

Table of Contents

RUNESWORD II GUIDE	3
QUICKSTART	4
ETERNIA.....	5
<i>Background</i>	6
<i>A Brief History Of Eternia</i>	7
<i>Races</i>	9
<i>Abilities</i>	11
<i>Vices</i>	12
<i>Skills</i>	13
<i>Becoming Famous and Feared</i>	16
<i>Creating Characters</i>	17
EXPLORING	18
<i>Taking Stuff</i>	19
<i>Opening Doors and Chests</i>	20
<i>Talking with Folks</i>	21
<i>Inventory</i>	22
<i>Using Skills</i>	24
<i>Options</i>	25
FIGHTING	26
<i>Weapon Attacks and Other Choices</i>	27
<i>Action Points</i>	28
<i>Defense and Armor</i>	30
<i>Spellcasting</i>	31
CREATOR GUIDE.....	32
STEP BY STEP: CREATING A TOME	33
THE BASICS.....	35
THE TOME AND AREA WINDOWS.....	36
DIRECTORIES FOR STORING STUFF.....	37
ENCOUNTERS	38
ENTRYPOINTS	40
CREATURES.....	41
ITEMS	44
THE MAP WINDOW	46
<i>“Paint Graphics” Mode</i>	46
<i>“Paint Encounter” Mode</i>	47
MAPS	48
THEMES	50
TILES	51
DEFAULT COMBAT WALLPAPER (SCENERY).....	52
TOME	53
TRIGGERS.....	54
SEARCH AND REPLACE.....	55

ENCYCLOPEDIA ARCANA	57
PROPERTIES	58
<i>Introduction.....</i>	58
<i>Creature Properties.....</i>	60
<i>Item Properties.....</i>	66
<i>Trigger Properties.....</i>	70
<i>Tile Properties.....</i>	72
<i>Encounter Properties.....</i>	73
<i>Tome Properties.....</i>	75
<i>Map Properties.....</i>	76
TRIGGER EVENTS	78
<i>Introduction.....</i>	78
<i>Creature Trigger Events.....</i>	81
<i>Item Trigger Events.....</i>	88
<i>Encounter Trigger Events.....</i>	97
<i>Dialog Trigger Event.....</i>	101
<i>Tome Trigger Events.....</i>	103
RUNEScript	105
<i>Introduction.....</i>	105
<i>Basic Statements.....</i>	106
<i>Combat Statements.....</i>	111
<i>Dialog Statements.....</i>	115
<i>Miscellaneous Statements.....</i>	118
<i>Weird Stuff in RuneScript.....</i>	122
SUB-TRIGGERS.....	126
<i>Giving Delayed Effects to a Target.....</i>	127
<i>Giving Long Lasting Effects to a Target.....</i>	127
<i>Executing Sub-triggers for Convenience.....</i>	129
VARIABLES	130
<i>Local Variables.....</i>	130
<i>Global Variables.....</i>	132
ELEGANCE IN DIALOG.....	135
ENCOUNTER REACTIONS.....	138
<i>Talk, Fight, Flee, Ignore, and Search.....</i>	139
<i>Computer-Control, Brains, and Friendliness.....</i>	139
LOCKS.....	141
<i>Locked Doors.....</i>	141
<i>Locked Containers.....</i>	143
TRAPS	145
<i>Traps on Items.....</i>	146
<i>Traps in the Area.....</i>	146
UNDER THE HOOD.....	147
APPENDICES.....	151
APPENDIX A: ARMOR	151
APPENDIX B: WEAPONS.....	152
APPENDIX C: SPELLS	153
APPENDIX D: SKILLS AVAILABLE TO RACES	156
APPENDIX E: DESIGNER NOTES	157

RuneSword II Guide

RuneSword II is actually two programs. RuneSword II is the game itself, and Creator is the construction set. This part of the docs is on playing RuneSword II; the rest is on the Creator.

A Player If you're interested in playing, not creating, you needn't bother with most of the *Creator Guide* or the *Encyclopedia Arcana*, though you'll probably find parts of them interesting. But in the *Creator Guide*, read the section on the Tome Wizard feature, which enables you to automatically generate new adventures.

I Create, Therefore I Am If you want to make role-playing games, or help others do so, you've come to the right place. Creators specialize several areas: map layout, storytelling, dialog, creating additional graphics and sound, and runescript. You don't need to master it all -- the component nature of RuneSword II enables others to pitch in. Start by playing a lot, watch the tutorial AVI videos, and then you'll be ready to get into the Creator docs.

If you have questions that aren't answered here, go to the RuneSword II web site at www.runesword.com. There's also free stuff there for players and creators.

RuneSword II Guide Shows how to play, character creation, and explains some aspects of the world of Eternia.

Creator Guide Basic stuff on the interface of the creator. Also shows the property sheets for creatures, items, and other "objects."

Encyclopedia Arcana The reference work for creators. Here you can look up a particular event, property, or a RuneScript command. The discussion of events and triggers is must reading.



In Eternia: A great ambidextrous swordsman, Fafnir Gudrun was the favorite of Tul'Shar. Once he acquired Hord's Sword of Rage and Sorrow, however, he became a legend.

QuickStart

Before Playing

Select *Characters* from the main menu. Create at least 3 characters.

Select *New Tome* from the main menu and choose *The Land Beyond* as your first adventure tome to play.

During Play

Travelling the lands, left click to move somewhere. Click the *Actions* button at the bottom of the screen to select an action for the active character (the highlighted portrait).

Click a different character portrait to change the active character. Click the *Action* button and *Inventory* for each character and make sure they're properly equipped.

In combat, click a space to move there or click an opponent to attack. Right-click to perform the character's special action.

Skim the pages in this RuneSword II Guide as you play.



Are you going to fight or read docs?!

Eternia

The setting for the included tomes is called Eternia. Other tomes may or may not be set there (they don't even need to be fantasy).



Map of Eternia

Background

Eternia was once highly civilized, with intriguing machines and ancient magic. Then the Great Calamity occurred in the southern region. Some people left the idyllic land, now known as the Land Beyond. Those that stayed died or were transformed. The knowledge of many machines was lost. Magic was debased into exploiting vices for power. In the current age, there remain a small group of spellcasters called Dreamers (or more derisively, Fish) that retain the old way of magic, but these albino mutants are despised with some justification. The 12 kingdoms, dominated by Jafra, fight many border skirmishes, occasionally murder ambassadors, but are not at war.

Time There are 200 turns in a day. Turns 1-20 are darkest night.

Skies and Stars The planet circles binary stars. One is named Mother Ibeza; the other simply called the Babe. There are four moons, Alern, Adad, Elohin, and Zylos, collectively known as the Brothers. They are associated with war and strife and chaos. Meteor storms are common on Eternia.

Castes

- Royals
- Soldiers (military and entourage of the Royals)
- Merchants (business owner)
- Makers (musicians, carpenters, farmers, etc)
- Slaves and criminals

Adventurers are somewhat outside the caste system as long as they have a reputation. The most famous adventurers are celebrities in some places, but most adventurers would be equivalent to Merchants or Makers in reputation. Dreamers are regarded as the very worst criminals.

The caste system plays into the laws of the 12, kingdoms, especially Jafra. A crime against someone of low caste, especially if done by someone of high caste, doesn't carry much of a penalty.

The 12 Kingdoms Each kingdom is dominated by a humanoid race with differing skills and abilities. These differences have grown more pronounced as the years pass. This process, known as "The Withering," is discussed by scholars in private, but is unknown in the wider world. There is, however, a general sense that much knowledge was lost at the close of the Age of Reason, over 700 years ago.

The Land Beyond A place beyond civilization in the far south, just beyond the knowledge of mapmakers, is a place called the Land Beyond. As the land of origin for all people, it still contains a few artifacts from its bygone glory days. It also houses many mutant denizens and creatures from the Cataclysm, whose power and hostility prevent entry from all but the most determined explorers. That would be you.

A Brief History Of Eternia

Please note that the following is NOT a synopsis for the first tome (adventure); it is what can be referred to as a "back story." With most computer RPG's, there is no distinction --- you're going to save the world and kill the big bad guy who has released his orc army to scour the land. Then in the next game release, you have to save the world all over again. Because of RuneSword's multiple adventure nature, however, our modules will connect with this back-story much more slowly. If your character(s) are to affect the course of history of Eternia, they will have played through many adventures.

The Age of Reason - Little is known of the First Age, a time of legends. This much is clear: times were peaceful, civilization was based in a region now called "the Land Beyond," and the enlightened populace used magic and sophisticated technology to bolster an idyllic culture.

In the year 758 First Age, a relic was found in the deepest mine of the realm. It was a plain box, of unknown material, with the image of a hanged man engraved on the lid. Despite warnings from the Court Engineer ("This is a forbidden thing, and should be buried again and forgotten"), the King was inquisitive and slid the lid away with his own hands.

A new race called the Eternals was released. They swept over the land, creating macabre variations on true animals and people, and these Counterfeits attacked and slew the True. Some say the Eternals could change their appearance from young to old, but do not age. They can be killed, though not easily. Only steel or magic can slay them, and that only by destroying their heart. It is said that they watched the first sweep of the hands of time and were imprisoned by Nameless Ones for their affronts until their accidental release. Some Eternals became insane because of their long imprisonment. The First Era ended when the last of the True ones left for a new land. The Eternals remained in the Land Beyond.

The Age of Separation - In the year 321 Second Age, five years of terrifying meteor storms swept across all Eternia. Differences in the physical appearance of peoples began to show, and differences in thinking followed soon after. The Splinter War began, lasted nearly 30 years, and by the end, the 12 kingdoms of today had formed. Jafra Ulshek the Magnificent subdued a great territory in the western reaches of the new realm, an empire that lasts to the current day. Thus ended the Second Era.

The Age of Tumult - In year 192 Third Age, the first Dreamers appeared, now derisively called Fish in slang. Dreamers are mutants, perhaps brought about by the awakening of the Eternals, or somehow linked with them, but none know for certain. All Dreamers have white eyes, white hair, and white skin. Their great powers made them rulers in most of the kingdoms, and for the most part, they ruled wisely. Dreamers were almost as powerful as the Eternals, but they had a life span like that of mortals. The magic of Dreamers was a curious thing, in that it's use had a "backlash" effect that could affect either the caster or others near the caster. In those days, the Dreamer rulers suffered for their art, and built many great cities and monuments. In 310 Third Age, a rogue faction of Dreamers rebelled, led by Thorn the Powerful, and his incredible lieutenant, Tul'Shar. These Dreamers successfully sieged the capital cities of Jaфра, Morea, and Argos within 3 years. Nearly all Dreamers were in the rebel fold by the time they marched on Skandar. The rebel Dreamers did not take the backlash effect on themselves and longer --- they spread it to their enemies. They probably would have conquered the remaining, lesser kingdoms in a few years were it not for the Invasion of the Eternals.

Led by an insane Eternal, the Unspeakable Undernight, the Eternals came in great ships with their armies of hideous mutant beasts. They slew many, including the Dreamer Thorn. The warring factions of Dreamers gathered together, as did the mortals, united against a common enemy and commanded by Dreamer Tul'Shar, the greatest general of the Age. Tul'Shar outflanked and outwitted the Undernight in battle after battle, against steep odds. The Eternals, previously thought unbeatable, were driven back, their remnants sailing back to the Land Beyond.

Tul'Shar, touched by the camaraderie of his fellows in the face of a great enemy, renounced his former ways and vowed to become a suffering Dreamer, to live in the traditional way of the elder Dreamers. A few other rebels also repented, and joined the small group of Dreamers that never left the path of suffering. Before this group had a chance to mount a challenge, however, Tul'Shar was betrayed by his wife and delivered into the hands of an Andorian duke, Fulk the Black. Tul'Shar was hung by his feet from a tree for five days before being cut down and beheaded in a public square in Anna Island. Fulk began the Cleansing Wind, his term for the extermination of all Dreamers. Thus ended the Age of Tumult.

The Fourth Age - As the fourth, unnamed age dawns, the Dreamers are dead or in hiding. Fulk the Black has the ear of the Jafran ruler Suthnic IV, and all mortal rulers are committed to the Cleansing Wind policy. Meanwhile, rumors abound of strange activity in the Land Beyond.

The first tome, *The Land Beyond*, begins in year 16 of the Fourth Age. A small group of friends, one of them hunted, is sailing south to avoid the kingsmen...

Races

Choosing your race is important in RuneSword II. Many years ago, after the exodus from The Land Beyond into the 12 Kingdoms, differences between the peoples began to emerge. This mutating effect has accelerated over the 350 years since the end of the Age of Separation.

The world is largely segregated. Most people live their lives in their land of birth. And though all peoples share an ancestry that is still spoken of by bards, there is a tremendous amount of racial strife. Moreover, there are castes within most countries. However, the characters you play are *travellers*, not royals, nobles, merchants, peasant, or slave – you are beyond caste, whatever land you're from.

Travelling adventurers do not hew to social tradition. Small bands of travellers will have Jafran with Morean, or Kyparissian with Argosan, or even more outrageous pairings. They depend on each other to survive in hostile places. The goods ones live well, but few live long.

Northlander - The Northlands is a place of rangers, hunters, and trackers - masters of the forest. Northlanders have a reputation of being untrustworthy and eccentric. Of course, they are still hired when royalty goes on hunting expeditions. The Northlands is run by a council of Elders.

Xal - Xal is a sparsely populated mountainous region of assassins, miners, and hermits. An enormous, intricate, never-mapped web of tunnels runs through the mountains. Xal is ruled by close-knit clans of associated families, each run by a matriarch.

Islander - Islanders are giant people. Those from Big Isle are usually good blacksmiths, those of Bit Isle tend to be jewelers. Despite their horrific appearance, Islanders can create items of surprising delicacy and beauty. Giants tend to be unfriendly to passing acquaintances. They have a natural 20% resistance to cold-based attacks.



In Eternia: Damien the Xal slew a lich single-handedly, but he would not speak of it. Damien's favorite foe was a lock.

Gaul - Gaul is the land of thieves and pirates, as well as a few shipwrights. Their boating expertise, however, tends toward the small and quick sloops and pinnacles, the better to terrorize the merchant marine of other lands. There is no recognized government in Gaul.

Talantan - Talanta is a harsh, barren land of unpredictable climes, and as such, serves as a perfect training ground for young Talantans. The children are all unwanted orphans recruited from other lands, who must undergo a sterilization and purification ceremony upon acceptance into a Talantan commune. Other races refer to Talantans as Myth Walkers, because their goal is to become such a famous warrior that bardic tales are sung in their memory. Whether they die or not at the end of the tale is unimportant.

Andorian - Andoria is the land of merchants and cooks. Their meals are legendary, almost as much as their shrewd dealing. Andorians have a reputation for good humor and are fun to be around. They often take a long-term view that leads to ultimate success.

Jafran - The major power in the world of Eternia, the Jafra High Court - ruled by Emperor Suthnic IV - exacts tribute from most of the other regions. Jafra's commercial and political interests, infighting, and machinations are second to none. The Emperor's intensely loyal personal army is known as the kingsmen.

Skandar - Skandar is a land of great sweeping plains. This is also a land of entertainers, horse tricks, circus performers, and the like. Fiddler's Bay has a particularly large concentration of artistic types. The ruler of Skandar is Elanor, Emperor Suthnic's sister.

Zurish - The Zurish are fishermen, merchant sailors, and master ship builders. Physically, they are very short and stocky. They are critical to the transport of goods all over the realm. Their strangely hardened skin is 30% resistant to fire and 20% resistant to edged weapons.

Kyparissian - Kyparissia is the land of scientists and explorers, hardheaded men and women who are legendary for their rudeness, or "blunt honesty" is you are kind. They feud with the Argosans, who are of dissimilar temperament.

Morean - This war-banding, clannish, crazed pack of heavily tattooed marauders does not answer to anyone. They hate the Emperor and do not pay tribute. If it weren't for the great sea between Morea and Jafra, they would have wiped each other out.

Argosan - The Argosans are mystical monks who reject the legends of the Enlightened Age. They have their own ideas about the relics and writing of that bygone time. Their blue-green skin tone comes from ingesting strange powders and potions in ceremonial rites.

Abilities

There are 3 abilities in RuneSword: Strength, Agility, and Intelligence. Each affects how your character performs both in and out of combat. The average person has a 10 in each, the higher the better. Heroic characters average about 14, but this varies due to the character's race. These abilities are not randomly determined.

- *Strength* gives a +1 to damage for every 3 points of Strength above 10.
- *Agility* gives a +1 to attack rolls for every 3 points of Agility above 10. Agility may also increase your quickness in combat.
- *Intelligence* reduces the chance for spells fizzling; often used for situation tests.

Occasionally, there are situational rolls against an ability. For example, a character may have to roll a 20-sider against his Intelligence score to see whether he notices a pit. If his Intelligence is 14, he must roll at 14 or less to succeed. Depending on the situation, the success number may be lowered for especially difficult situations (Intelligence -3 for an nearly invisible trip wire) or raised for easy ones (Intelligence + 5 for a small switch).

Other Racial Differences

Health represents life force. Everyday people have 10 or so Health, but heroic adventurers have more. "Death" occurs when health goes to zero or below, but this is a heroic "not quite dead" from which it is possible to return. Usually. Most enemies stay dead when killed.

Size affects Defense (how often you're hit) against attack rolls. Larger creatures are easier to hit. Of the 12 races, 10 are medium sized, so attackers need a 13 or better to hit. The small Zurish have Defense 15, and the giant islanders have Defense 11. Enemies come in other sizes as well.

Race	Health	Strength	Agility	Intelligence	Size
Northlander	40	12	18	17	Average
Xal	35	10	18	10	Average
Islander	45	20	10	11	Large
Gaul	33	12	15	9	Average
Talantan	38	16	12	12	Average
Andorian	35	11	9	17	Average
Jafran	41	13	13	14	Average
Skandar	34	12	16	12	Average
Zurish	38	10	12	13	Small
Kyparissian	35	8	15	16	Average
Morean	42	15	14	9	Average
Argosan	36	12	10	17	Average

Vices

The six vices of RuneSword are Greed, Wrath, Pride, Lunacy, Lust, and Revelry. There are variances based on the character's race. Vices apply only to sentient creatures and characters, not animals.

Good Guy? Note that characters with high vice amounts are not considered "evil," just as those with low amounts are not considered "good." It's more of an issue of self-control. The D&D concepts of Good, Neutrality, Evil, Law, and Chaos are not reflected in RuneSword, or at least not statistically. Of course, there will be consequences for your actions. Not all evil acts are punished, nor all good acts rewarded, but stuff happens. There is karma and chaos in the lands.

Situations Like Attributes, Vices may come into play on certain role-playing situations. The classic example is an attractive member of the opposite sex attempting to entice a character into doing something unwise. Vices foster role-playing in situations where a character may act differently than a cautious player. The player still has control, but the vices inject a bit of Fate into the proceedings.

Spell Resistance Vices also affect spell resistance. When you are targeted by a hostile spell, your chance of resisting it is based on your vice score in that school. Average in a vice is 15% resistance.

Race	Lunacy	Revelry	Wrath	Pride	Greed	Lust
Northlander	Weak	Weak	Great	Good	Average	Average
Xal	Average	Average	Weak	Good	Weak	Average
Islander	Great	Good	Uncontrolled	Uncontrolled	Weak	Weak
Gaul	Average	Average	Average	Average	Weak	Good
Talantan	Superb	Superb	Superb	Uncontrolled	Superb	Superb
Andorian	Good	Average	Good	Average	Good	Average
Jafran	Average	Good	Average	Average	Weak	Average
Skandar	Weak	Uncontrolled	Good	Weak	Great	Uncontrolled
Zurish	Good	Good	Good	Good	Good	Great
Kyparissian	Good	Good	Uncontrolled	Uncontrolled	Average	Good
Morean	Weak	Weak	Uncontrolled	Uncontrolled	Great	Good
Argosan	Weak	Good	Weak	Weak	Average	Good

Creator Note on *Spell Vice Resistance*

In the Creator program, you see the actual numbers behind these text ratings: Superb (15 or less), Great (16), Good (17), Average (18), Weak (19), and worst of all, Uncontrolled (20+). To succeed on a d20 spell resistance roll, the target must equal or exceed the his vice value of that spell's school. Great spell resistance is 25%, for example, while Weak spell resistance is 10%.

Though some artifacts and spells may temporarily change vice values, they do not change through character advancement. Extraordinary events could permanently alter vices, however.

Skills

Starting characters receive 60 skill points to spend, but not all need to be spent during character creation. Skills are either "Active," which means that they are always in effect, or "Choose," which means they must be selected as an action.

When purchased, skills are at Novice level. Skills may be increased at any time, as long as the character has sufficient skill points. The levels are Novice, Adept, Expert, Master, Master +1, Master +2, and so on. Some skills may not be increased beyond a certain level.

Appendix D shows the starting skill choices for the races.

Name	Cost	Type	Max Level	Description
Acrobatics	7	Active	Infinite	Special abilities in tumbling and jumping. +1 agility each time you increase acrobatics.
Ambidex	16	Choose	Infinite	Enables you to wield 2 one-handed weapons at the same time. The 2 nd weapon costs no action points and has a +2 to hit per skill level. Use Ambidex to designate the 2 nd weapon; this choice lasts until you unequip the primary weapon. If you get this skill, don't bother with shields.
Appraise	4	Choose	Master +1	Assess value of any item. Accuracy increases per skill level, and at Master +1, it's spot on.
Arcane Lore	5	Active	Infinite	Gives a bonus to Intelligence rolls concerning history and magic. Also gives resistances to mind and evil-based attacks at +10% per skill level.
Athleticism	12	Choose	Infinite	Gives +4 to action points 2 times per day. The bonus increases +2 and gets 2 more uses per skill level. Also gives bonus to jump, swim, and climb rolls.
Axe	25	Active	Infinite	Gives +2 to hit and +2 to damage per skill level with axes.
Backstab	14	Choose	Master +5	Give 5% chance of instantly killing foe on a successful hit with a dagger, but uses all of the attacker's action points. You get a +5% chance per skill level.
Bardsong	18	Choose	Master	You must possess a quality musical instrument, though it need not be in hand. Twice per day, this gives +5 action points for all party members for 3 turns. Additional levels enable you to use Bardsong 2 more times per day.
Battle Armor	11	Active	Master	This skill is a must for wearing body armor because it negates body armor's action point penalty. Novices can wear leather, Adepts can wear scale, Experts can wear chain, and Masters can wear plate.
Bloodshield	14	Active	Master +6	A miss on an incoming short-range attack gives you a 25% chance of bashing the attacker for 2d8+2 damage with an equipped shield. You get a +5% chance per skill level.
Body Building	7	Active	Infinite	Gives +1 to strength per skill level.
Bow	25	Active	Infinite	Gives +2 to hit and +2 to damage per skill level with bows.
Charge	5	Choose	Novice	Target a foe to rush across room with movement bonus and +4 to hit. Charger suffers a -4 Defense penalty for the next turn.

