword II User's Manual, so prepare the contents from our tenth folder into our Tome and open Creator 2, using it.

Double-click the Graphic of the Trading Outpost and select the "Paint Encounter mode", by clicking the appropriate button, and you should get something similar to this (I've zoomed out before capturing the image to make for a smaller picture, so don't despair if the size of your picture is bigger).



Note that you "paint" your Encounter like a tile, so the Encounter area can be as small as one tile, or as big as the room or area you're using. The picture of the Encounter contents, items or creatures, will show up on the tile where you select the tick box under "Creatures:" or "Items:".

If you check the Dungeon map, the other Encounters are already in place. And that's it! In case something was lost in the process, the full mini adventure is in the Simple Tome – 11 folder.