Name	Cost	Type	Max Level	Description
Cleave	6	Choose	Master	When you kill a foe with a short range weapon, you follow through for an additional “free” attack with +2 Attack. More levels adds +2 to the Attack bonus.
Club	25	Active	Infinite	Gives +2 to hit and +2 to damage per skill level with clubs.
Critical Strike	7	Active	Infinite	Gives +5% chance per level of landing attacks on target’s most vulnerable location.
Dagger	25	Active	Infinite	Gives +2 to hit and +2 to damage per skill level with daggers.
Deflect Arrows	6	Active	Master	You dodge or deflect all arrow attacks.
Evasion	8	Active	Master +2	Gives +1 defense per skill level. You must have 17 agility to raise evasion to Adept level, and 21 agility to raise it to Master level.
Fencing	10	Active	Master +6	When wielding a sword, you have a 20% change of disarming short-range attackers when they miss. You get a +5% per skill level.
Fine Arts	7	Active	Master +1	Diplomacy and music abilities that enable you to engage all levels of society in a pleasant, persuasive manner. You also get +1 Intelligence per skill level. Fine Arts is necessary to use magical instruments.
Greedspell	20	Choose	Infinite	Enables you to cast spells from the Greed school. Hallmarks include treasure recovery, cleverness, and illusions of grandeur. You gain additional spells and increase the power of some spells per skill level.
Healing Hands	17	Choose	Infinite	May heal 1d10 health up to 6 times per day. During combat, you must be in close range to the target. An additional 1d10 is healed per skill level.
Health	15	Active	Master +1	Gives +10 to health and maximum health per skill level.
Heroism	9	Choose	Infinite	Once per day, you can make your next successful hit do more than double damage. You get 1 more use, 1 more heroic hit per use, and a bit of extra damage per skill level.
Leadership				Bonus +1 per skill level to damage to hit rolls for entire party for 3 turns. Can be used +1 times per day per skill level.
Lunaspell	20	Choose	Infinite	Enables you to cast spells from the Lunacy school. Hallmarks include loss of senses, necromancy, and madness. You gain additional spells and increase the power of some spells per skill level.
Lustspell	20	Choose	Infinite	Enables you to cast spells from the Lust school. Hallmarks include enticing deceptions and enlisting aid from others. You gain additional spells and increase the power of some spells per skill level.
Mad Rush	8	Choose	Master	Improved version of Charge still taught by beserkers in Morea. Gives +6 to hit against target. For each level above Novice, it also bestows +3 damage. User suffer –2 Defense for 1 turn.

Name	Cost	Type	Max Level	Description
Martial Arts	28	Choose	Infinite	<p>Short-range attack that costs 7 action points, has a +1 to hit, does at least 4d4 damage, and has a 20% chance of inflicting special martial arts effects.</p> <p>At Novice level, the special effects are simply bonus damage or stunning. By Master level, the special effects include all sorts of results, including instant death. You get a +1 to hit, more damage, and an increased chance of special effects per skill level.</p> <p>When using Martial Arts, you cannot wear armor or hold anything in hand. Note that you must unequip armor before combat begins.</p> <p>Once you click it on your turn, you stay in Martial Arts mode as long as an enemy is in short range (you can just click again on the target to attack with Martial Arts again).</p>
Perfect Hit	6	Active	Master +1	Rolling 10 above the target's defense does more than the usual double damage. Adding more skill levels makes you do extraordinary damage in such cases.
Pick Locks	5	Active	Infinite	Gives +20% chance to pick locks. You get +5% per skill level.
Pridespell	20	Choose	Infinite	Enables you to cast spells from the Pride school. Hallmarks include bravery, intimidation, and increased abilities. You gain additional spells and increase the power of some spells per skill level.
Remove Traps	6	Active	Infinite	Gives +10% bonus on finding and removing traps per skill level.
Revelspell	20	Choose	Infinite	Enables you to cast spells from the Revelry school. Hallmarks include healing, distraction, and immunities. You gain additional spells and increase the power of some spells per skill level.
Revive	13	Choose	Infinite	You can bring "dead" party members back to life once per day as long as the target has –5 health or more. You get 1 more use and can revive characters with 5 less health per level.
Staff	25	Active	Infinite	Gives +2 to hit and +2 to damage per skill level with staves.
Sword	25	Active	Infinite	Gives +2 to hit and +2 to damage per skill level with swords.
Whirlwind	10	Active	Master	Give 15% chance of striking other enemies in close range when you kill a foe with a short range weapon. Bonus 15 % to the chance and +1d8 damage per skill level.
Wizardry	4	Choose	Infinite	Reduces chance of target saving against spell effects by 5% per skill level.
Wrathspell	20	Choose	Infinite	Enables you to cast spells from the Wrath school. Hallmarks include elemental fire, direct damage, and rage. You gain additional spells and increase the power of some spells per skill level.

Becoming Famous and Feared

Experience Levels

The RuneSword Dungeon Master awards “experience points” to characters for excellent play. At the start, accomplishing difficult quests may earn a character 300 to maybe 1000 experience points. For high level characters performing legendary accomplishments, the reward could be 10,000 experience or more. Killing tough opponents also earns experience points.

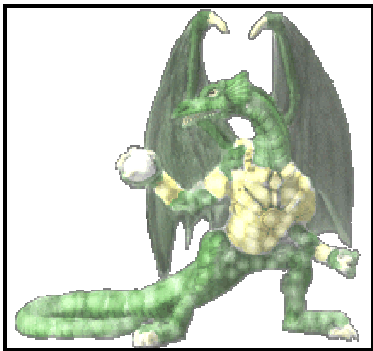
Level	Experience Points
1	0-999
2	1000-2999
3	3000-5999
4	6000-9999
5	10,000-14,999

...and so on in progressively greater increments. Going up an experience level provides many subtle improvements, but the best part is that you get 10 or more skill points to spend on improving your character’s skills. You may be also be awarded skill points for superb play.

Skill Points

Once skill points are earned, they may be spent immediately, or saved for when you collect enough skill points to improve an expensive skill.

Even right after a character is created, if you have enough skill points left over, you could immediately boost a skill’s level. For example, if you made a character with Sword skill (costs 25), and nothing else, you would have 35 skill points left over. You could boost your Sword skill from Novice to Adept at the start of play.

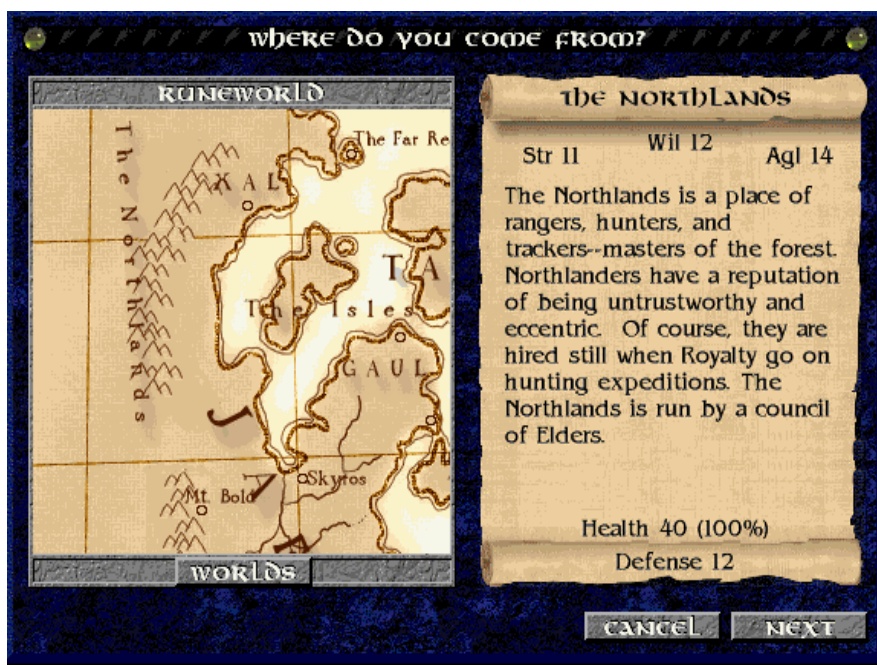


Legend has it that goblins laughed and sang of Chester the spellcasting dragon. Most songs were grotesque commentaries on Chester’s foppish appearance. One day, Chester heard a particularly insulting tune, which is why so few goblins survived to the Age of Reason.

Creating Characters

You may create as many characters as you like. RuneSword allows you to create an almost infinite variety of characters because of the many races and skills to choose from. On the main menu, select **Characters** to start this ball rolling.

Where Do You Come From? Your land of origin determines your character's race (each race comes from a different land). RuneWorld races are humanoid, though appearances vary widely. To select a different land, hold the left mouse button down on the map to "drag" it over. Then click of the name of a land.



What Skills Do You Have? Each race has its own set of skills offered, but there is considerable overlap. You have 60 skill points to spend. Leftover skill points are kept for you to use later, when the character advances a level. Raising a skill 1 level costs the same as the initial cost. So don't worry about spending all your skill points right now. In fact, if you save enough, you could raise a skill right away during play (using the Inventory screen).

What Do You Look Like? Pick a portrait and sex. Sometimes the portraits are specific to a race. Sex doesn't affect your character's stats, but it can affect certain encounters.

Name and Save Your Character Pick one of the suggested names or enter one of your own. Each race also has different types of names. To enter your own name, backspace to delete the current one, and type yours in. Click Save to finish.

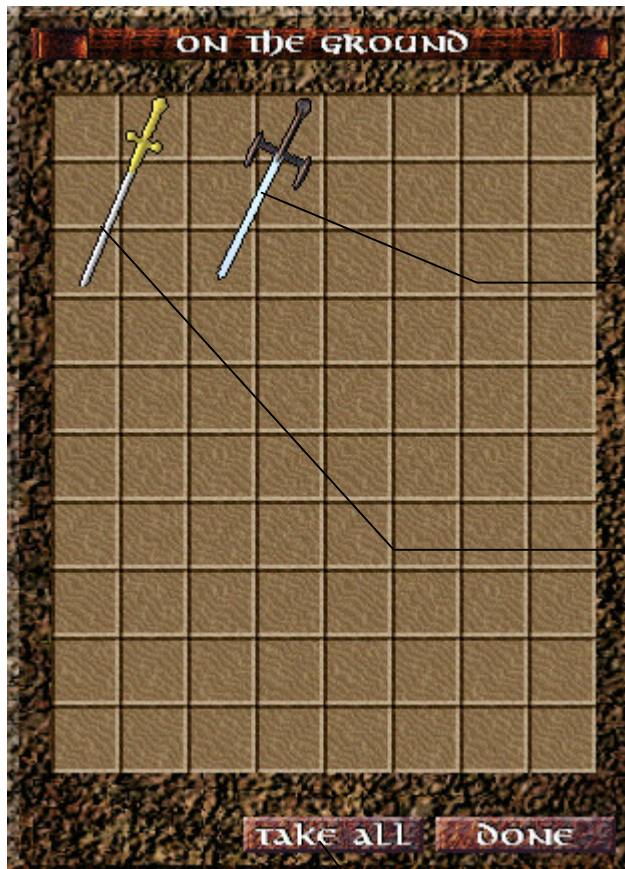
Exploring

Travelling Moving on the map screen is simple enough, you just click anywhere and the engine attempts to pathfind the party there. The party cannot split up. Right-clicking on the map searches in the current location.



Taking Stuff

Search is one of the standard actions. It also occurs automatically after combat.



Hovering the pointer over an item brings up a brief description on the command bar. Right-clicking an item brings up the full description.

To pick up an item, left-click and drag it to a character portrait

Click the Take All button to give all items to the active character. If the active character cannot carry any more, the items are distributed to the other party members.

Opening Doors and Chests

Doors

Party movement stops at a closed door whether it's locked or not. While next to the door, right click to bring up the action menu and select Open. If it's locked, an unlock attempt will occur. If it's locked with a key and the current character is carrying the key -- it doesn't have to be in hand -- the key will automatically be used.

Chests

Locked chests (and any other kind of locked container) show you two options when you attempt to open them: Bash and Pick. If you have any character with the Pick Lock skill, use him to make a Pick attempt. Some unsuccessful pick attempts will result in the lock becoming jammed. In this case (or if you lack a character with the Pick Locks skill), you must resort to bashing the chest open.

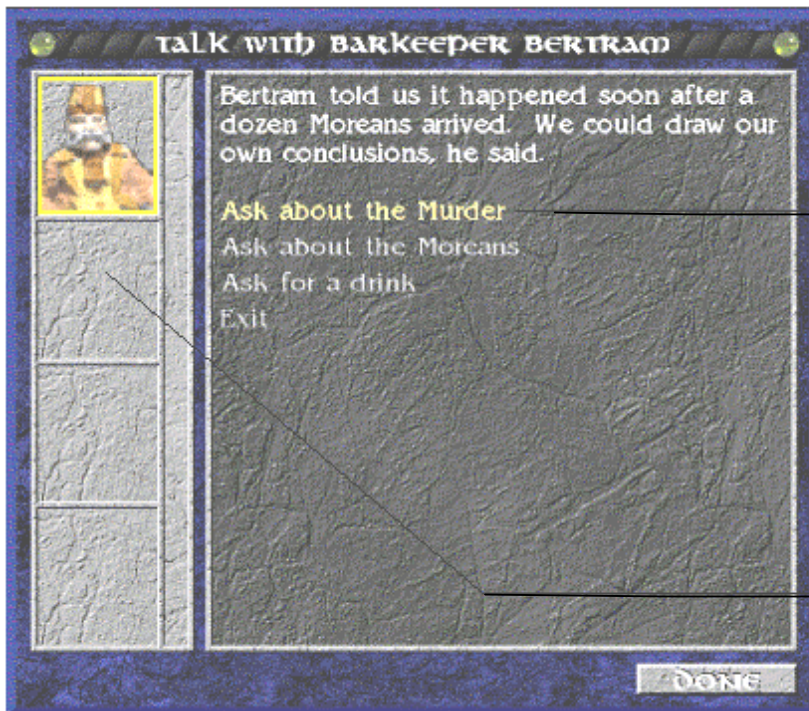
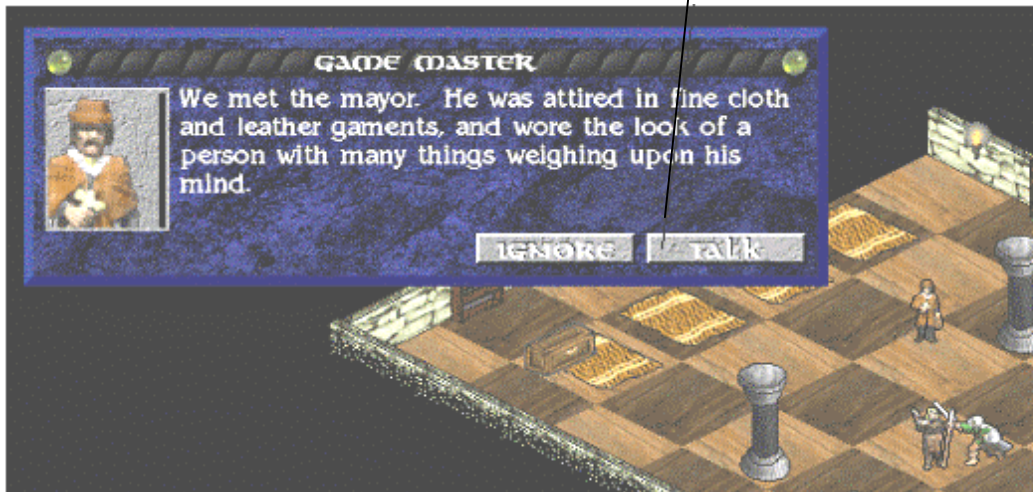
Bashing should be done by the strongest character. It will eventually succeed, but bashing may damage goods within the chest. Durable items and money (such as gold coins) will not suffer, but other items may get damaged. Fragile items such as potions in glass containers have a higher chance of getting damaged.



Picking is always better than bashing, but you need the Pick Locks skill to have a chance.

Talking with Folks

The encounter reaction screen may give one or more of these choices: Ignore, Talk, Search, Fight, and Flee.



Most conversations are done in "click a topic" style. Certain responses can cause dramatic action, while others may simply gather information.

You may occasionally be prompted to answer a riddle in a pop up text input screen.

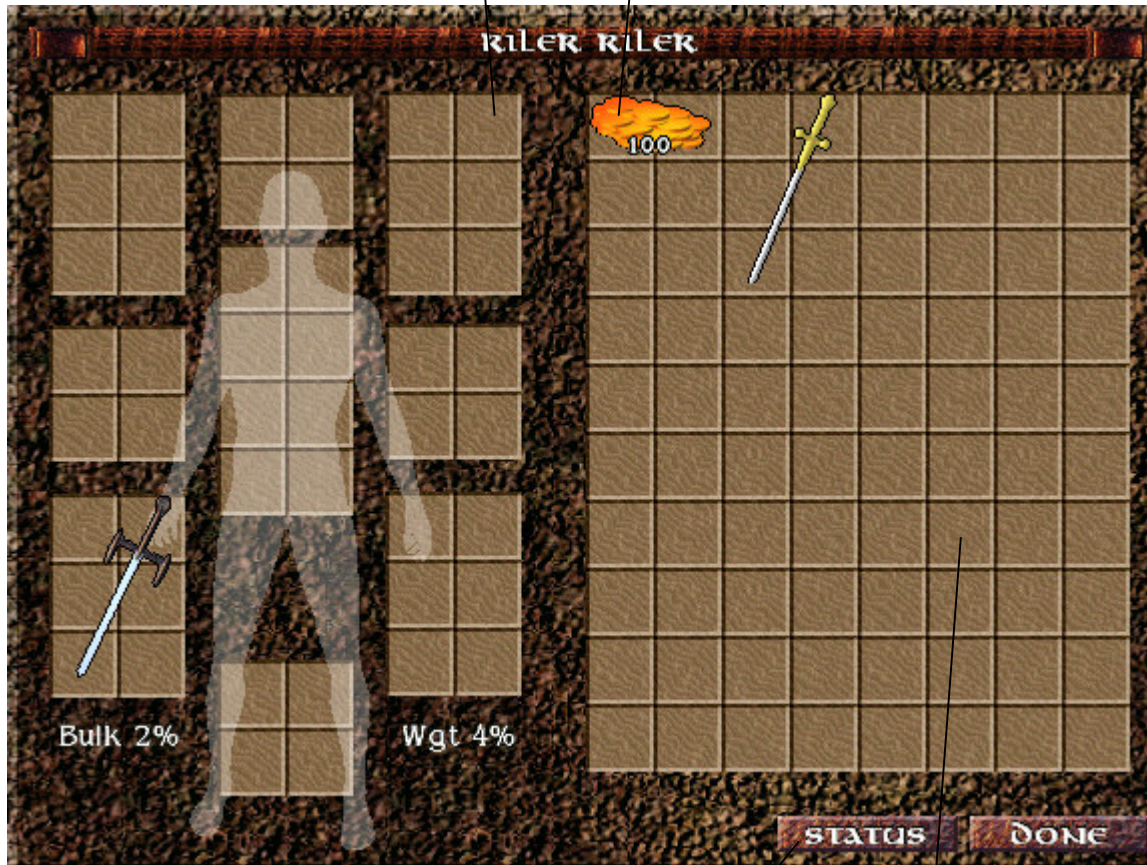
If there were other people to talk to besides Bertram, you would click their portrait here.

Inventory

Equip The Equip button shows the inventory for the active character.

Equipped items are being worn or wielded in hand. When an item is dragged to the character shadow image, it is automatically put in the correct body location.

Carried items appear in this large grid. Right-click an item to examine or use it.



Click the Status button to switch to the character's status screen.

Some items, such as gold coins, automatically combine. Rarely, you may find special items that must be manually combined, such as potion ingredients. To combine them, drag either item on top of the other one.

Status The Status button shows detailed character info.



Using Skills

Click the Skills button to access a character's unique set of skills.

Skill name and level. Each spell appears here, too.

Skill description and cost for raising

skill	Level
Bardsong	Novice
Health	Novice
Pick Locks	Novice
Revelspell	Novice
Essence Touch	*
Phase Self	*

BARDSONG

Three times per day, Bardsong gives +5 AP to all other party members for 3 turns. Extra skill levels enable you to bardsong more often. Performing Bardsong requires you to carry a musical instrument (but you don't need to put it in hand).

Cost: 18

Right-Click Action?

Lower Raise Use Done

If you have gained enough skill points, you can raise your skills, making them more effective (some skills have a limit). The Lower button is there to correct mistakes, and can only be used immediately after you raise a skill.

Click the Use button to use a skill or spell once. Some skills are always active, and cannot be used.

Left-click here to assign usable skills or spells to the right mouse button. This is handy for a commonly used skill or spell.

Options

The options may be accessed and changed any time during play. Users with

Play Music Turn this on if you want to hear music play during the adventure & combat. The default mode is ON. Note: Assuming you installed RuneSword II with the defaults, you may put your own MIDI or MP3 format music in the following directories:

- ❖ Fighting Music **Program Files\Crosscut Games\RuneSword\Data\Music\Combat**
- ❖ Travelling Music **Program Files\Crosscut Games\RuneSword\Data\Music\Adventure**

Set Game Speed Click this option to cycle through the game speeds. Slow is 1, Fast is 20. The default speed is 12.

Show Dice Rolling Turn this option on if you want to see dice rolling during combat. Turning it off will speed up combat slightly. Either way you'll see the resulting amount of damage. The default is ON.

Play Sound Effects Turn this on to hear sound effects in combat & while adventuring. This does not affect the audible click when using the mouse. The default is ON.

Use Debug Mode Turn this on if you want to see the Trigger Statements right after they fire. This helps in debugging what is happening in your Tome. The default is OFF.

Overswing Penalty Turning this on means all combatants may may a weapon attack even with insufficient AP. The AP shortage penalizes the attack roll. The default is ON.

Play MouseClick Sound Turning this on means you'll hear a sound when you click the mouse buttons. Default is ON.

Auto End Turn Turning this on means a character's turn will end when there is no one left in weapon range. The default is OFF.

Fast Move Fast Move ON means the party on the map will move faster than normal. Default is OFF.

Interface Settings Change the look of RuneSword's borders and menus.

Dice Set Pick a good looking dice set. Or a lucky one....

Fighting

On the combat screen, each character gets his own turn, has his own graphic, and can move separately. Click a unoccupied space to move, just as you do on the map screen. Movement and most other combat choices cost Action Points (AP).

Chrastos has 14 AP.

If this Harosaur was within range of a weapon attack from Chrastos, the harosaur would be highlighted. In this case, Chrastos is not wielding a weapon at all.

Chrastos has Revelspell skill, so he should pay attention to the Rune supply.

Chrastos already has Fireball assigned to the right mouse button, so he could right-click to cast that spell on the harosaur.

The Harosaur's Defense is 11, so that's what Chrastos would have to roll on a d20 weapon attack to hit.

If Chrastos wants to use another skill or spell, left-click the Skill button and pick one.

Standard Combat Action buttons.

As you target an enemy, 0-4 shield icons appear that are silver, or gold. The number of shields is a rough approximation of how well the target is covered by armor or tough hide. Two shields, for example, would be about 50% coverage. The wood, silver, and gold approximate how protective the armor or hide is: wood 25%, silver 50%, and gold 75%.

This harosaur's tough hide covers his entire body with 50% damage reduction.

Weapon Attacks and Other Choices

See Appendix B for the weapons list.

Click an unoccupied place on the combat screen and the character will move there (or as close as his movement range allows).

When you hover the pointer above a foe and the pointer changes to a sword, that means the foe is in range for an attack with your weapon – or your character can move to attack range. So if you click such a foe, your character will move if necessary and then attack.

Dice are usually rolled for damage. A typical long sword does 3d6 (3 six-sided dice) in damage. An arrow does 3d4 damage. For every 3 strength points above 10, the attacker does 1 additional point of damage. Of course, some special weapons also do additional damage.

Action Points and Overswing All weapons have an Action Point (AP) cost. Typically, a long sword costs 10 AP, a dagger costs 6, a two-handed sword 11, a bow 8, etc. The lower the cost, the more attacks you can get in per turn. Note that even if you lack the AP for an attack, you can still attempt one "overswing" attack before the end of your turn. You simply suffer a to hit penalty that's appropriate for the amount of AP you lack. See the section on Action Points for more info.

And In the Other Hand? Normally, you can only wield one weapon at a time. Most characters will want to hold a shield in the other hand. Using a two-handed weapon like a bow, however, prevents you from having a shield equipped. *Note:* Characters with the Ambidex skill are an exception; they can wield two weapons at once.

Not Attacking

Defend To end a character's combat turn before his Action Points are used up, click Defend. This converts leftover AP into a Defense bonus.

Wait Choosing Wait enables you to take delay the rest of the character's turn to the end of the combat round.

Flee Sometimes discretion is the better part of valor. Characters will automatically move along the best route to flee the combat to reach the screen edge (it may take more than one turn of fleeing). Sometimes, the flee attempt fails.

Action Points

Action Points (AP) govern how much a character can do on his turn in combat. AP has no effect outside of combat. AP is the character's experience level plus the average of Strength, Agility, and Intelligence. When AP is used up, the character's turn ends. AP is reset to the maximum at the beginning of a turn.

Action Point Cost (Combat Only)

- Combat movement costs 1-6 AP per space, depending on Agility.
- Attacking with a weapon costs the weapon's AP amount. For instance, a typical long sword costs 10 AP to attack, while a dart only costs 5 AP.

➤ By default, the overswing option is on. Weapon attacks occur with insufficient AP, but there is a penalty.

- Using certain skills costs AP. Check the skill description.
- Casting a spell costs 10 AP.
- Using an item, like a key, a wand, a lamp, or whatever, costs at least 1 AP.
- If you're desperate for protection, you may decide to wear body armor even if you lack the Battle Armor skill. If you do so, you pay an huge AP penalty at the start of your turn.

A typical 1st level character has around 14 AP each combat turn. Most long swords use 10 AP per attack. A dagger uses only 6 AP, while a two-handed battle axe uses 13 AP.

Moving costs 4 AP per space for creatures with normal Agility, but a few races have high enough Agility to reduce the movement cost to 3 AP per space. The full listing, if you're curious:

Agility	Movement Cost Per Space
4 or less	6 Action Points
5-9	5 Action Points
10-14	4 Action Points
15-19	3 Action Points
20-24	2 Action Points
25+	1 Action Point



Goblins were that rare combination of many action points, but also ineffectual, idiotic, or even suicidal actions.

Defense and Armor

See Appendix A for the armor list.

The target's defense number -- the minimum attackers need to strike rolling a 20-sided die -- is 13 for normal size creatures. Large creatures are easier to hit; smaller ones are harder. The evasion skill and some spells can increase the defense number. Armor generally detracts from defense (see below).

Armor

Wearing armor in RuneSword makes you easier to hit, but it reduces the damage that you receive. At maximum, you can wear one body armor type, shield, and helm simultaneously. This will cover most of the possible hit locations; the only exposed area is your neck. If a character receives loses more than 25% of his health max in one blow to the head, he may suffer additional adverse effects (stunned, minuses to hit or damage, etc.). Helms decrease the chances of this happening.

Wearing armor is better than not wearing armor, especially for characters that tend to fight with short-range weapons. However, body armor should not be worn unless the character has the Battle Armor skill (this negates the Action Point penalty). Martial Artists cannot wear any type of armor. Spellcasters cannot wear shields or helms.

Other Protection

Some creatures, like dragons, may have an especially tough hide that works just like armor, except that they don't actually wear armor items. Some creatures may be completely covered by this natural armor, or maybe they have vulnerable spots like the head or a soft underbelly.

Also, creatures may have protection or vulnerability to certain types of attacks, like fire, cold, mind, evil, etc. They may suffer additional damage, or less damage, from these special attack types. Some magical items may offer these kinds of protection.

Spellcasting

See Appendix C for the spell list.

To cast a spell, click the **Skills** button, select the spell, and click the **Use** button. A commonly used spell can be assigned to the right mouse button for quick casting.

There are six spell skills in RuneSword, each based on a vice. To be a spellcaster, you choose one or more of the following skills: Lustspell, Lunaspell, Revelspell, Pridespell, Wrathspell, and Greedspell. Some may count Dreamspell as a 7th skill, but this is not available to player characters. If you gain a level in a spellcasting skill, you gain an additional spell. Also, many spells are more powerful when the skill is a higher level.

Runes Spells are cast by ordering runes, which are drawn from nearby magical energy. This energy varies in areas. The energy can be drained, but recharges at varying rates. In combat, spells with 1-2 runes fire immediately, 3-4 rune spells take 1 turn, and 5-6 rune spells take two turns.

Fizzling Spells that are beneficial, or don't target anyone, have a chance of failure based on the Intelligence of the spellcaster. The base chance is 50% failure, but each point of Intelligence above 10 reduces the failure chance by 5%.

Resistance Harmful spells may be resisted by a sentient target. The target's vice score in the spell's school determines his chance of resisting the spell. Spell resistance is 5% with vice score of 20, with a +5% for each point less. Suppose you cast a fireblast spell from the Wrath school against a soldier with 18 Wrath (an average amount). The soldier would have a 15% chance of resisting the spell. Because non-sentient creatures lack vice scores, targeted spells automatically succeed against them.

Theory Some RuneWorld academics have speculated that this resistance proves that all magic is illusory, because a real wall of flame, or an animated axe, could never be "resisted." Other say that magical effects are real, and spell resistance is simply an inner magic that all higher beings possess. Still others claim that only Dreamer magic is of the essence, and Vice magic is merely a lesser shadow.

Creator Guide

The Creator Guide describes the Creator's interface and some basics for tome construction. Much of this guide shows the "property sheets" of tome components.

Some people in the RuneSword II community specialize in certain areas, like drawing new graphics, composing new sound effects or music, or scripting complex situations for the non-programmers. Few people are "do-it-all" types, so rely on the community for support. In fact, the underlying idea of the game is collaboration.

Tome Creators are storytellers.

If you are a newcomer to RuneSword II and interested in creating tomes, this would be ideal:

The Path of Creator Enlightenment

- 1. Play the included adventures extensively.**
- 2. Watch the Creator videos (they're in AVI format on the CD). For the learning style of most people, the videos are incredibly helpful.**
- 3. Fire up Creator and load the tomes you've played. Look at how they're put together. Make some changes to them and see how they affect play.**
- 4. Skim this book, the Creator Guide. Get comfortable with the interface and follow the step-by-step instructions on making a tome from scratch.**
- 5. Most of the Encyclopedia Arcana is reference, but read the intro sections on Events and Triggers.**
- 6. Make a new item, or creature, or a map. Or even make a tome. Don't go for an epic right away – try for something small and cool. And don't forget to post it on the web site, because we want to play!**

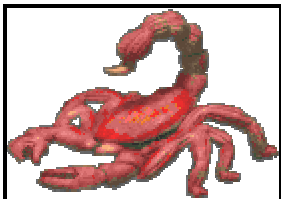
Come to www.runesword.com for answers to your questions, find out what other members of the RS community are doing, see the developer's plans, etc.

Step By Step: Creating a Tome

Let's do a step-by-step small tome to highlight some basics. Most stuff is loaded from a subdirectory within the directory path PROGRAM FILES\CROSSCUT GAMES\RUNESWORD\LIBRARY.

If you have any questions about the interface, watch the videos and skim the rest of the Creator Guide, especially these sections: The Basics, Paint Graphics Mode and the Paint Encounter Mode screens.

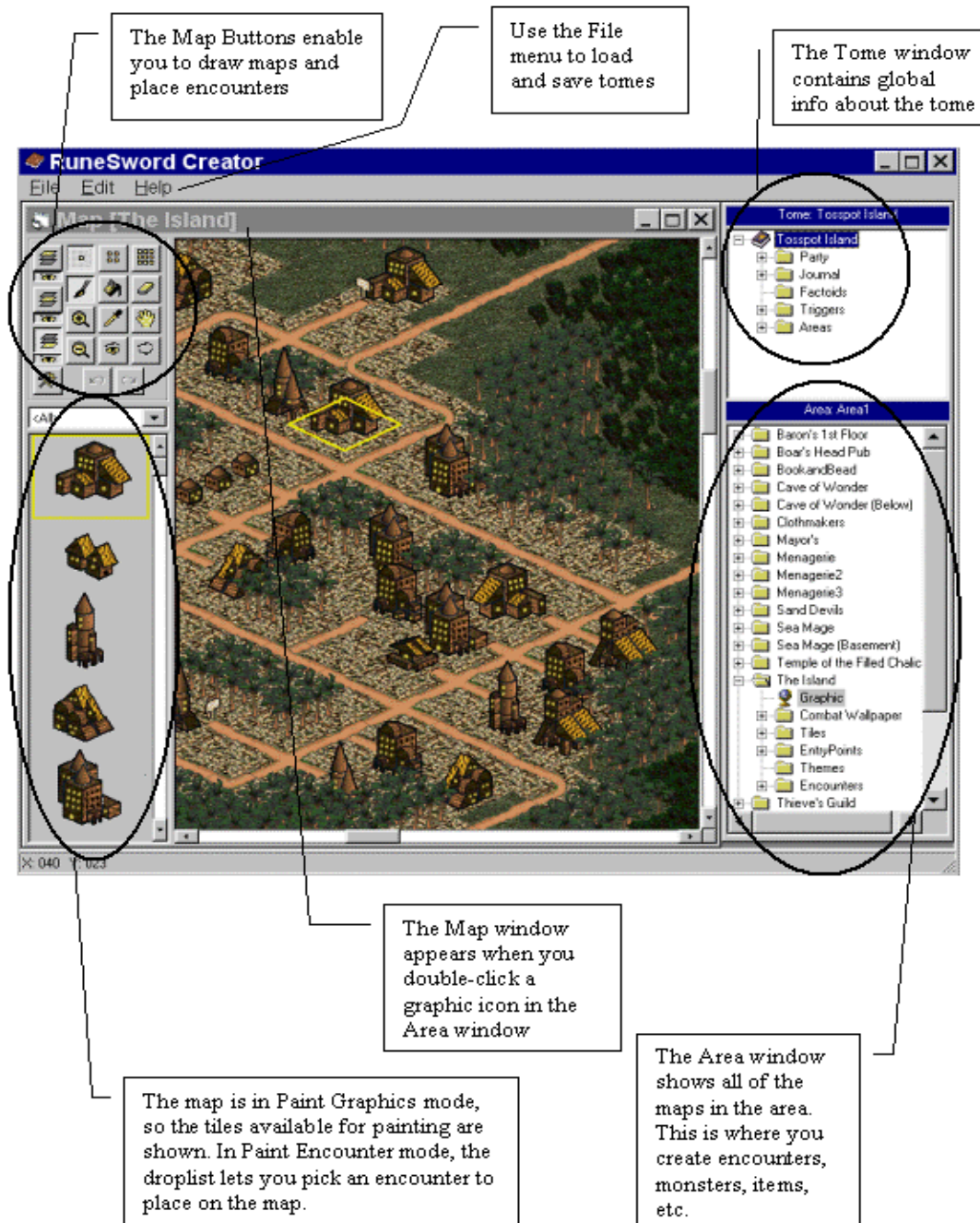
1. Fire up the Creator program. Hit Cancel when the Tome Wizard fires up. By default, it's ready for a new tome. We'll make a tiny map, put an encounter on it, create a test character, and check it out.
2. The map is first. We will load a sample map so that the tile properties are set, but we will erase the demo map and create our own. In the Area Edit box on the lower right, right click the Map1 (the default map) and select **Insert Map from Library**. Within the LIBRARY\MAPS directory, select *Underground*. After it loads, you'll have 2 maps, Map1 and Underground. Go ahead and delete MAP1 by right-clicking it again and selecting **CUT**.
3. Double-click the graphic icon below the map name. The diamond-shaped map is grey at the borders, and the map itself shows a tiny cavern. If you want to practice map painting, click on the **Eraser** icon and hold the left button down, moving around the map to erase stuff. Change back to paint mode and paint a little map. Make sure to put the floors in on the bottom layer and the walls on the middle layer.
4. Time to make an encounter. Further down in the Edit Area box, right click on the Encounters folder and choose **Insert Encounter from Library**. In the LIBRARY\ENCOUNTERS folder, select the *Oh No A Scorpion* encounter. For extra credit, right click the scorpion monster and alter its stats.



Double my strength,agility, and health!

5. Our scorpion's ready to rock, but we haven't put him on the map. Click on the **set encounter** button in the upper left of the map window. Our encounter is selected because it's the only one on this map. Click four or five tiles in the middle of the map to set the encounter area. To put the scorpion down, click the check box and then click somewhere in your highlighted area.
6. We need a starting point (a place where the intrepid adventurers will appear). Right-click the Entrypoints folder and choose **New EntryPoint**. For now, don't worry about anything but the X Y coordinates for where the party should appear (we don't need to make this entrypoint link to another map or anything. The coordinates 0,0 are the left diamond point. The coordinates appear in the lower left status bar as you run the diamond cursor over the map. It doesn't really matter which coordinates you choose, as long as they're outside our scorpion encounter. Click **OK**.
7. Now you need to tell RuneSword to use that particular EntryPoint as the Starting point. Right-click on the Tome Name ("Untitled") and select **Tome Properties**. Select *Underground* as the map in area for the starting point. As soon as you do so, our newly minted EntryPoint will appear as the starting point, because it's the only entrypoint on the map! (With more than one entrypoint, you'd have to choose the correct one from a droplist). Don't forget to name your tome now -- this is the name that appears on the main menu when you choose a tome to play.
8. Use Windows Explorer to make a subdirectory named **TEST TOME 1** within the RUNESWORD\TOMES directory. Back in the Creator, choose File and Save. and give your tome it's filename (most Creators use the Tome's name as the filename). The extension .TOM is automatically appended.

The Basics



The Tome and Area Windows

The Tome window is at the top, the Area window at the bottom. Right-clicking on objects in these windows shows the pop up menu for cut, copy, paste, properties, etc.

Right click here to access the tome's property sheet. There, you can set the starting entry point, how many character (and how powerful) are allowed, and other critical info.

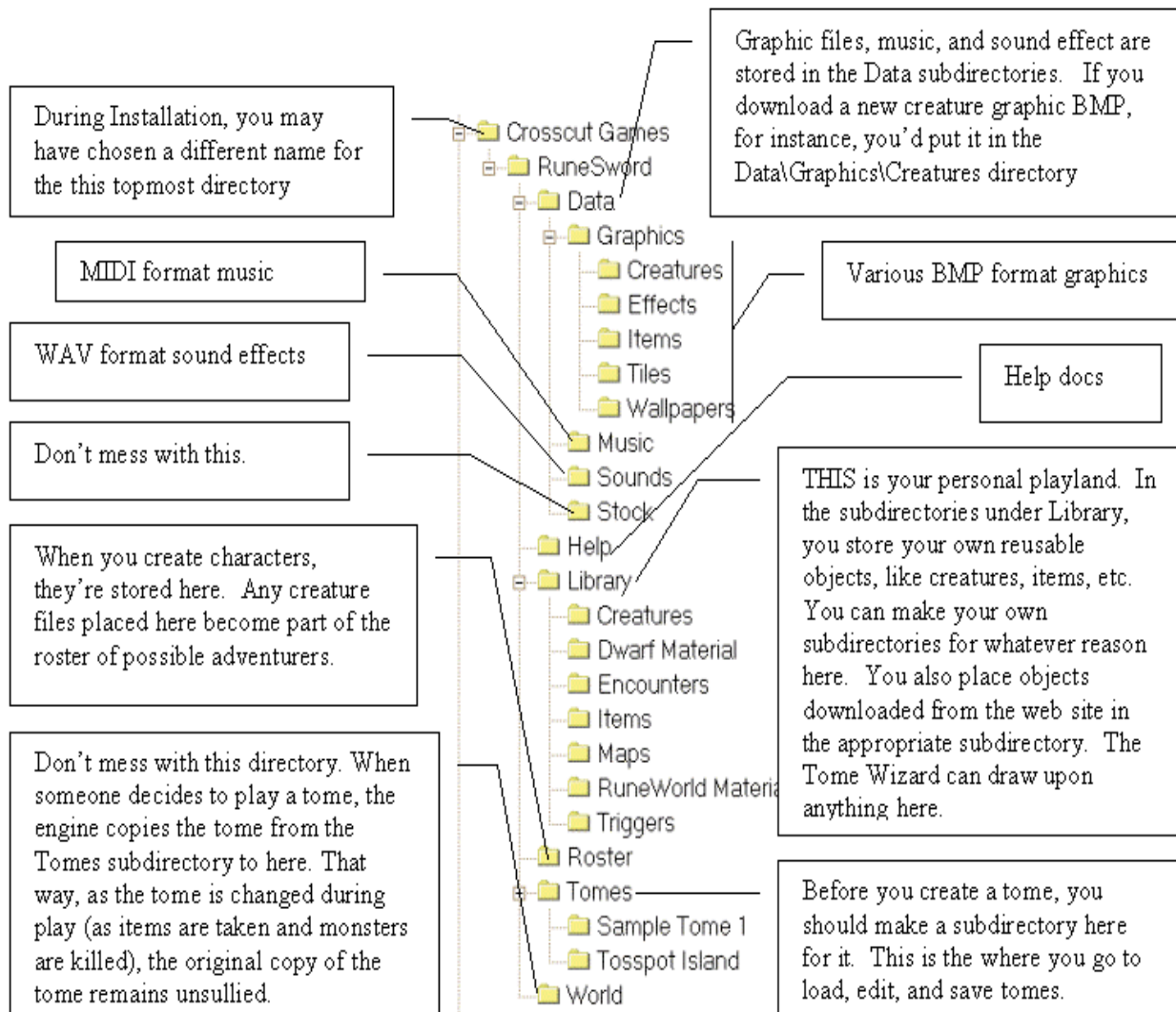
Tome Window
You can create a tome without messing with any of these folders. The only one that is commonly used in the party folder, and even that is unnecessary if you want your players to use only their characters

Party pasting characters here makes them automatically part of the adventuring group
Journal if you want a entry to appear in the journal at the start of the adventure, put it here
Factoids if you want these "global variable" to have pre-set values, you could add them here
Triggers if you want events to occur upon starting or exiting the tome, this is the place
Areas an area may hold many maps, but if your tome has more the 25 or so, you may add additional areas.

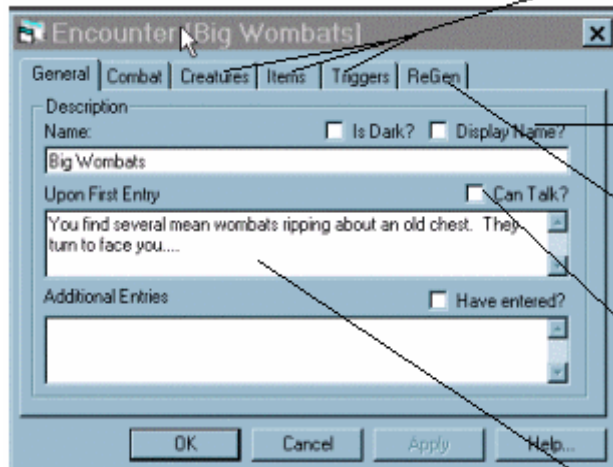
The Area Window
Each main folder represents a map's contents, and each map has it's graphic and five subfolders. This is the place where you create stuff.
Graphic double click this to bring up the Map window, which enables you to paint maps and place encounters
Combat Wallpaper the default background for the combat screen. Individual encounters may override this.
Tiles the tiles available for the map
Entrypoints these are where you enter and exit from one map to another
Themes generate extra maps and randomly fill them
Encounters the most commonly used folder, this is where you create encounters of monsters and items for characters to find

Right-click here to access the map's property sheet. You can set the map size, whether it's randomly generated, the rune supply, and so on.

Directories for Storing Stuff



Encounters



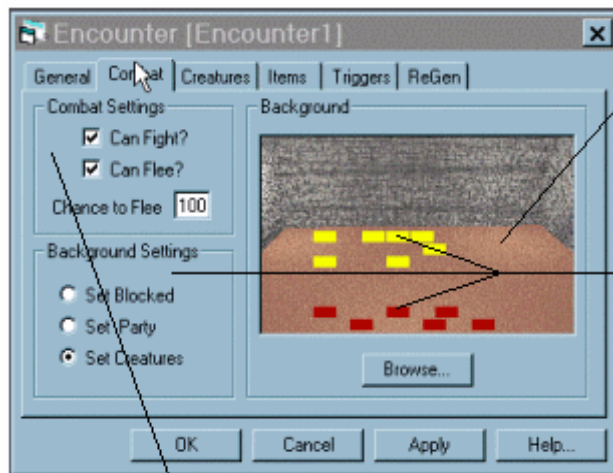
The Creatures, Items, and Triggers tabs allow you to access such stuff when they're attached to the encounter. You can also access them in the "tree view" of the area window.

The Regen tab enables you to randomize the encounter a bit. The "Automatic" video shows how to generate encounters, among other things.

Determines whether the Talk option appears on the encounter reaction screen (you still need to put a convo on a creature)

If checked, hovering the pointer over the encounter displays the encounter name

This is the text on the encounter reaction screen. If you have text in Addition Entries, that shows if the encounter occurs again.



Use the Browse button to set up a combat wallpaper for an encounter (if not, the default combat wallpaper for the map is used)

Click a background setting object, then left click the picture to place it. If you don't, the engine will do it for you randomly. Party starting locations are yellow, creatures red, and blocked spaces blue. Blocked spaces aren't normally used (see the section on special combats at the bottom of this page).

If checked, Fight and/or Flee appear on the encounter reaction screen. You can also set the success chance for fleeing.

Notes on Combat Encounters

Customizing combat encounters can be fun. The only option that is a “must do” is the first, combat scenery. The rest are simply some ways you can spice up combat terrain.

Combat Scenery Normally, you need not mess around with the Background Settings for combat screens. Just pick a default combat wallpaper for the entire map, and override it if necessary for certain encounters with the above screen.

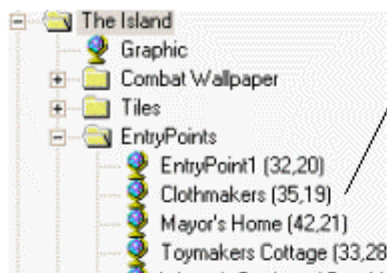
Creature Placement The RuneSword II engine will randomly place your party members and the enemies. If you want to specify where the combatants appear, left-click the combat background graphic with *Set Party/Set Creatures* selected (right-click to remove them).

Large Obstructions But what if you have a special combat wallpaper that has a river of lava flowing down the middle of the screen? Or a stone idol in the lower right? It would sorta destroy the illusion if characters just walked all over those things like any other piece of turf. That's where the Set Blocked spaces comes in. Just put them in the right places, and characters won't be able to walk there.

Obstacles One other late addition to the construction set is the "inanimate" setting for creatures -- it's on the Creature property sheet, but these docs haven't been updated with it yet. Inanimate creatures are placed randomly on the combat screen, and they don't get turns or anything (events don't even fire for them). "Creatures" is a misnomer; it's really a way of getting barrels, or weapons, or other item graphics placed as combat scenery.

EntryPoints

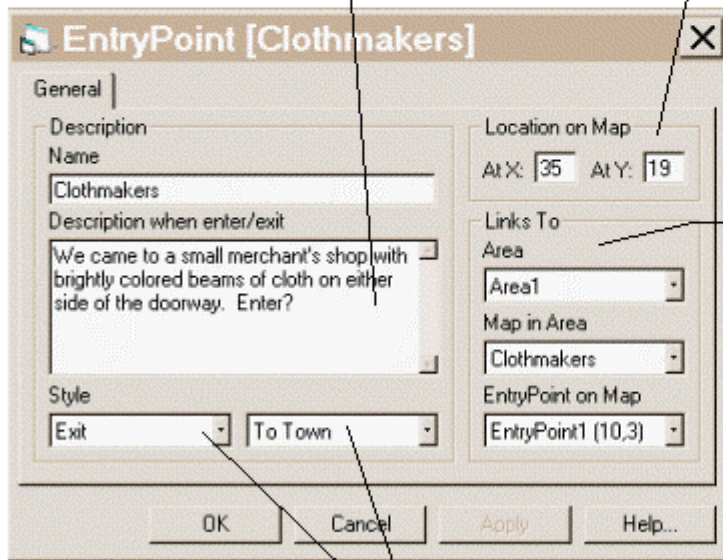
EntryPoints are the means of starting the adventure, connecting maps, and leaving the adventure.



Right-Click this!

Whenever a player steps onto an entrypoint, he is asked a yes/no question about using it.

Entrypoints are not painted on the map like encounters. Their position is defined here



All entrypoints link to another entrypoint or the main menu (to exit from the tome).

For the first entrypoint you make, you must wait to establish the link until a 2nd entrypoint is created.

Unimplemented

Creatures

Applies to player characters only. The engine checkmarks this when a character is on an adventure, preventing him from appearing on the roster

The Item and Trigger tabs are simply shortcuts to access those objects.

The yellow box determines which part of the graphic is being used for the portrait

The ExpPts and SkillPts is how much is gained by the party for killing this creature (SkillPts are seldom awarded for any but the toughest creatures).

Prevent Search means that you can't take anything until this creature is gone or dead

Friendly means that the creature won't automatically attack

If even one creature in an encounter has Can't Ignore checked, players cannot ignore the encounter.

If an NPC joins your party, this determines whether he/she can exit the tome and become a real travelling character... like Pinocchio!

Health now and health max. For 1st level creatures, this is usually 10-12. For beginning player characters, this is 40-50.

Current amount of Eternium, a substance needed for spellcasting. Eternium does not

You cannot edit movement and action pts directly (they're derived from SAW). Both apply to combat only.

Vices only apply to sentient creatures. For player characters, they average around 18 on a 1-20

A human would be medium

Check all that apply. Makes spells and traps that target certain family type work

Strength, Agility, and Will (SAW) are the 3 ability scores. For beginning characters, they average around 12-14.

General | Stats | Resistance | Sounds | Items | Talk | Triggers

Description

Name: Shaman Kivex Not in Roster? ☐ Race: Kivex

ExpLvl: 1 ExpPts: 18 SkillPts: 0

Comments: ☒ Computer Controlled?

☒ Can't Ignore? ☒ Prevent Search? ☐ Friendly? ☐ Locked in Tome?

Picture: ☐ Male ☐ Female

Browse...

OK Cancel Apply Help...

General | Stats | Resistance | Sounds | Items | Talk | Triggers

General

Health: 15 / 15 Eternium: 9 Strength: 10 Agility: 10 Will: 10 Movement: 3 Action Pts: 11

Family: ☐ Sentient ☐ Plant ☐ Reptile ☐ Bird ☐ Insect ☒ Undead ☐ Blob ☐ Aquatic ☐ Animal ☐ Magical

Size: ☐ Tiny ☐ Small ☐ Medium ☐ Large ☐ Huge

Vices: Luncy: 10 Revly: 10 Wrath: 10 Pride: 10 Greed: 10 Lust: 10

OK Cancel Apply Help...

This is not worn armor, like chain mail (that's done with an armor-type item). This is for tough hides, like a dragon. It's also a good place to specify unusual body part types, like tentacles, because the engine will show that as the hit location.

For humanoid in RuneWorld, we don't get into specifics for the body type name much. Only 1 location (Head) is special – with enough damage, the creature must roll on a head damage “chart.” The others are just descriptive unless you specify a resistance %.

	Body Type	Resistance	Covered By
1	Body	0 %	Body Armor
2	Body	0 %	Body Armor
3	Body	0 %	Body Armor
4	Body	0 %	Body Armor
5	Body	0 %	Shield
6	Body	0 %	Shield
7	Head	0 %	Helm
8	Body	0 %	<None>

	Resistance Bonus
Sharp	<None>
Blunt	<None>
Cold	50%
Fire	Double Damage
Evil	<None>
Holy	<None>
Magic	<None>
Mind	<None>

Buttons: OK, Cancel, Apply, Help...

This is the % damage is reduced if the creature is hit in that location. Humanoids generally lack tough hides (maybe for lizard people, though...), so they have 0 in all 8 locations.

The “Covered By” column is just for your information. Humanoids can wear armor (3 types – body, shield, & helm) and this shows which location are covered by the armor types. Note that location 8 is always a clean shot.

On a successful hit, the engine randomly selects 1 of 8 possible locations

Any special resistance bonus selected here applies no matter where the creature is hit, as long as the damage is of the specified type. You may select a % of damage reduction, or you can even choose to make the creature suffer double or triple damage from certain attack types.

Events:

- When Move in Combat
- When Attack in Combat
- When Hit by an Attack and Damaged
- When Die

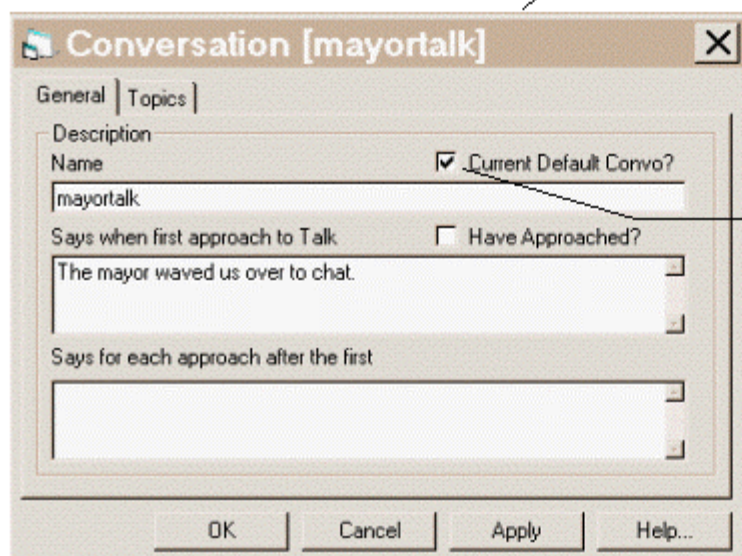
Sounds:

[Default]

Buttons: Browse..., Play, One Time Only?

Buttons: OK, Cancel, Apply, Help...

On the Sounds tab, you can customize which WAV file plays (or make your own) when certain things happen. Personally, I favor audio clips from Blazing Saddles and Dr. Strangelove. Burn likes Pinky and the Brain....



Conversation [mayortalk]

General | Topics

Description

Name ☒ Current Default Convo?

mayortalk

Says when first approach to Talk ☐ Have Approached?

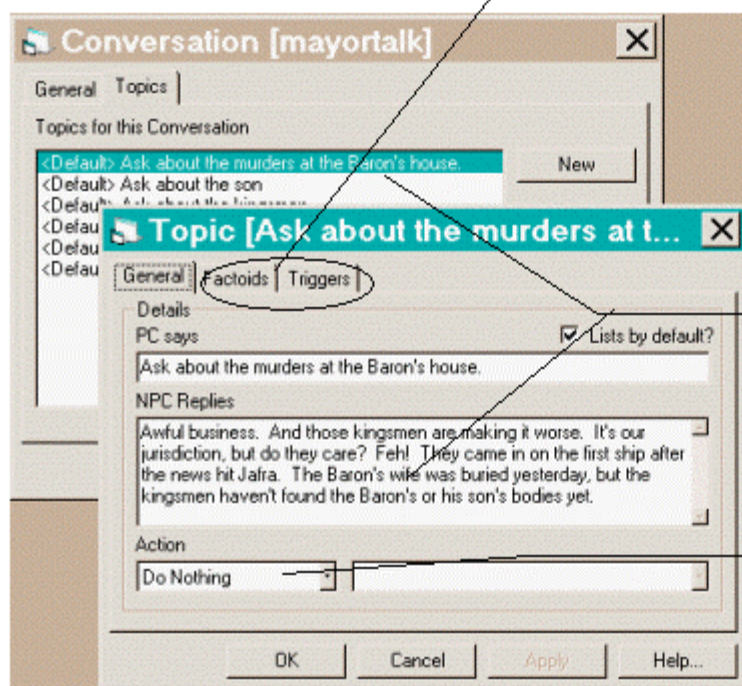
The mayor waved us over to chat.

Says for each approach after the first

OK Cancel Apply Help...

On the Talk tab for creatures, you can add conversations, like this one with the Mayor of Tossport

If the creature has multiple conversations, the default one is the first convo to fire up



Conversation [mayortalk]

General | Topics

Topics for this Conversation

<Default> Ask about the murders at the Baron's house. New

<Default> Ask about the son

<Default> Ask about the kingsmen

<Default> Ask about the murders at the Baron's house.

<Default> Ask about the son

<Default> Ask about the kingsmen

Topic [Ask about the murders at t...]

General | Factoids | Triggers

Details

PC says ☒ Lists by default?

Ask about the murders at the Baron's house.

NPC Replies

Awful business. And those kingsmen are making it worse. It's our jurisdiction, but do they care? Feh! They came in on the first ship after the news hit Jafra. The Baron's wife was buried yesterday, but the kingsmen haven't found the Baron's or his son's bodies yet.

Action

Do Nothing

OK Cancel Apply Help...

Factoids are unimplemented here.

Triggers (pre or post topic) are occasionally used to do something special, like give the character a key for choosing a certain topic. Or even preventing a topic from showing unless some condition is met.

On the topics tab, you enter the topics that the PC can ask this creature about. Double-click the topic to open up the topic window, where you can enter the creature's response.

You can specify a result if the player selects this topic, like ending the convo, starting a new convo, removing the topic, bringing up a new topic, etc.

Items

If checked, this item will autocombine when picked up with other same name items. The name of the resulting item is "pluralized."

Value is expressed in gold pieces or whatever the standard monetary unit is. Note that if Count > 1, Value applies to the total number of the item, not per item.

How many. The name will be "pluralized" by the engine automatically in play.

The screenshot shows the 'Item [Sword]' dialog box with the following fields and annotations:

- General Tab:**
 - Name:** 'Sword' (An annotation points to the 'Can combine' checkbox, stating: 'If checked, this item will autocombine when picked up with other same name items. The name of the resulting item is "pluralized."')
 - Comments:** 'This elegant blade has a fleur de lis engraved in its hilt.' (An annotation points to the 'Use as Description' checkbox, stating: 'Use as Description')
- Attributes:**
 - Value:** '15' (An annotation points to this field, stating: 'Value is expressed in gold pieces or whatever the standard monetary unit is. Note that if Count > 1, Value applies to the total number of the item, not per item.'
 - Count:** '1' (An annotation points to this field, stating: 'How many. The name will be "pluralized" by the engine automatically in play.'
 - Weight:** '15' (An annotation points to this field, stating: 'In RuneWorld, coins, flasks, gems, etc. all have 0w/0b. A sword or club is 15w/5b. A small idol or doo-dad might be 3w/3b. Body armor is 20-/150w and 25b. Small shields are 12w/3b. Large chests are 30w, 101b. Weight limit is 15xStrength. Bulk limit is 100 (items have 0 bulk while equipped).')
 - Bulk:** '5' (An annotation points to this field, stating: 'Capacity is how much bulk the item can hold (if the item isn't a container, leave this 0). If you put any items inside (on the Contents tab), this must be > 0.')
 - Capacity:** '0' (An annotation points to this field, stating: 'Check Soft Capacity if the item is a bag, sack, or similar "soft" container.'

Bonus on the to hit roll (the 20-sider)

How many action points it costs to use this item

The lower two-thirds of this screen will vary according to what Family is selected.

If the item is attached to a creature, checking this means the item will be equipped.

Positive number here helps a creature's defense number (number others need to hit him). Negative number will hurt Defense. This usually applies to some armor and magic items, but you are allowed to enter a value on any item.

Normal body armor hurts the defense (-1 leather, -2 chain, -3 plate). Armor makes up for this by reducing damage with resistance %. Select the Armor family to see this. The default defense number for 1" level creatures is 13.

2 handed items do not allow the you to equip a shield

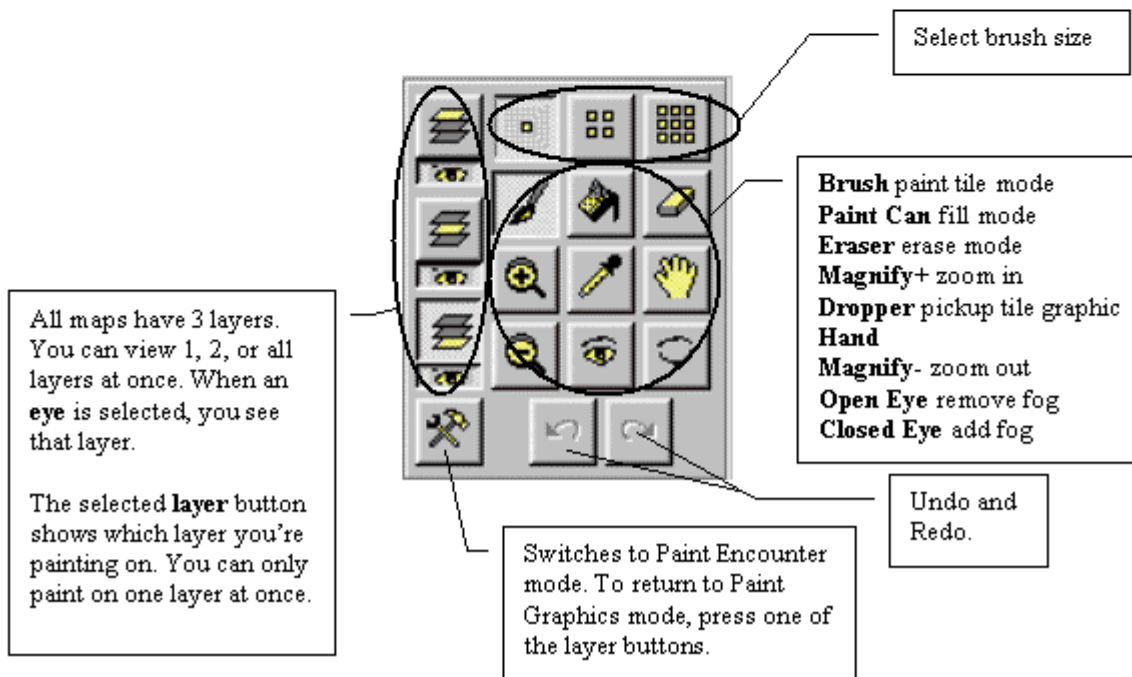
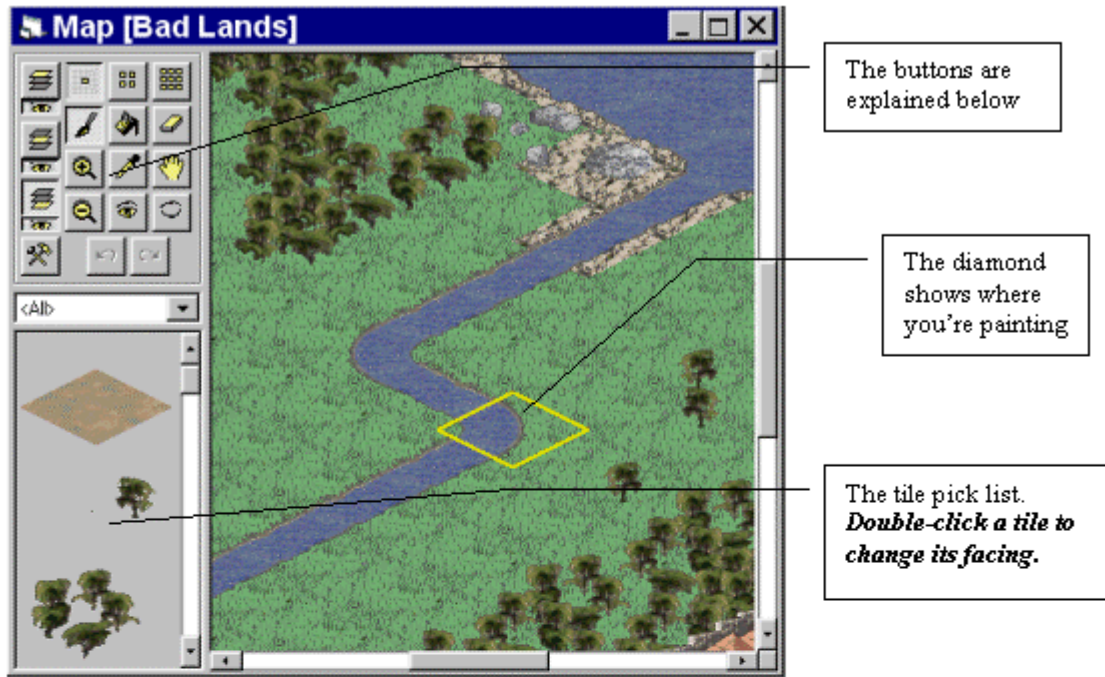
Short is 1 space, Medium is 1-2 spaces, Long is 3+ spaces away on combat screen.

This means three six-sided dice

The screenshot shows a dialog box titled 'Item [Sword]' with tabs for General, Type, Contents, and Triggers. The General tab is active. It features a 'Family' dropdown set to 'Weapon (Melee)' and a checked 'Is Equipped?' checkbox. The 'Attack' section includes 'Action Points' (10), 'Attack Bonus' (2), 'Defense Bonus' (0), and 'Range' (Medium). The 'Damage' section includes 'Damage Type' (Sharp), 'Damage' (3d6), and 'Damage Bonus' (0). A 'Requires two hands?' checkbox is at the bottom left. Annotations with leader lines point to various fields: 'Bonus on the to hit roll (the 20-sider)' points to 'Attack Bonus'; 'How many action points it costs to use this item' points to 'Action Points'; 'The lower two-thirds of this screen will vary according to what Family is selected.' points to the 'Family' dropdown; 'If the item is attached to a creature, checking this means the item will be equipped.' points to 'Is Equipped?'; 'Positive number here helps a creature's defense number (number others need to hit him). Negative number will hurt Defense. This usually applies to some armor and magic items, but you are allowed to enter a value on any item.' points to 'Defense Bonus'; 'Normal body armor hurts the defense (-1 leather, -2 chain, -3 plate). Armor makes up for this by reducing damage with resistance %. Select the Armor family to see this. The default defense number for 1" level creatures is 13.' points to the 'Damage Bonus' field; '2 handed items do not allow the you to equip a shield' points to 'Requires two hands?'; 'Short is 1 space, Medium is 1-2 spaces, Long is 3+ spaces away on combat screen.' points to 'Range'; and 'This means three six-sided dice' points to 'Damage'.

The Map Window

“Paint Graphics” Mode



“Paint Encounter” Mode

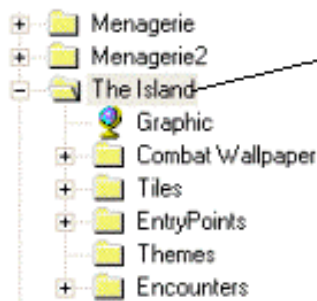
This button selects Paint Encounter mode. To return to Paint Graphics mode, click one of the level buttons above it.



Select the encounter to place from the droplist – note the encounters are actually created in the Area window. The creatures and item graphics in the encounter (if any) are placed by clicking the check box and then clicking on the map.

Clicking on the map places the encounter by “yellowing” the tiles

Maps



The Encounters, TileSets, Tiles, and Themes tabs just provide another way of accessing those folders

The ReGen property page is detailed on Maps Part II. It only applies to randomly generated maps. To populate those maps with critters and items, you must also use a theme.

Map [The Island]

General | Encounters | TileSets | Tiles | ReGen | Themes

Description

Name: The Island Width: 64 Height: 128

Comments: The main outside map of this tome, this shows the village of Tossport and its environs

Map Style: Town

Tiles

Picture File: island.village.bmp Auto-Generate? ☐ Browse...

Available Runes

Blood	1	Suns	2
Bile	2	Space	3
Oil	2	Insect	1
Nectar	3	Man	2
Fire	1	Fish	2
Earth	2	Animal	3
Water	2	Twilight	1
Air	3	Abyss	2
Time	1	Dreams	2
Moon	2	Eternium	3

OK Cancel Apply Help...

Max is 256 x 256 (dat's BIG). If you need more, you can connect maps in a way that makes them seem bigger.

Unimplemented

The Picture file bmp determines which tile graphics are available. Selecting Auto-Generate will automatically create 1 tile for each graphic within the Tiles folder, but it will not set the properties for the tiles (see screen 8).

This shows the rune supply for spellcasting for this map. You may reduce or raise these values for low and high magic maps

Unimplemented

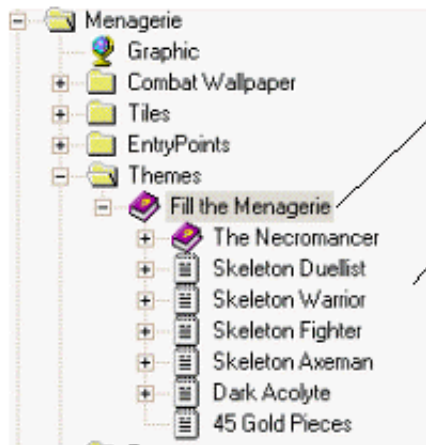
This style changes how the maps are made.

The screenshot shows a dialog box titled "Map [The Island]" with a close button (X) in the top right corner. The dialog has several tabs: "General", "Encounters", "TileSets", "Tiles", "ReGen", and "Themes". The "ReGen" tab is currently selected. Inside the "ReGen" tab, there are two main sections. The left section is titled "ReGenerate Map" and contains a "Difficulty" dropdown menu set to "Average", a "Generate Style" dropdown menu set to "Tetris Spin", a checked checkbox for "Can be reused in Wizard?", an unchecked checkbox for "ReGenerate upon first entry?", and a "ReGen Now" button. The right section is titled "Default Size" and contains input fields for "Width" (32), "Height" (32), "Encounters" (30), and "ExperiencePoints" (0). At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help...". Annotations with lines pointing to specific elements are present: one points to the "Difficulty" dropdown with the text "Unimplemented"; another points to the "Generate Style" dropdown with the text "This style changes how the maps are made."; a third points to the "ReGen Now" button with a text box explaining its function; and a fourth points to the "ExperiencePoints" input field with a text box explaining its purpose.

Click this to test how your map might look (if you don't select ReGen upon 1st entry, this will be how your map looks). If you don't get a good map, make sure that the style is set correctly for enough tiles (see screen 8).

- This creates "empty" encounters, that are necessary if you have a theme for this map.

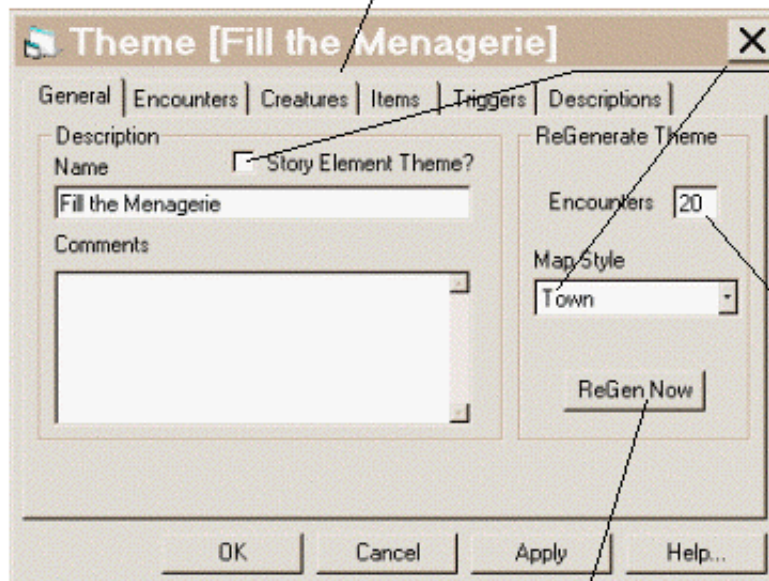
Themes



Right-click this!

Notice that you can attach encounters, creatures, and monsters to themes. Any pre-set attached encounters are used first, before the theme generates the random encounters by using the attached creatures and items.

In this example, I have 1 “showdown” encounter with an ultimate bad guy, the Necromancer (that encounter has a specific description, creatures, and items). The skeletons, acolytes, and goldpieces are used to randomly populate other encounters.



These other tabs are simply an alternate way of accessing these objects.

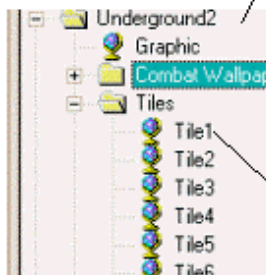
Unimplemented

This is how many encounters will be filled. The pre-set encounters (if any) attached to the theme are used first, then the attached creatures and items are used to fill in the rest.

You can click this to generate the theme in the creator, and the check to see how it works (look at the map’s encounters to see what it’s put in).

Themes can populate hand-drawn or randomly-generated maps, but in either case, the map must have enough empty encounters to accommodate the theme (empty means the encounter has a name, but nothing else). Randomly generated maps can also generate empty encounters.

Tiles



If you're a lazy slob (my people!), you may think to yourself, "All this tile stuff is sorta cool, but setting properties for the 60 or so tiles I'm using in my map is gonna take a half hour! I'll miss the Simpsons!"

Never fear. Instead of loading tile graphic in the map properties, generating the tiles, and setting the properties for each tile, there is a better way. Load a "Starter Map" instead from the MAPS directory. These are sample maps that shows off how tiles can be put together. *They already have the tiles loaded and their properties set.* Once you're done looking at the sample map, erase it and you're ready to go!

Right-click this!

This middle section is only used for door tiles. The % to open applies only to attempting to bash the door open. If you check a certain pattern of Key Mask boxes here, a key-type item with a matching Key Mask will open the door.

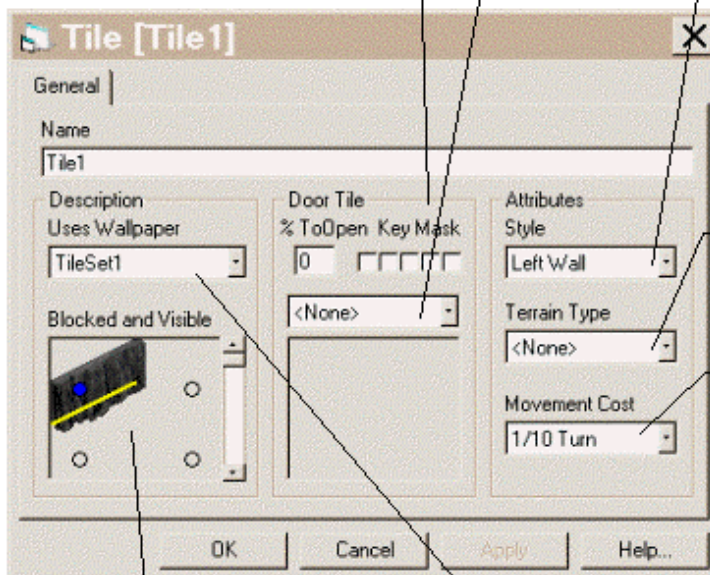
If you want the locked door to transform into an open archway when opened, select the archway tile here (a preview of the tile appears below it)

If you want the engine to generate a map, you MUST set up at least one tile for each of the 10 or so styles. This tells the engine whether the tile is a floor, wall, or whatever, so it knows how to place it.

Unimplemented

This is how much time it takes to move through the tile. You may want rough rocky areas to take longer to move through

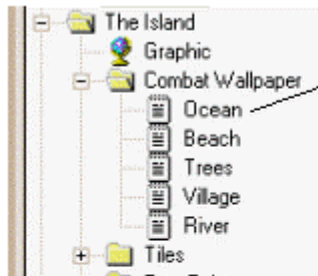
The Tileset (and default wallpaper) for this tile



Click on the dots to toggle whether the tile can be seen through from that angle (the blue dot on the wall shows that you cannot from that side). Click on the diagonals to toggle whether the tile blocks movement from that angle (the yellow line on that same facing means that the wall does block movement on that side)

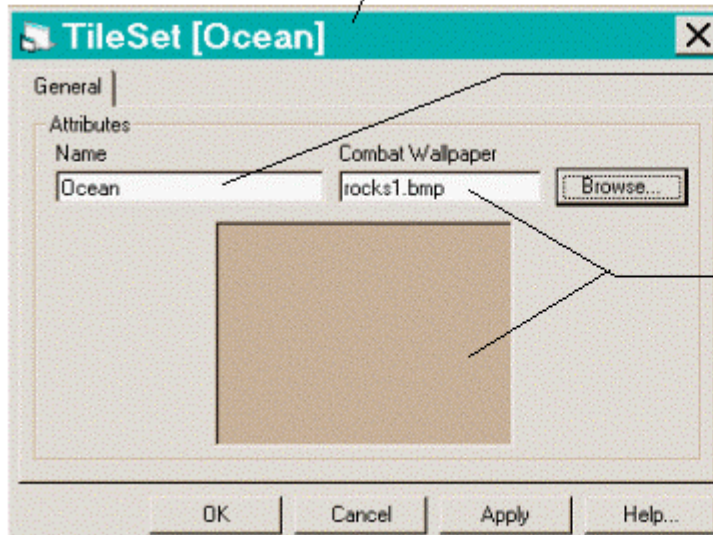
The scroll bar enables you to pick a tile graphic.

Default Combat Wallpaper (Scenery)



Right-click this!

Note: Tilesets are simply a way of grouping tiles and assigning default combat wallpaper for them. In this example, we have water type tiles for ocean, sand type tiles for beaches, and so on. It's helpful while painting the map to be able to select a specific set so that you don't have to scroll through the whole set when you're painting the map.



Once a tileset is created, individual tiles are assigned to it within the tile's properties (see screen 8)

The combat wallpaper bmps are stored in the WALLPAPER directory. Normally, a thumbnail preview of the wallpaper displays in the box, but my version is acting buggy today. Go figure.

Tome

Tome: Tossport

- Tossport Island** (Right Click this!)
- Party**
- Journal**
- Factoids**
- Triggers**
- Areas**

The other tabs are just a different way to access the other Tome folders

Unimplemented

Tome Properties

General | Party | Journal | Factoids | Triggers | Areas

Description

Name: ☐ On Adventure

Comments:
Authors: Adam and Dan
Other Contributors: None

Starting Location

Area:
Map in Area:
EntryPoint on Map:

Current Location

At X: At Y:

Party AvgLvl Turn Day Moon Year

5	2	45	20	6	1215
---	---	----	----	---	------

Turn, Day, Moon, and Year are setting the world's date and time

Unless you're editing an "in-play" tome (one in the World directory), the party's current location isn't useful

Party shows the maximum number of party members allowed. AvgLvl shows the average level for party members allowed. In this example, you could have 5 2nd level characters, 2 5th level characters, etc.

Tome Comments are shown before play, when the player is selecting which tome to play. Comments are a useful place to keep a version number on your tome during development. When the tome is ready to publish, put a short description of the tome in, and credit anyone who contributed stuff

OK Cancel Apply Help...

Triggers

Looking at a screen shot is not going to make you an instant expert on triggers. You'll find that the tutorial videos on triggers helpful, and the Encyclopedia Arcana contains reference info.

The screenshot shows the 'Creature Trigger [SpitPoison]' window. It has several sections: 'Description' with 'Event Timing' set to 'On-Attack' and 'Duration' set to 'Infinite'; 'Name' set to 'SpitPoison'; 'Comments' field; 'Class' checkboxes for various traits like Curse, Poison, Evil, Fear, Magic, Lunacy, Pride, Revely, Wrath, Greed, Lust, and Dreamer; and a 'Statement Editor' at the bottom left. The 'Statement Editor' shows an 'If' command with 'CreatureNow' as the object, 'RangeToTarget' as the property, and '> Pos 002' as the condition. The main 'Statements' pane on the right contains a script for a poison attack. Annotations with leader lines point to various parts of the interface.

Timing determines when the trigger fires

Don't check this unless you're creating a skill for player characters

Determines how often the trigger executes before it destroys itself.

This is the easy way of creating runescript. Select a command in the upper-left droplist (it shows the IF command here), select options from the other droplists that appear, and then click the update button.

This window shows the runescript. You may also type commands directly instead of using the statement editor to the left. If you enter a command incorrectly, you are notified.

This window also supports cut and paste to text editors and other triggers. Ctrl-C to copy, Ctrl-X to cut, and Ctrl-V to paste. Weenies may use the buttons instead.

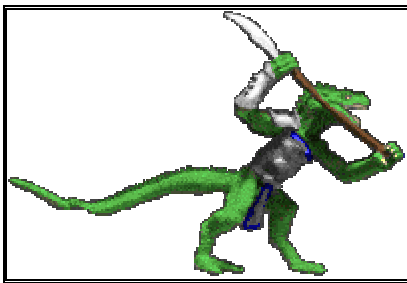
Search and Replace

After you've been working on a tome for a while, you might forget where you put something. Suppose you just remembered to give a key to the Night Watchman, but you cannot remember which encounter has the Night Watchman in it! Bring up the Search window, type *night watchman* in the Find What box, click *Creatures* as the search object, then click the *Find Next* button. The property sheet for the Night Watchman is opened.

Mistake? What Mistake? Here's a much better scenario. Suppose you're like me, and made a healing potion that didn't really function quite right (hard to mess up a simple healing potion, but I did it, thank you). Also like me, you compound the problem by saving the buggy healing potion in the library, and pasting a copy of the buggy healing potion in a dozen or so places in a huge tome.

Fixing this is easy:

1. Fix one of the healing potions. Then, in the Area window, right-click the fixed potion and *Copy* it
2. Select *Search* from the Search menu and type *Healing Potion* in the Find What box.
3. Click *Items* as the Search object type. Because the Healing Potion was one of the last 8 objects you copied, it will appear as one of the choices in the Replace With box.
4. Choose the Healing Potion in the Replace With box, then click *Replace All*. You've just fixed every buggy healing potion in the area. If there is more than 1 area in the tome, you must repeat the search after accessing the area.

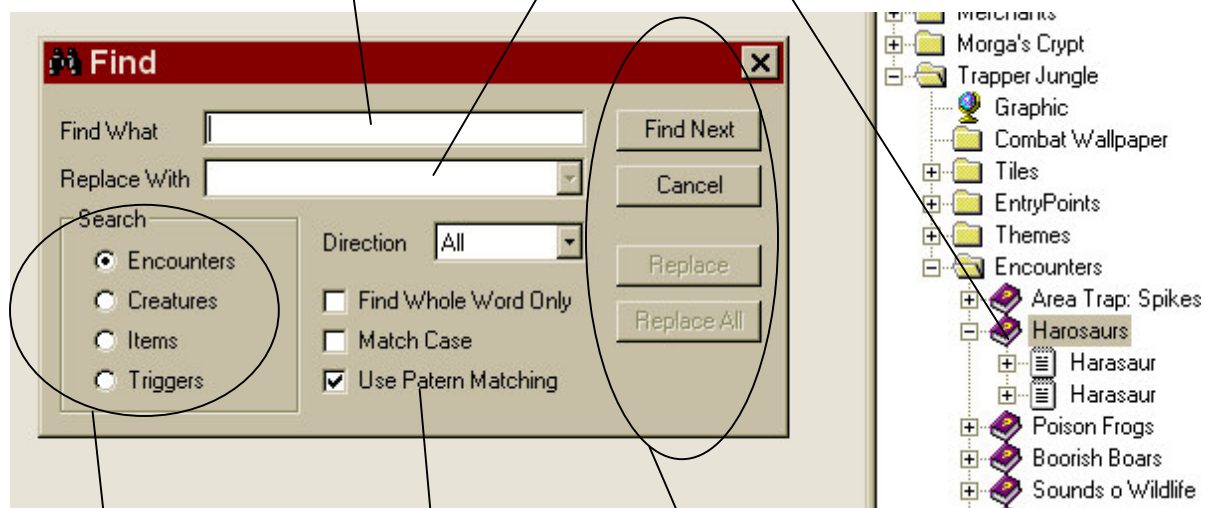


Search and Replace? Replace my glorified knife on a stick with a flamethrower, and I'll show you how I treat dungeon delvers.

Search and Replace

Type all or part of the name of what you're searching for.

Before an object shows up in the *Replace With* box, you must right-click and *copy* the object in the area window



Select the object type you're searching for.

Spelling, shmelling. It works.

Find Next brings up the property sheet on the next matching object.

Replace does the same, and then prompts you whether you want to replace the match.

Replace All just does it, no property sheets are shown. Be careful though – there is no undo function.

Encyclopedia Arcana

Before you delve into the Encyclopedia Arcana, you should play RuneSword extensively and watch the Creator videos. The Encyclopedia Arcana is a reference work, so unless you're needing sleep, you probably shouldn't attempt to read it straight through.

That said, there is more info on the inner workings of RuneSword here than anywhere else. You should find the introductory sections on Trigger Events, Properties, and RuneScript especially helpful.

Arcane bits include the following:

Properties Almost every aspect of every object in RuneSword is exposed for you. This is a catalog reference of those properties.

Trigger Events Triggers are the objects that can move mountains in this game. They're fire when certain events occur during play, and this section explains all.

RuneScript The programming language at the heart of the Creator.

Sub-Triggers An unusual type of trigger that remains dormant until you "bring it into play."

Variables The local and global variables.

Elegance in Dialog The finer points of dialog and how vices and skills affect it

Encounter Reactions A few pointers on how Encounters affect the Creatures within them.

Locks How to lock doors and containers, and how they're opened

Traps Trap construction in ways that Grimtooth never thought of

Under the Hood Advanced topics that deal with creating your own races, skills, maps, etc.

As you peruse the Encyclopedia Arcana, it's often a good idea to run the Creator program, open up a Tome, and flip back and forth to look at real examples.

Properties

Introduction

Properties are the characteristics of creatures, items, encounters, etc. The health of a character, the value of a gem, the bonus to hit of a sword, what time it is – these are all properties.

The most commonly accessed properties, such as a character's name or health, appear on the property sheets (as shown in the Creator Guide). All properties, whether they're on a property sheet or not, can be accessed in script.

It's very common to check properties for all sorts of reasons. If it takes at least 14 Strength to wield a magic battle-axe, well, the battleaxe needs Pre-Attack trigger that checks the attacker's Strength property. If only female characters are enchanted by a Satyr, you need to check the Sex property. And so on.

Of course, the power-hungry among you are saying, "*Check* the properties? I'm a Creator! *Change* the properties is more like it! I'm going to permanently lower Sir Valliant's strength to 1! Bwa ha ha!" The power is yours.



Nothing like eye-stalk blasts to change a character's health property.

In RuneScript, the syntax is Object.Property. For example, look at the following code:

```
'If the active character's agility is higher than 16, he dodges  
'a boulder. Agility is a property of Creature objects.  
,
```

```
If CreatureNow.Agility > POS 16 Then  
    DialogShow "You successfully dodge the bouncing boulder"  
End If
```

So Which Objects Have Properties?

- ❖ **Creatures** CreatureNow, CreatureTarget, CreatureA, CreatureB, CreatureC
- ❖ **Items** ItemNow, ItemTarget, ItemA, ItemB, ItemC
- ❖ **Triggers** TriggerNow, TriggerTarget, TriggerA, TriggerB, TriggerC
- ❖ **Tiles** TileNow
- ❖ **Encounters** EncounterNow
- ❖ **Tomes** Tome
- ❖ **Maps** Map

Notice that the Creature, Item, and Trigger objects come in several flavors: Now, Target, A, B, and C.

Now - objects are the active ones. For example, on Sir Valliant's turn, Sir Valiant is the CreatureNow, and the sword he's holding is the ItemNow.

Target - objects are those that an active object is targeting. For example, when Sir Valiant attacks a ghoul, the ghoul is the CreatureTarget. Or when Merlin uses his wand to enchant a shield, the shield is the ItemTarget.

A, B, C - objects are handles. They're used when you need to access objects other than the active or target objects. During Eleric's turn, for example, you may want to know whether the Ghost Knight is still alive or how many zombies are still attacking. See *Weird Stuff with Objects* for more info on handles.

Creature Properties

.ActionPoints - This is the current amount of action points a Creature has in combat. Most weapons and skill require a certain amount of action points to use. It's represented by the set of lights at the top of the combat screen. The initial value is kept in the ActionPointsMax property.

.ActionPointsMax - This value is computed from the following formula:
 $\text{AVG}(\text{Intelligence} + \text{Agility} + \text{Strength}) + \text{ExperienceLevel}$. A typical value is 12-15 for 1st Level Creatures. At the start of each Combat round, the Creature's ActionPoints is set to their ActionPointsMax value. You cannot change this value.

.Agility - This is one of the three basic Creature's stats, the other two being Intelligence and Strength. A normal human would have a score of 10 in each, but heroic characters start out a little better than that. For every 3 points of Agility above 10, the Creature gets a +1 to on the "To Hit" roll in combat (e.g. 13 is +1, 16 is +2, 19 is +3, etc).

.AgilityBonus - Normally 0, and it gets reset to 0 on the TurnCycle event. This affects the Agility property for 1 turn. Like all bonuses, this property is normally modified in a Pre-Turn trigger.

.ArmorProtection [1, 2, 3, 4, 5, 6, 7, 8] - Checking the ArmorProtection shows you the % normal (e.g. weapon) damage is reduced by equipped armor when a blow lands at that location. Some creatures have tough hides, or hard bodies that reduce damage -- see the BodyProtection property. This property is read-only; to change it, you must access the armor itself.

.AttackBonus - Normally 0, and it gets reset to 0 on the TurnCycle event. This affects the "To Hit" dice roll for 1 turn. Like all bonuses, this property is normally modified in a Pre-Turn trigger.

.BodyPart [1, 2, 3, 4, 5, 6, 7, 8] - This shows the name of the body part in a hit location. For a normal humanoid in RuneWorld, 1-6 are "Body," 7 is "Head," and 8 is "Lucky." By the way, body armor protects locations 1-4, a shield protects 5-6, and a helm protects 7. "Lucky" is on location 8 because no armor can cover that location -- therefore, the hit was a lucky one that will do full damage.

.BodyProtection [1, 2, 3, 4, 5, 6, 7, 8] - Checking the BodyProtection shows you the % normal (e.g. weapon) damage is reduced by the target's tough hide or hard body when a blow lands at that location. Some creatures wear armor that reduces damage -- see the ArmorProtection property.

.Bulk - The current amount of bulk the Creature is carrying. It is equal to the sum of the bulk of all non-equipped Items in the Creature's inventory. You cannot change this value.

.Comments - Text associated with a Creature. It is not displayed anywhere in RuneSword while playing.

.DamageBonus - Normally 0, and it gets reset to 0 on the TurnCycle event. This affects the combat damage when the creature hits for 1 turn. Like all bonuses, this property is normally modified in a Pre-Turn trigger.

.Defense - The number required to hit the Creature. By default, this is 13 for all creatures. This number is often altered by armor, and sometimes by spells or skills. You cannot alter this directly. If you want to change a creature's Defense, use the DefenseBonus property.

.DefenseBonus - Normally 0, and it gets reset to 0 on the TurnCycle event. This affects the Defense number (the value needed by others to hit this creature) for 1 turn. Like all bonuses, this property is normally modified in a Pre-Turn trigger.

.ExperienceLevel - This value is probably the best measurement of how dangerous a character is in combat. When an experience level is gained, the character is awarded a certain number of skill points to spend on enhancing skills. To change this value, you must use the AwardExperience statement.

.ExperiencePoints - The amount of experience points gained through great deeds and slaying monsters. When enough experience points are gained, the character gains an experience level. It's cumulative, taking 1000 to gain level 2, 3000 for level 3, 6000 for level 4, 10,000 for level 5, 15,000 for level 6, and so on. To change this value, you must use the AwardExperience statement.

.Greed - A vice associated with sentient creatures, normally ranging from 1 – 20 with a lower amount being better (less of the vice). The normal value for most characters is around 18. To succeed on a saving throw vs. greed, a character must roll this value or more on a 20-sided die. This saving throw is made in applicable situations, and also against spells from the Greed school.

.HealthMax - This is a Creature's maximum health. Because adventurers are heroic, their life force averages 40-50 at level one. First level creatures, on the other hand, average about 10-15 HealthMax.

.HealthNow - Represents a creature's life force. When this drops to 0 or below, the creature dies. A graphic representation of HealthNow is shown next to the character's portrait.

.IsAgressive - Unimplemented?

.IsDMControlled - If true (1), the creature is controlled by the computer in combat. Monsters are always computer controlled. You can make party members computer controlled as well.

.IsFriendly - Determines if a Creature is an enemy (0) of the Party or a friend (1). When the Party enters an Encounter with all Creatures having IsFriendly = 1, the Party is not presented with a Fight option.

.IsFrozen - This value is 1 if the Creature is Frozen, else it is 0. Frozen creatures lose their turn in combat. At the beginning of every TurnCycle, all creatures are "unfrozen" (i.e. Creature.IsFrozen is set to 0).

.IsGuarding - Determines if a Creature is protecting the Items in an area or not. IsGuarding is 1 if they are protecting, and 0 if not. If there is at least one Creature with IsGuarding = 1, the Party cannot search for Items nor target any Item in the Encounter with a spell. By default, Creatures have IsGuarding = 1.

.IsMale - This value is 1 for males, 0 for females (sexist programming, I know). If you really want to irk your players, just make a wand of gender changing. On a serious note, you should check this property to set up possible romantic entanglements in towns (also check the ol' Lust property). Romance is vastly underused in CRPG's, and it really elevates a tome above the commonplace hack n slash. Having an attractive member of the opposite sex make a pass at you in a pub is first-rate fantasy for most folks.

.IsSpellCaster - Returns a 1 if true (creature has a Sorcery skill), or 0 if false. Sometimes this property is checked in On-Attack "Brains" triggers, because some creature may enjoy put a high priority on smacking spell casters.

.IsType [Animal, Bird, Blob, Aquatic, Huge, Sentient, Insect, Large, Magical, Medium, Reptile, Small, Tiny, Plant, Undead] - Creatures are classified into one or more types, each is 1 if true, 0 if false. An Undead Dragon might be IsTypeHuge, IsTypeReptile, IsTypeUndead and IsTypeMagical. This comes in handy for magic that affects certain types of creatures in a special way. For example, a spell that only affects undead would have a statement like this: If CreatureTarget.IsTypeUndead = Global.True Then

.IsUnconscious - This value is 1 if the Creature is unconscious, else it is 0. Unconscious Creatures lose their turn in combat. Unlike IsFrozen, IsUnconscious Creatures are never awakened automatically. A spell or some other magical effect must change this property to 0.

.Lunacy - A vice associated with sentient creatures, normally ranging from 1 – 20 with a lower amount being better (less of the vice). The normal value for most characters is around 18. To succeed on a saving throw vs. Lunacy, a character must roll this value or more on a 20-sided die. This saving throw is made in applicable situations, and also against spells from the Lunacy school.

.Lust - A vice associated with sentient creatures, normally ranging from 1 – 20 with a lower amount being better (less of the vice). The normal value for most characters is around 18. To succeed on a saving throw vs. Lust, a character must roll this value or more on a 20-sided die. This saving throw is made in applicable situations, and also against spells from the Lust school.

.MapX, MapY - The current location of the Creature on the Map. Map coordinates start at 0 (e.g. (0,0) is the upper left corner of the Map). You cannot change these values. If you want to teleport the party, use the MoveParty Statement. If you want to know the creature's location on the combat screen, see Row, Col.

.Name - The name of the Creature. Any change here should be temporary, i.e. a polymorph trap might change a creatures name to "Toad" for a few turns.

.Pride - A vice associated with sentient creatures, normally ranging from 1 – 20 with a lower amount being better (less of the vice). The normal value for most characters is around 18. To succeed on a saving throw vs. Pride, a character must roll this value or more on a 20-sided die. This saving throw is made in applicable situations, and also against spells from the Pride school.

.Protection [Blunt, Fire, Holy, Magic, Sharp, Evil, Cold, Mind] - The damage a creature absorbs from an attack may be normal, or one of these 8 types of damage. Some creatures, either naturally or via a magic item, may be resistant or especially vulnerable to certain types of damage. If the creature is resistant, DamageModifier returns a positive number (10, 20, 30; 100) showing the % that the damage is reduced. If the creature is vulnerable, DamageModifier returns a –200 for double damage or a –300 for triple damage.

.ProtectionBluntBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when blunt type combat damage hits the creature. The values 10, 20, 30; 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when any type of combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionColdBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when cold type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionEvilBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when evil type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionFireBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when fire type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionHolyBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when holy type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionMagicBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when magic type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value –100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionMindBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when mind type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value -100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.ProtectionSharpBonus% - Normally 0, and it gets reset to 0 on the TurnCycle event. For 1 turn, this affects the percentage of damage taken when sharp type combat damage hits the creature. The values 10, 20, 30 & 100 absorb that percentage of damage. Value -100 means the creature suffers double damage, -200 means triple damage. This property is normally modified in a Pre-Turn trigger.

.Race - Shows the land of origin for a creature, which in RuneWorld, is the same as the character's race (i.e. Argosans are a different race than Talantans). The RuneSword engine does not use it, except during character creation. Of course, if you have a creature that reacts a certain way to certain races (i.e. my Sprites despise Moreans and attack them on sight), this is the property to check.

.RangeToTarget - This is the number of tile spaces between CreatureNow and CreatureTarget. It's only applicable in combat. You cannot change this value during play. It's useful to determine how far away one Creature is from another.

.Revelry - A vice associated with sentient creatures, normally ranging from 1 – 20 with a lower amount being better (less of the vice). The normal value for most characters is around 18. To succeed on a saving throw vs. Revelry, a character must roll this value or more on a 20-sided die. This saving throw is made in applicable situations, and also against spells from the Revelry school.

.Row, Col - The Creature's current location on the Combat screen. The Combat grid is 11 squares wide and 8 squares high. Coordinate (0,0) is the upper left corner of the Grid.

.RuneQue [1, 2, 3, 4, 5, 6] - Spellcasters can queue up to 6 at once to cast a spell. Rune1 is the first rune queued up, Rune6 is the last. When you check this property, it'll return a value that identifies which rune is queued at that spot as follows: 0 - None, 1 - Blood, 2 - Bile, 3 - Oil, 4 - Nectar, 5 - Fire, 6 - Earth, 7 - Water, 8 - Air, 9 - Time, 10 - Moons, 11 - Suns, 12 - Space, 13 - Insect, 14 - Man, 15 - Fish, 16 - Animal, 17 - Twilight, 18 - Abyss, 19 - Dreams, 20 - Eternium.

.SkillPoints - The total of unspent SkillPoints. Skillpoints can be spent at any time to increase the character's ability in a chosen skill. Characters receive 10 or so SkillPoints with each ExperienceLevel gained. You should normally update this using the AwardExperience Statement, but for special deeds accomplished (or evil acts committed), you can update it directly.

.Status - Text describing the current status of a creature: Dead, Unconscious, Frozen, And Exhausted. You cannot change this directly (but naturally, if you, say, knock a creature's health to 0, his status will change to Dead!). This is the same text that is displayed in Inventory for a Creature.

.Strength - This is one of the three basic Creature's stats, the other two being Intelligence and Agility. A normal human would have a score of 10 in each, but heroic characters start out a little better than that. For every 3 points of Strength above 10, the Creature gets a +1 to damage in combat (e.g. 13 is +1, 16 is +2, 19 is +3, etc).

.StrengthBonus - Normally 0. This affects the Strength property for 1 turn. Like all bonuses, this property is normally modified in a Pre-Turn trigger.

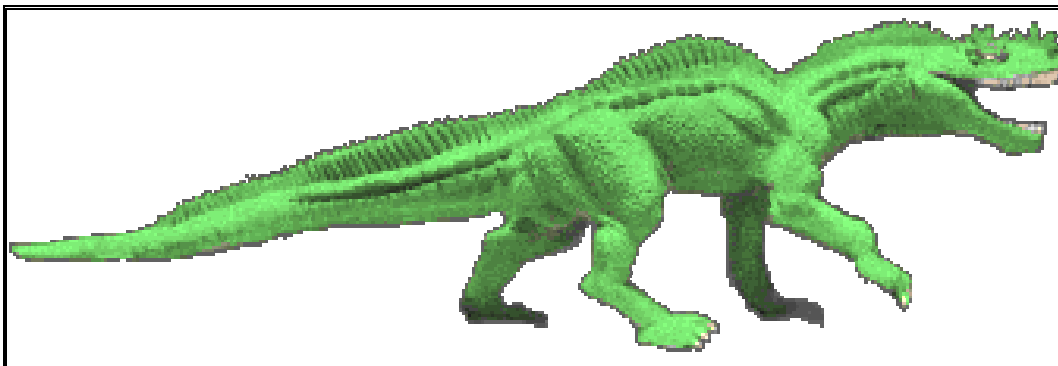
.Weight - The current amount of weight the Creature is carrying. It is the sum of the weight of all Items in the Creature's inventory. You cannot change this value.

.WeightMax - This is the maximum amount of weight the Creature can carry. It is equal to the Creature's Strength x 15. You cannot change this value.

.Intelligence - This is one of the Creature's basic stats, the other two being Agility and Strength. Stats normally range from 0 to 20 (heroes would be around 12-14). Intelligence affects how well the Creature can detect and remove traps, how often spells fail, and many "awareness" checks during play.

.IntelligenceBonus - Normally 0, and it gets reset to 0 on the TurnCycle event. This affects the Intelligence property for 1 turn. Like all bonuses, this property is normally modified in a Pre-Turn trigger.

.Wrath - A vice associated with sentient creatures, normally ranging from 1 – 20 with a lower amount being better (less of the vice). The normal value for most characters is around 18. To succeed on a saving throw vs. Wrath, a character must roll this value or more on a 20-sided die. This saving throw is made in applicable situations, and also against spells from the Wrath school.



Reptile yes, sentient no.

Item Properties

.ArmorProtection - How much the item reduces the damage (as a %) to the wearer. For example, 10 points of damage coming through a 50 Armor shield results in 5 points of damage taken by the character.

.Bulk - How large/awkward the Item is. A Creature can only carry 100 points of bulkiness. Items that are worn or equipped do not count toward this total . See the *Capacity* and *IsSoftBulk* properties.

.CanCombine - The values are 1 for yes and 0 for no. A yes means that the item automatically combines with other Items of the same name. The best example of this is gold pieces – if you pick up a pile of 15 gp, then pick up 5 more, your inventory will show 1 item, which is 20 gp. This property has nothing to do with combining special items, like a hilt and a blade that combine to make a sword (for those, see the *Combine* event in the Triggers help doc).

.CanWear - [Shield, Backpack, Mask, Boot, Ring, Glove, Pants, OneHand, Necklace, Body, Belt, Bracelet, TwoHand] Whether the Item can be worn at that body location. Each item is worn or equipped at a certain location. By default, an Item is set as One-Hand type—the CanWearOneHand property is true.

Remember, all items that a character has are considered "carried," but *equipping* an item means you're wearing or holding it. Creatures can equip only 1 item of each CanWear location, except for the following:

Ring (up to 10)
Glove (up to 2)
Bracelet (up to 2)

Note that the OneHand (swords, wands, etc.) and TwoHand (bows, two-handed swords, etc.) type items are special because when such items are equipped, they're the active item--see the IsInHand property. When the character's turn comes, these become the ItemNow object.

You may be looking at the above list and say "I can equip 2 glove type items, and 2 bracelet type items, so why can't I equip 2 One-Hand items?" Even though it makes sense to do so (ya got two hands after all), it's a programming and play-balancing hassle. Even in pen&paper RPGs, few DM's allowed players to wield an ice wand in the left hand and swing a blade with the right. The main reason people want it is for wielding two weapons at once, and for that, we've provided the Ambidex skill to get around this limitation.

There are two special rules: you cannot equip a OneHand and a TwoHand item at the same time; and you cannot equip a TwoHand and a Shield item at the same time.

.Capacity - This is how much Bulk a container type Item can hold. All items have bulk and weight values. Characters can carry up to 100-bulk total, but equipped items do not count against this total. Bags and other containers with a capacity value enable characters to carry more.

For example, you could make a bag with a Capacity of 20. Any item put into the bag does not count against the character's capacity limit while it's inside. The bag itself still has a bulk value, but it's relatively small. If an Item has 0 Capacity, it's not a container. Note that there is also a weight limit for carrying stuff, and capacity doesn't affect this. See `IsSoftBulk`.

.Comments - This is just text associated with an Item. It is not displayed anywhere in RuneSword while playing, unless the `UseAsDescription` property is true. You could use this to make notes to yourself about the item, or you use this as a text variable.

.Count - This is the number of Items this single Item represents. For example, to make a pile of 30 gold pieces, you make a "Gold Piece" item and set its Count property to 30 (the engine will automatically display the name as plural). But even though this will be displayed as "30 Gold Pieces" on the inventory screen, the true name of the item is still "Gold Piece."

.DamageDice - This is a number representing the amount of damage the Item does in combat. Of course, in the Creator's interface, you pick the damage dice from a droplist. But the value is actually stored as follows:

0 no damage (the default value)
1-5 from 1-5 four-sided dice
6-10 from 1-5 six-sided dice
11-15 from 1-5 eight-sided dice
16-20 from 1-5 ten-sided dice
21-25 from 1-5 twelve-sided dice
26-30 from 1-5 four sided dice with +1 per die

and so on to ridiculous amounts of damage. For the record, the highest value is 250, which is 5 twelve-sided dice with +9 per die.

.Family - This is name of the family the Item belongs to. Family names include Locked, Jammed Shut, Fragile, Durable, Money, etc. There are some special Families including Random Weapon and Random Armor, which are covered in the Random Generator section.

.IsAmmoArrow - This is a 1 if the item is an arrow and 0 if not. Arrows are ammo for Bows. See `IsShooter` for a full discussion.

.IsAmmoBullet - This is 1 if the item is a bullet and 0 if not. Bullets are ammo for Guns. See `IsShooter` for a full discussion.

.IsAmmoStone - This is 1 if the item is a stone and 0 if not. Stones are ammo for Slings. See IsShooter for a full discussion.

.IsEquipped - This is 1 if the item is currently being worn and 0 if not. Items that are being carried are not equipped.

.IsInHand - This is 1 if the Item is currently in hand, else it is 0. InHand Items are those that are used to attack with in combat and can be operated with the Use command in RuneSword. They are the ItemNow when the creature's turn comes up. You can only have one Item in hand at a time. Only CanWearOneHand and CanWearTwoHand type items are considered InHand when equipped (see .CanWear).

.IsRangeShort - This is 1 if the item can only strike creatures within short range (within 1 space on the combat screen). Items are short range by default.

.IsRangeMedium - This is 1 if the item can only strike creatures within medium range (within 2 spaces on the combat screen).

.IsRangeLong - This is 1 if the item can only strike creatures within long range (3-10 spaces away on the combat screen). Items that are long range but *not* shooters are considered "thrown" items. Thrown items once used are destroyed. See IsShooter for a full discussion.

.IsShooter - This is 1 if the item shoots ammo and 0 if not. You cannot change this value during play. A typical bow will have the following properties set to true: IsRangeLong, IsShooter, and IsShooterBow. An arrow is a separate item with the IsAmmoArrow set to true, IsRangeShort set to true, Count set to the # of arrows, and DamageDice set to 3 for 3d4 damage (the ammo will be inside a container of some sort). Remember, the ammo determines the amount of damage done by a shooter.

.IsShooterBow - This is 1 if the item is a bow and 0 if not. Bows use Arrows for ammo. See IsShooter for a full discussion.

.IsShooterGun - This is 1 if the item is a gun and 0 if not. Guns use Bullets for ammo. See IsShooter for a full discussion.

.IsShooterSling - This is 1 if the item is a sling and 0 if not. Slings use Stones for ammo. See IsShooter for a full discussion.

.IsSoftBulk - There are two types of Items than can carry other Items: Hard (IsSoftBulk=0) and Soft (IsSoftBulk=1). Hard Items have their own Bulk regardless of their contents. For example, a Box can have a Bulk of 25 and a Capacity of 100. This allows you to carry more Bulk than normal if you put stuff inside the box. Soft Items have a Bulk equal to the largest Item contained inside. For example, a Bag can have a Bulk of 1 when empty and a Capacity of 100. You put a Sword inside with a Bulk of 10. Now the Bag has a Bulk of 10.

.IsType - [Air, Blunt, Earth, Fire, Holy, Disease, Magic, Sharp, Evil, Cold, Illusion, Water, Fear] This is 1 if the item is of a particular type and 0 if not. For armor, this is the type of damage it protects from (halves damage in addition to the armor's normal protection). For any other type of item, it is for reference. This plays in if, say, a Sharp type sword strikes a character is vulnerable or resistant to Sharp type damage.

.Key - This defines the key that is necessary to open the item if the container is locked or jammed shut. The key (see the Family property) must have a matching value in this property. type itemvalue of the key if the item is a key (see Family below). There are 31 possible key values numbered 1 to 31.

.MapX, .MapY - This is the current location of the Item on the Map. Map coordinates start at 0 (e.g. (0,0) is the upper left tile of the Map). You cannot change these values.

.Name - The name of the item. The only weirdness about this is items with the Count property set.

.Defense - This value modifies the creature's Defense value. Without any modifiers, creatures have a defense value of 13, so attackers need 13+ on a 20-sided die roll to hit them. The Defense value can help (add to the Defense number) or hurt (subtract from it), but only when the Item is equipped. In RuneWorld, unlike some games, armor usually lowers the Defense number because wearing such gear makes it more difficult to evade blows (armor makes up for this deficiency by reducing the amount of damage taken). For example, Field Plate Armor might have a Defense value of -3, making you easier to hit.

.ToHitBonus This is the amount added to your attack roll when attacking with the Item. For example, a Sword with a ToHitBonus of 1 would add 1 to the 20-sided dice when rolling to hit.

.UseAsDescription - When true (1) the Comments for this Item is used as the description when a character examines it.

.Value - This is the value of the Item in monetary units. For Items with a Count other than 1, this is the value of *all* the Items. For example, 1 platinum piece might be worth 5 gold pieces, your standard monetary unit. Suppose you want to create a pile of 25 platinum pieces, each with a value of 5. You'd create a "Platinum Piece" item with a Count property of 25 and a Value property of 125.

.Weight - This is the weight of the Item. The weight limit for a creature is Strength * 15, so a creature with 13 Strength can carry up to 195 weight. Unlike Capacity, the weight limit affects all items, whether they're equipped or not. For Items with a count other than 1, this is the total weight of all the Items.

Trigger Properties

.ByteA, .ByteB, .ByteC - These are variables that stay with the Trigger even over several executions (in programmer terms, these are static variables). Suppose you create a Wand that has a 1% chance of exploding when used. Moreover, each time it's used successfully, this chance increases 1%. The best place to store this chance is in one of these Byte variables. ByteA, B and C can be set to anything from 0 to 255.

.Comments - This is just text associated with a Trigger. It is not displayed anywhere in RuneSword while playing.

.IsTimed - Makes an item lose 1 turn (during the TurnCycle event) whether or not it's used. This property returns a 1 if true, else it is 0. For IsTimed to function, you must also set the Turns property to a positive number.

The Turns property is more commonly used with IsTimed set to false, because it is fairly common to give triggers a certain number of "charges" before they become useless. A teleport wand, for example, may work 4 times and then stop functioning. Giving the wand a "teleport" trigger with the Turns property set to 4 does this.

IsTimed is set to true only when you want the trigger to disappear after a certain amount of time has passed, no matter what. For example, a character might heat up a flask of oil, and you give the oil a "boiling hot" trigger with IsTimed set to true and Turns set to 3. The character has to use it within 3 turns or the oil reverts to normal.

.IsCurse, .IsPoison, .IsMagic, .IsEvil, .IsFear - This is 1 if the Trigger is of the specified type and 0 if not. These should be set to true where appropriate. Doing so enables other spells and effects to function correctly.

Suppose that you create a hag that curses a character. Has there ever been a fantasy story where you could get away with disrespecting an old woman? You copy a "Bad Luck" trigger to the character that gives him a -5 to hit on up to 7 attacks per day, randomly chosen. You should remember to check the *Is Cursed?* property on the property sheet. That way, a remove curse spell will work. The spell removes any triggers assigned to the Creature that have IsCurse = true.

Note that these DO NOT affect the damage sustained by a character, even when the character has a resistance or vulnerability to these properties. Resistances and vulnerabilities affect damage from a RollDamage or CombatAttack statement, where you can set what type of damage it is.

.IsDreamer, IsGreed, IsLunacy, IsLust, IsPride, IsRevelry, IsWrath - This is 1 if the Trigger is of the specified vice and 0 if not. This is used by the RuneSword engine to determine what school a Trigger belongs to and impacts saving throws. See Global.IsAttackTrigger.

.IsSkill - This is 1 if the Trigger is a skill and 0 if not. Skill Triggers attached to Creatures are listed in that Creature's inventory and can be raised in power by spending skill points. IsSkill has no effect on Triggers unattached to Creatures.

.IsTrap - This is 1 if the Trigger is a trap and 0 if not. IsTrap Triggers attached to Items or Encounters can be searched for by the Party. If found, they have a chance of removing the Trap (destroying the Trigger) or, failing that, a chance to "trip" the Trigger. If the latter happens, the Trigger is executed immediately.

.Name - This is the name of the Trigger.

.SkillPoints - The number of skill points spent on the Trigger and determines what "level" the skill is currently at. It only applies to IsSkill Triggers.

.TriggerType - The name of the type of Trigger. For example, "Pre-Search" or "Post-Turn". You cannot change this value during play.

.Turns - The number of times the Trigger will execute. Each time the Trigger executes, 1 is subtracted from Turns. When it reaches 0, the Trigger destroys itself. If you want a Trigger to execute infinite times, set Turns = 0.

.SkillLevel - This is the current level of a skill Trigger. You cannot change this value during play.

Tile Properties

.CanSee [Left, Right, Up, Down] - Each tells you if the Party can see a direction, 0 if yes, 1 if not. You cannot change this value during play. ??? opposite day?

.ChanceToOpen - This is the percent chance to open a door. A 0 means you cannot open the door. To see if a door exists in this tile, check the DoorTileName property.

.IsBlocked [Left, Right, Up, Down] - Tells you if the Party can move a direction. The value is 0 if no, or 1 if yes. You cannot change this value during play.

.Key - The value of the key necessary to open the door tile. A key value of 0 means the door is not locked. To see if a door exists in this tile, check the DoorTileName. Setting Key to 0 unlocks a locked door, but it does not open the door; the party has to use the Open command to do that.

.Movement Cost - The amount of time it takes to move through the Tile. A 1 is 1/10th of a turn, which is the default. You cannot change this value during play.

.Name - The name of all tiles in the stack (Bottom, Middle, and Top) separated by the "pipe" character (the "|"). You cannot change this value during play.

.Style - This is a text description of all Styles used in the current tile space separated by the "pipe" character (the "|"). You cannot change this value during play.

.Terrain Type - This is a text description of all Terrain used in the current tile space separated by the "pipe" character (the "|"). You cannot change this value during play.

.DoorTileName - This is the tile name of the door in the tile space. A door name of "<None>" means there is no door in this tile space. You cannot change this value during play.

Encounter Properties

.CanFight - This indicates if fighting is allowed in this Encounter. If set to 0, when the Party tries to fight, RuneSword will say "There's no reason to fight here." If set to one, fighting is allowed, which is the default.

.CanFlee - When the Party enters an Encounter and the value of CanFlee is 1 (true), an option to Flee is presented. If set to 0, no option to flee is given. The % chance to flee is determined by the ChanceToFlee property.

.CanTalk - When the Party enters an Encounter and the value of CanTalk is 1, an option to Talk is presented. If set to 0, no Talk option is given. Creatures must still have Conversations attached to talk.

.ChanceToFlee - The percent chance that the Party can flee a combat. 0 is no chance to flee, 100 is no chance of failure, which is the default. The CanFlee property must be true for this to have any effect.

.Randomize: Classification - The type of Encounter. Used only with the Tome Wizard. The default is <None>.

.FirstEntry - The text that appears when the Party enters the Encounter for the first time. If blank, a default description appears (e.g. "There are 12 Goblins here.") It is not displayed if the party returns to the encounter.

.HaveEntered - This indicates if the Party has entered the Encounter. If 0, the Party has not entered this Encounter before. If 1, they have.

.IsDark - If 1, this Encounter is dark. The light value of the Tiles controls how well lit it is. If 0, the Encounter is well lit (this overrides the light value of the Tiles).

.Name - The name of the Encounter. It is displayed when the mouse hovers over the Encounter if the ShowHint property is true.

.Randomize: ThemeName - The name of the Theme assigned to the Encounter. It's used when the Encounter is randomized. You can change it to any other Theme Name associated with the active Map.

.Randomize: AddCreatures - Indicates if Creatures should be added to the Encounter when it's generated. No is 0 and yes is 1. Creatures are created based on the assigned Theme.

.Randomize: AddItems - Indicates if Items should be added to the Encounter when it's randomized. No is 0 and yes is 1. Items are added based on the assigned Theme.

.Randomize: Description - Indicates if the Encounter Description should be replaced when it's randomized. If 0, the description is not replaced. If 1, the description is selected from those attached to the assigned Theme.

.Randomize: GenerateUponEntry - Indicates if the Encounter should randomize when the Party enters it for the first time. Used primarily for making a game different each time you play.

.SecondEntry - The text that appears when the Party enters the Encounter every time after the first. If blank, no secondary description appears.

.ShowHint - If true (1), previously visited encounter names are displayed as a "hint box" when you run the mouse over them on the map, and unknown encounters show question marks. If false (0), nothing appears for the encounter.

.CreatureCount - The number of Creatures -- alive or dead -- attached to the Encounter. You cannot change this value during play.



It was best to attack the Movaul with bows or slings. Up close, they were terrifying.

Tome Properties

.Comments - The creator's text description of the tome.

.Factoids - This property is the most commonly accessed one for tomes. A factoid is RuneScript's way of globally sharing information on the rare occasions it's necessary. It beats using global variables for storage because globals can be overwritten easily, and globals are reset when the tome is saved anyway. See the `AddFactoid` and `DestroyFactoid` in Miscellaneous Statements.

.MapX, MapY - The party's current location on the map (not the combat screen). This info is also available in character properties.

.MoveToX, MoveToY - The destination spot on the map that the player has clicked.

.Name - The name of the tome.

.TimeDay, TimeMoon, TimeTurn, TimeYear - The values for each of these "time" variables. You can rename the labels themselves with the global variables `timeday`, etc (see the [Variables](#) section).

Map Properties

.Comments - The comment text for the active Map. It isn't displayed anywhere by RuneSword and it is saved with the Map.

.Randomize: EncounterCount - When RuneSword randomly generates a Map in Creator or while playing, this determines the number of Encounters to create.

.Randomize: MapHeight - When RuneSword randomly generates a Map, this determines the height (the number of tiles in the Y coordinate). Maximum value of 255.

.Randomize: MapStyle - When RuneSword randomly generates a Map, this indicates the Style of Map which to generate: 0 Tetris Spin, 1 Cavern Sprawl, 2 Bubble Tubes, 3 Rectangles, 4 Huge Cavern.

.Randomize: MapWidth When RuneSword randomly generates a Map, this determines the width (the number of tiles in the X coordinate). Maximum value of 255.

.Randomize: Difficulty - Unimplemented.

.Randomize: ExperiencePoints - Unimplemented.

.Height - The height (Y coordinate) of the active Map. You can't change this.

.Left - The X coordinate of the upper left corner of the active Map. You can't change this.

.Name - The name of the active Map. It isn't displayed while playing, it's just for reference.

.Randomize: GenerateUponEntry - If this is 1, then when the party enters the Map, it will randomly generate. After the Map generates, this is automatically set to 0.

.Rune: [Blood, Bile, Oil, Nectar, Fire, Earth, Water, Air, Time, Moon, Sun, Space, Insect, Man, Animal, Fish, Twilight, Abyss, Eternium, Dreams] - The number of runes available for each rune type while the party is in the active Map. When spells are cast in combat, runes are drained. When the party leaves combat, the Runes are set back to these values. Outside of combat, the number of Runes are reset to these values every turn cycle.

So if you change these values, you are impacting the entire map for the duration of the adventure. Indirectly, you're also impacting combat on this map because these Runes are copied to the RunePool before combat begins.

This is a powerful way to set "high magic" and "low magic" areas in your world.

.Top - The Y coordinate of the current (on screen) upper left corner of the Map. You can't change this.

.Width - The width (X coordinate) of the active Map. You can't change this.

.IsOutsideMap - This is 1 if the active Map is an outside Map, 0 if not. Outside maps are subject to daylight/nighttime lighting effects. You can't change this.

Trigger Events

Introduction

Triggers and their Events - Triggers are objects containing snippets of RuneScript code. They enable you to do practically anything with the RuneSword engine. Every trigger is tied to an *event*. Various events occur as you play RuneSword. If your character decides to pick up a magic ring, you click the search button and then drag the ring to inventory--you've set off the Search event. The RuneSword engine looks for any triggers that are assigned to that event and activates them. Usually, there probably aren't any (after all, you're just picking up a magic ring). But if the creator decided to make 6 demons appear if anyone touches the ring, well, he can do it with a Search trigger.

Do I need triggers for everything? - Nope. Many objects work with just their properties set, like most weapons and monsters. However, if you want to create a spell-casting creature or give someone a key because they said the right thing to a guard, you need a trigger.

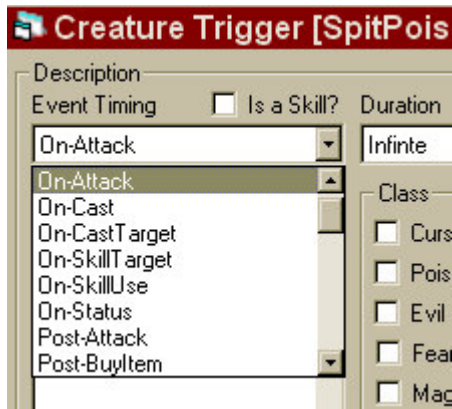
I don't want to write code! - You don't have to. You can cut, copy, and paste triggers anywhere you like, and all the normal, often used triggers are easy to reuse. A lot of Creators don't write and code at all, but they're good at tweaking the existing stuff (let's make that fireball spell do 80 points of damage!). You can be a tweaker, not a coder, and this document will still be useful.

Choosing the Right Object and Event for a Trigger

Triggers do not stand alone. They are usually attached to one of the 5 RuneSword objects: creatures, items, encounters, tomes or dialog topics. Simple logic usually dictates which object you choose, e.g. if you want a sword to glow when goblins are around, you attach the trigger to the sword item.

It can be a bit trickier to figure out the trigger's event property, which specifies when the trigger fires. The events you are permitted to use depend on what kind of object the trigger is attached to:

- ❖ **Creature Events** Attack, Cast, Status, SkillUse, TargetOfAttack, Examine, RollAttack, Turn, Turncycle, TargetOfApplyDamage, TargetOfUseOnCreature, PickLock, RollArmorChit
- ❖ **Item Events** Combine, UseInEncounter, UseOnItem, ApplyDamage, Attacked, CombatMove, Death, DropItem, Equip, Examine, RollArmorChit, RollAttack, RollDamage, SearchTraps, Split, Take, Turn, Turncycle, Unequip, PickLock
- ❖ **Encounter Events** Ignore, Listen, Combat, EnterEncounter, Search, SearchTraps, StepEncounter, Take, Turncycle, UseInEncounter, ExitTome, Open, Unlock
- ❖ **Dialog Event** Topic
- ❖ **Tome Events** EnterTome, ExitTome, TargetOfApplyDamage, TurnCycle



This whole section is about picking the right thing from this droplist. But hey, it's important.

The Active Only Rule

Events usually occur on the "active" objects. Characters and monsters take turns in combat, so the active creature is the one whose turn it is. When you're moving around on the map, the character with the yellow highlight is the active character. At one time, a character, the objects he's holding or wearing, and the encounter he's in (if any) are active.

For example, when a character casts a spell, only that character's Cast event fires (the Cast events of other creatures in the party or encounter remain dormant). Some events apply to multiple objects: when a character searches, the Search event for the encounter and the searching character fire.

Equipped vs. Unequipped - Active items include the items in hand and worn – triggers for unequipped items, ones that are merely carried, do not fire. And of course, only the current encounter's triggers fire. The triggers of all other encounters on the map are ignored until characters walk onto them, making them active.

Breaking this Rule - Exceptions exist to the Active Only rule. There are a few events that fire regardless of who is active: Turncycle, EnterTome, etc. Also, a few events fire on the target creature instead of the active creature: TargetofAttack, TargetofUseOnCreature, etc.

Sub-triggers are also an exception. Sub-triggers are triggers that are attached to a normal trigger instead of one of the five standard objects. Sub-triggers do not fire, even when their event occurs. See the [Sub-triggers](#) section for more info.

Specific Timing (Pre – On – Post)

Okay, so we've pounded it into your head that events are passing as you play the game. Each trigger has specific timing, the Pre and Post timing fires before or after an event. The special On timing makes the trigger run instead of the default engine event.

- ❖ **Pre** Fires just prior to an Event. Pre triggers are often used to modify the event (as in, "give the attacker a +2 to hit") Pre timing can even abort the event and any other triggers that fire on the event (as in, "you're stunned and don't get a turn").
- ❖ **On** Fires instead of an Event. Important wording, that "Instead Of" there. You are actually substituting for the event with On timing. A common example is an On-Attack trigger on a computer-controlled creature. Instead of using the default combat logic for its attack, the creature attacks using the RuneScript code in your attached On-Attack trigger (that hobgoblin is smart!).
- ❖ **Post** Fires just after an Event. Sometimes, you may want the Event to fire normally, but then make something happen immediately afterwards (as in, "sure you died, but now your Muse is offering resurrection - for a price).

Commonly Used Events

Probably 95% of all triggers use one of the events with the specific timing shown on this brief list. So don't worry about esoteric events like Post-RollArmorChit (great for a battle axe with a magical shield splitting effect) because you can look that kinda stuff up when needed. This Top 5 list shows the events/timing you should know by heart.

- ❖ **Pre-Turn attached to a Creature** Number one for a reason. When you want a certain character to get a +2 to hit on his next turn, or -3 to defense, or whatever. You can also attach pre-turn triggers to items, but it's not necessary for a bonus to hit or damage because that can be built in to the item.
- ❖ **On-Attack attached to a Creature** This event governs how computer-controlled creatures act in combat. It's unnecessary for simple fighters – the RS engine enables critters to fight competently with short and long-range weapons, no trigger needed. But that's it; no built in spellcasting brains or special attack/defense logic. Don't worry, though, some great On-Attack triggers are available in our libraries for cut and paste. You don't have to code such tricky stuff. Good On-Attack triggers containing attack logic are named "brains."
- ❖ **On-UseOnCreature attached to an Item** A classic. When you point that wand at someone, you want it to do something.
- ❖ **Post-Topic attached to a Topic** When you say something to someone, sometimes you want something to happen as a result. Could be an old man giving you a key - or maybe a curse.
- ❖ **Pre-Turncycle attached to an Encounter** In combat, this fires just before characters and monsters get their turns. After they go, it fires again. Outside of combat, this fires every 10 steps the party travels. This is good for stuff that affects the whole party, like damaging them whilst standing in a room of boiling ooze.

Creature Trigger Events

Note: The term "creature" includes both characters and monsters

Attack

Cast

CombatMove

Death

Examine

PickLock

RollArmorChit

RollAttack

RollDamage

SkillUse

Status

TargetOfApplyDamage

TargetOfAttack

TargetOfUseOnCreature

Turn

Turncycle

Attack (Timing: On)

When: On computer-controlled creatures when their turn comes up in combat (this event never fires on player-controlled characters, because they control their attack).

Why: This event is commonly used to give monsters "brains" in combat. If you don't give computer-controlled creatures an On-Attack trigger, they are only competent with short and long-range weapons.

This event is only available for creatures.

Cast (Timing: On)

When: This is a special event that is used only for spells. For player characters, they appear as sub-triggers for a spellcasting skill trigger. For monsters, they appear as sub-triggers for an On-Attack "brains" trigger.

Why: Spellcasting player characters must have an On-SkillUse trigger containing a Sorcery statement. This makes the skill show up as a menu choice when the player right-clicks during play. The actual spells are On-Cast sub-triggers attached to this Sorcery trigger.

Spellcasting monsters also have On-Cast triggers, but they're attached to an On-Attack trigger that serves as the monster's "brains." Within the On-Attack trigger, a SorceryQueRunes statement determines which spell is cast.

This event is only available for creatures.

CombatMove (Timing: Pre, Post)

When: Pre timing occurs before the creature moves, Post timing occurs after the creature has moved his full distance. This does not fire for each space moved.

Why: You could check the Row and Col properties of the character on this event to trigger a trap on a certain space.

This event is only available for creatures.

Death (Timing: Pre, Post)

When: Unlike most events, Death fires for the target of an attack, not the active character. The Pre timing fires when the targets HP is 0 or less, but before the death sound is played or the graphic bones appear for the character's icon. The Post event occurs after the sound and graphic.

Typical Use: On the Pre timing, you could prevent the character's death from occurring by changing the CreatureTarget.HealthNow variable. On the Post timing, you could have another character laugh, mourn, or give the player a wish that enables him to resurrect the dead character.

For example, if the target is wearing a Necklace of Immortality, you could put a Pre-Death trigger on it to reset the CreatureTarget.HealthNow to 1.

This event is also available for items.

Examine (Timing: Pre, Post)

When: This fires just before (Pre) and after (Post) a character examines an item. This applies to examining items that are carried or those in the encounter. Examine normally reports the item's description if the Use as Description box is checked. Otherwise, the player gets the "You notice nothing unusual about the [Item.Name]" message.

Why: This event is usually attached to items. On a Pre-Examine, You might make the examine fail unless something has happened (Ruben never notices the diamond has a flaw unless Josephine has told him about it). On a Post-Examine, looking at the mirror could give the player a curse. Attaching this event to a creature would be a special case. Maybe a skill that tells the player the value of gems when they're examined.

This event is also available for items.

PickLock (Timing: Pre)

When: When a character is near a locked door, or highlighting a locked chest (or other item) in a search, the he gets a chance to pick the lock. The Pre-PickLock event fires just before this option is presented.

Why: This is almost always used to alter the chance of picking the lock by adding or subtracting from the Global.PickLockChance variable. The lockpicking skill does this, but few other triggers would. Maybe a spell that temporarily increases the thieffy character's chances

This event is also available on items.

RollArmorChit (Timing: Pre, Post)

When: If dice are going to be rolled to inflict damage, this fires just before (Pre) and after (Post) that the armor chit is randomly selected. Every creature has 8 armor chit locations. If the target is not wearing armor and doesn't have tough hide, it doesn't matter which chit gets picked. All the chits would have the "Clean Shot" icon and hit do full damage. Often, however, targets wear armor or have tough hide, and the chit locations are represented by "Body" or "Helm" or "Shield" icons or whatever. Usually, these armorchit locations reduce or eliminate the damage.

Note that the CombatRollDamage statement (sometimes used for spells and magic items) has a Check Armor? checkbox. If left unchecked, armor is bypassed, the RollArmorChit event does not fire, and the target suffers full damage.

Why: Rarely used. You can override which chit is chosen by setting the Global.ArmorRoll variable to a 1 thru 8 value. On Pre timing, your choice will show graphically, but on Post timing, it does not (it shows the random chit the engine selected). Either way, the armor chit you specify will be the one damage is assessed against.

One example of this event is the Martial Arts skill trigger, which has a % chance to attempt to find an unprotected chit location on a target.

This event is also available on items.

RollAttack (Timing: Pre, Post)

When: The Pre timing fires after the victim is targeted with a "roll to hit" (usually weapons) and before the d20 is rolled. The Post timing fires after the d20 to hit is rolled and before the engine figures out whether the roll succeeded. The timing of events around the attack follows:

Pre-TargetOfAttack (target creature)
Pre-RollAttack
<i>To Hit d20 Diceroll</i>
Post-RollAttack
Post-TargetOfAttack (target creature)
<i>Engine resolves whether attack succeeded</i>

Why: Rarely used. This event is used to add or subtract from the roll to hit using the Global.Diceroll variable. The Pre timing can give a + or – to the roll. The Post timing could check the diceroll result and alter it, but the graphic appearance of the diceroll has already shown. Even though this event is available on items, you don't need it to give an item a + or – to hit because that can be set on the item's property sheet. An example for this event might be an amulet that automatically made your

next 3 attacks succeed. You would put a Post-RollAttack trigger on the amulet that checked the Global.Diceroll value against the Global.ToHit value and upped the diceroll value if necessary.

This event is also available on items.

RollDamage (Timing: Pre, Post)

When: If dice are rolled to determine damage, this fires before (Pre) the damage diceroll and after (Post). This event occurs before the damage is adjusted due to the target's vulnerabilities or resistances. It also occurs before the ApplyDamage event of the target.

Note that some spells and magic items do not roll dice to inflict damage.

Why: Rarely used. Damage rolls are usually altered in a Pre-Turn trigger (e.g. you use the statement Put CreatureNow.DamageBonus = CreatureNow.DamageBonus + POS 1 and set the Pre-Turn trigger's number of turns). But there may be something you want to happen during the damage diceroll.

This event is also available on items.

SkillUse (Timing: On)

When: This fires when the player chooses the skill. It's used to create a menu-selected skill for a character, such as Martial Arts or Moon Sorcery. Skills that are "always on," such as a weapon skill, don't use this event. Note that the Skill checkbox must be checked for the skill to appear on the menu. This event is unnecessary for NPC's or monsters.

Why: If you want to create skills that are available to players during character creation, there is a special tome called Rune in the Roster folder. Open it, and you'll see that the "characters" in that tome actually represent races and the allowable skills for those races. You can edit this to modify the normal skills or add new ones. You can also, of course, simply edit the skills of particular player-controlled characters in a given tome.

This event is only available for creatures.

Status (Timing: On)

When: This fires when the player chooses examine a character's status.

Why: Seldom used except by advanced creators who want to make new ability scores beyond Strength, Intelligence, and Agility. If you want to create a new ability, such as "Willpower" or "Charisma" or whatever, you can make an On-Status trigger attached to the character. When status is checked, the trigger's name and its ByteA variable are reported on the status screen.

This event is only available for creatures.

TargetOfApplyDamage (Timing: Pre, Post)

When: Unlike most events, TargetOfApplyDamage fires for the target of an attack, not the active character. The Pre timing fires just before the value in Global.DamageRoll is subtracted from the target's health. The Post timing occurs afterward but before the Death event (if health has been reduced to 0 or less).

The TargetOfApplyDamage event fires no matter how the damage was inflicted (dice, the CombatRollDamage statement, or a Put CreatureTarget.HealthNow – xxx into CreatureTarget.HealthNow statement).

Note that the TargetOfApplyDamage event also runs for the active items of the target creature. That way, if the target is wearing a Ring of Invulnerability, you could put a Pre-ApplyDamage trigger on it to reduce the Global.DamageRoll to 0.

Why: The Pre timing is the last chance to alter the damage that the target will suffer. The Post timing is sometimes a good place to put a "flee" trigger or dialog "Ahhh! I taste my own blood! Soon I shall drink yours!"

This event is also available for items.

TargetOfAttack (Timing: Pre, Post)

When: This event is unusual because it fires for the target creature of an attack, not the active creature. The Pre timing fires after the victim is targeted with a roll to hit (usually weapons) and before the d20 rolls. The Post timing fires after the d20 to hit is rolled and before the engine figures out whether the roll succeeded. The timing of events around the attack follows:

Pre-TargetOfAttack (on target creature)
Pre-RollAttack (on active creature)
To Hit d20 Diceroll
Post-RollAttack (on active creature)
Post-TargetOfAttack (on target creature)
Engine resolves whether attack succeeded

Why: Rarely used. The Pre-TargetOfAttack event is usually used for some sort of special protection or weakness to an attack. Note that there is a lot of vulnerable/resistant stuff that is handled automatically with checkboxes (e.g. you set one of these to True in a Pre-Turn trigger and copy that trigger to a target. See the Sub-Trigger section). But if you wanted to do something unusual, say, give a creature immunity from golem attacks, you could check to see if the active creature – the CreatureNow variable -- is a golem, and alter the Global.DiceRoll variable. The Pre timing can give a + or – to the roll. The Post timing could check the diceroll result and alter it, but the graphic appearance of the diceroll has already shown.

This event is also available for items.

TargetOfUseOnCreature (Timing: Pre, Post)

When: This event is unusual because it fires on the target creature instead of the active creature. The Pre and Post TargetOfUseOnCreature events fire just before and after the active creature uses an object on the target.

Why: Rarely used. The Pre timing is occasionally used to make the item fail. If you wanted to make a character immune to wands, you'd create a Pre-TargetOfUseOnCreature trigger that checks for wands and makes them fail. I dunno what wacky thing you'd put in the Post timing, but I'm sure somebody will think of something. Maybe the character laughs derisively if he lives through someone using an ice-based magic item on him.

This event is only available for creatures.

Turn (Timing: Pre, Post)

When: Fires just before (Pre) and after (Post) the character's turn during combat. Outside of combat, it fires every turncycle. A turncycle passes when the party moves 10 steps, or when a character uses a skill twice.

Why: The Pre timing is very commonly used. When you want a certain character to get a +2 to hit on his next turn, or -3 to defense, or whatever, this is the event to use. You can also attach pre-turn triggers to items, but it's not necessary for a bonus to hit or damage because that can be built in to the item. The following properties are often modified (+ or -) in pre-turn triggers:

StrengthBonus

AgilityBonus

IntelligenceBonus

AttackBonus (adds to your d20 Attack roll)

DamageBonus (adds to any Damage roll)

DefenseBonus (adds to your Defense number)

ArmorBonus (subtracts from incoming damage)

The statement to do this would be Put CreatureNow.StrengthBonus + Pos 1 into CreatureNow.StrengthBonus. These "bonus" variables last for the entire turncycle, not just the character's turn.

Of course, there's all sorts of things you might use the Pre timing for - freeze a character, make him go berserk, whatever. The Post timing is used less often, but it occasionally comes in handy. You could have a character as for healing if his health has dropped below a certain level, for instance.

This event is also available for items.

TurnCycle (Timing: Pre, Post)

When: In combat, this event occurs before any characters get their turns, just after the EnterEncounter event. Unlike that event, TurnCycle recurs after the all creatures get their turns. Outside of combat, this event occurs every 10 steps on the map or when a party member performs 2 actions.

Why: This is a fairly commonly used event because applies to all creatures (not just the active one), all equipped items, the active encounter, and the tome. I like to use it for traps that have a certain % chance of going off, when the party doesn't need to do anything to set it off, or other occurrences that happen every so often. This is a good event to use to check "time based" events, like a volcanic eruption or something (you keep checking to see if it's the right time, and when it is, blammo!).

This event is also available for items, encounters, and tomes.

Item Trigger Events

Combine
Death
DropItem
Equip
Examine
PickLock
RollArmorChit
RollAttack
RollDamage
SearchTraps
Take
TargetOfApplyDamage
TargetOfAttack
TargetOfUseOnItem
Turn
TurnCycle
UnEquip
Unlock
UseInEncounter
UseOnCreature
UseOnItem

Combine (Timing: Pre, On, Post)

When: Pre timing occurs before the items are combined, Post timing occurs afterwards. On timing gives you control over what happens when the items are combined.

Why: Pre timing is generally used to make the combine fail if certain conditions aren't met. Maybe you need the "Weaponsmith" skill to combine certain items, for example. If the character is not a weaponsmith, the Pre-Combine trigger executes a Let Local.Fail = Global.True statement, which prevents the event from occurring.

On timing is commonly used to create a new item when you combine two items. Suppose that you want to create the following puzzle: When a player combines a certain sword hilt and blade, he gets the cool Sword of Doom.

1. Create the 3 items: hilt, blade, and the macho sword.
2. Put an On-Combine trigger on either the hilt or the blade (in this example, put it on the hilt).
3. Paste the sword into the trigger.
4. Put the following script into the trigger:

```
CopyText "blade" into Local.TextA  
If ItemTarget.Name Like Local.TextA Then
```



```

CopyItem Sword Of Doom Into CreatureNow
DialogShow "You wield the Sword Of Doom!"
Destroy ItemTarget
Destroy ItemNow
Else
DialogShow "This hilt cannot combine with the [ItemTarget.Name]"
EndIf

```

This event is only available on items.

Death (Timing: Pre, Post)

When: Unlike most events, Death fires for the items of the target of an attack, not the active character. The Pre timing fires when the targets HP is 0 or less, but before the death sound is played or the graphic bones appear for the character's icon. The Post event occurs after the sound and graphic.

Typical Use: On the Pre timing, you could prevent the character's death from occurring by changing the CreatureTarget.HealthNow variable. On the Post timing, you could do something like have another character laugh, mourn, or give the player a wish that enables him to resurrect the dead character.

For example, if the target is wearing a Necklace of Immortality, you could put a Pre-Death trigger on it to reset the CreatureTarget.HealthNow to 1.

This event is also available for creatures.

Dropltem (Timing: Pre, Post)

When: The Pre timing fires just before the engines allows you to drop or trade an item, Post fires afterwards.

Why: Pre timing is a place to "curse" items, scripting Let Local.Fail = Global.True will prevent the player from getting rid of it (just make sure that you check the Cursed? property or he really won't be able to get rid of it). Post timing could be used for all sorts of things... maybe a wand that mysteriously fades into nothingness when dropped. If you want to prevent a creature from unequipping an item, see the UnEquip event.

This event is only available for items.

Equip (Timing: Pre, Post)

When: The Pre timing fires before the item is equipped, Post fires afterwards

Why: Pre timing can be use to prevent the item from being equipped. Maybe you don't want a non-spellcaster to even pick up a certain wand, so you put a

Let Local.Fail = Global.True.

Post timing is a great place to put descriptions and other interesting effects, as in

DialogShow "The helmet glowed a bright blue when [CreatureNow.Name] put it on."

This event is only available for items.

Examine (Timing: Pre, Post)

When: This fires just before (Pre) and after (Post) a character examines an item. This applies to examining items that are carried or those in the encounter. Examine normally reports the item's description if the Use as Description box is checked. Otherwise, the player gets the "You notice nothing unusual about the [Item.Name]" message.

Why: This event is usually attached to items. On a Pre-Examine, You might make the examine fail unless something has happened (Ruben never notices the diamond has a flaw unless Josephine has told him about it). On a Post-Examine, looking at the mirror could give the player a curse. Attaching this event to a creature would be a special case. Maybe a skill that tells the player the value of gems when they're examined.

This event is also available for characters.

PickLock (Timing: Pre)

When: When a character is near a locked door, or highlighting a locked chest (or other item) in a search, the he gets a chance to pick the lock. The Pre-PickLock event fires just before this option is presented.

Why: Easy. This is almost always used to alter the chance of picking the lock by adding or subtracting from the Global.PickLockChance variable. Thief's tools would have a trigger with this event to temporarily increase the thieffy character's chances.

This event is also available on characters.

RollArmorChit (Timing: Pre, Post)

When: If dice are going to be rolled to inflict damage, this fires just before (Pre) and after (Post) that the armor chit is randomly selected. Every creature has 8 armor chit locations. If the target is not wearing armor and doesn't have tough hide, it doesn't matter which chit gets picked. All the chits would have the "Clean Shot" icon and hit do full damage. Often, however, targets wear armor or have tough hide, and the chit locations are represented by "Body" or "Helm" or "Shield" icons or whatever. Usually, these armorchit locations reduce or eliminate the damage.

Note that the CombatRollDamage statement (sometimes used for spells and magic items) has a Check Armor? checkbox. If left unchecked, armor is bypassed, the RollArmorChit event does not fire, and the target suffers full damage.

Why: Rarely used. You can override which chit is chosen by setting the Global.ArmorRoll variable to a 1 thru 8 value. On Pre timing, your choice will show graphically, but on Post timing, it does not (it shows the random chit the engine selected). Either way, the armor chit you specify will be the one damage is assessed against.

One example of this event is the Martial Arts skill trigger, which has a % chance to attempt to find an unprotected chit location on a target.

This event is also available on creatures.

RollAttack (Timing: Pre, Post)

When: The Pre timing fires after the victim is targeted with a "roll to hit" (usually weapons) and before the d20 is rolled. The Post timing fires after the d20 to hit is rolled and before the engine figures out whether the roll succeeded. The timing of events around the attack follows:

Pre-TargetOfAttack (target creature)
Pre-RollAttack
<i>To Hit d20 Diceroll</i>
Post-RollAttack
Post-TargetOfAttack (target creature)
<i>Engine resolves whether attack succeeded</i>

Why: Rarely used. This event is used to add or subtract from the roll to hit using the Global.Diceroll variable. The Pre timing can give a + or – to the roll. The Post timing could check the diceroll result and alter it, but the graphic appearance of the diceroll has already shown. Even though this event is available on items, you don't need it to give an item a + or – to hit because that can be set on the item's property sheet. An example for this event might be an amulet that automatically made your next 3 attacks succeed. You would put a Post-RollAttack trigger on the amulet that checked the Global.Diceroll value against the Global.ToHit value and upped the diceroll value if necessary.

This event is also available on creatures.

RollDamage (Timing: Pre, Post)

When: If dice are rolled to determine damage, this fires before (Pre) the damage diceroll and after (Post). This event occurs before the damage is adjusted due to the target's vulnerabilities or resistances. It also occurs before the ApplyDamage event of the target.

Note that some spells and magic items do not roll dice to inflict damage.

Why: Rarely used. Damage rolls are usually altered in a Pre-Turn trigger (e.g. you use the statement Put CreatureNow.DamageBonus = CreatureNow.DamageBonus + POS 1 and set the Pre-Turn trigger's number of turns). But there may be something you want to happen during the damage diceroll.

This event is also available on creatures.

SearchTraps (Timing: Pre, Post)

When: The Search screen includes the Search for Traps button. This event fires just before the search occurs (Pre) and just after (Post). Traps are simply any triggers in the encounter (or on an item in the encounter) with the *Is A Trap?* property checked.

When the player searches for traps, the active character attempts to find the trap, and if that attempt fails, the trap goes off. If the trap is found, the active character automatically attempts to deactivate it, and if that fails, the trap goes off. The % for finding is (Intelligence * 4), the % for deactivating is (Agility * 4), so a character with 17 Intelligence and 13 Agility has a 68% find and a 54% deactivate.

Why: The Pre timing is often used to alter the % chance of finding or deactivating the trap. The variables are Global.FindTrapsChance and Global.RemoveTrapsChance.

This event is also available for encounters.

Take (Timing: Pre, Post)

When: This fires just before a character takes an item (Pre) and just after (Post).

Why: This is an excellent place to put traps on your treasure. Just use a Pre or Post-Take trigger and check the *Is A Trap?* property. Put a curse on the thief, make the item explode, have a panel of the wall slide aside as flames shoot out, oh, the imagination just wanders. Usually, you'll want to set the trigger's *Duration* property to 1 so that the trap fires once.

Besides traps, fun as they are, we often use this event for little puzzles like "you need x in order to get y." Suppose you have a magical sword stuck into a stone, and the character needs to hold the Pendragon Cloth to pull out the sword. To do this, put a Pre-Take trigger on the sword with script like this:

```
CopyText "Pendragon Cloth" to Local.TextA
If ItemNow.Name Like Local.TextA then
DialogShow "[CreatureNow.Name] withdraws the sword Excalibur!"
Else
Set Local.Fail = True
DialogShow "The sword is stuck fast in a stone. You cannot withdraw it."
End If
```

This event is also available for encounters.

TargetOfApplyDamage (Timing: Pre, Post)

When: Unlike most events, TargetOfApplyDamage fires for the target of an attack, not the active character. The Pre timing fires just before the value in Global.DamageRoll is subtracted from the target's health. The Post timing occurs afterward but before the Death event (if health has been reduced to 0 or less).

The TargetOfApplyDamage event fires no matter how the damage was inflicted (dice, the CombatRollDamage statement, or a Put CreatureTarget.HealthNow – xxx into CreatureTarget.HealthNow statement).

Note that the TargetOfApplyDamage event also runs for the active items of the target creature. That way, if the target is wearing a Ring of Invulnerability, you could put a Pre-ApplyDamage trigger on it to reduce the Global.DamageRoll to 0.

Why: The Pre timing is the last chance to alter the damage that the target will suffer. The Post timing is sometimes a good place to put a "flee" trigger or dialog "Ahhh! I taste my own blood! Soon I shall drink yours!"

This event is also available for creatures.

TargetOfAttack (Timing: Pre, Post)

When: This event is unusual because it fires for the target creature of an attack, not the active creature. The Pre timing fires after the victim is targeted with a roll to hit (usually weapons) and before the d20 rolls. The Post timing fires after the d20 to hit is rolled and before the engine figures out whether the roll succeeded. The timing of events around the attack follows:

Pre-TargetOfAttack (on target creature)
Pre-RollAttack (on active creature)
To Hit d20 Diceroll
Post-RollAttack (on active creature)
Post-TargetOfAttack (on target creature)
Engine resolves whether attack succeeded

Why: Rarely used. The Pre-Attacked event is usually used for some sort of special protection or weakness to an attack. Note that there is a lot of vulnerable/resistant stuff that is handled automatically with checkboxes (e.g. you set one of these to True in a Pre-Turn trigger and copy that trigger to a target. See the Sub-Trigger section). But if you wanted to do something unusual, say, give a creature immunity from golem attacks, you could check to see if the active creature – the CreatureNow variable -- is a golem, and alter the Global.DiceRoll variable. The Pre timing can give a + or – to the roll. The Post timing could check the diceroll result and alter it, but the graphic appearance of the diceroll has already shown.

This event is also available for creatures.

TargetOfUseOnItem (Timing: Pre, Post)

When: Some items can target other items with an On-UseOnItem trigger. This unusual trigger fires just before (Pre) and just after (Post) this trigger.

Why: Seldom used. This gives the target item a chance to react to being targeted. The Pre timing is occasionally used to make the effect fail. For example, maybe you've created a powerful artifact that cannot be affected by other magic items. If you put Set Local.Fail = Global.True in a Pre-TargetOfUseOnItem event, that Wand of Destruction won't bother your artifact in the least.

This event is only available for items.

Turn (Timing: Pre, Post)

When: Fires just before (Pre) and after (Post) the character's turn during combat. Outside of combat, it fires every turncycle. A turncycle passes when the party moves 10 steps, or when a character uses a skill twice.

Why: The Pre timing is very commonly used. When you want a certain character to get a +2 to hit on his next turn, or -3 to defense, or whatever, this is the event to use. You can also attach pre-turn triggers to items, but it's not necessary for a bonus to hit or damage because that can be built in to the item. The following properties are often modified (+ or -) in pre-turn triggers:

- ❖ **StrengthBonus**
- ❖ **AgilityBonus**
- ❖ **IntelligenceBonus**
- ❖ **AttackBonus** (adds to your d20 Attack roll)
- ❖ **DamageBonus** (adds to any Damage roll)
- ❖ **DefenseBonus** (adds to your Defense number)
- ❖ **ArmorBonus** (subtracts from incoming damage)

The statement to do this would be Put StrengthBonus + Pos 1 into StrengthBonus. These "bonus" variables last for the entire turncycle, not just the character's turn.

Of course, there's all sorts of things you might use the Pre timing for; freeze a character, make him go berserk, whatever. The Post timing is used less often, but it occasionally comes in handy. You could have a character as for healing if his health has dropped below a certain level, for instance.

This event is also available for creatures.

TurnCycle (Timing: Pre, Post)

When: In combat, this event occurs before any characters get their turns, just after the EnterEncounter event. Unlike that event, TurnCycle recurs after the all creatures get their turns. Outside of combat, this event occurs every 10 steps on the map or when a party member performs 2 actions.

Why: This is a fairly commonly used event because applies to all creatures (not just the active one), all equipped items, the active encounter, and the tome. I like to use it for traps that have a certain % chance of going off, when the party doesn't need to do anything to set it off, or other occurrences that happen every so often. This is a good event to use to check "time based" events, like a volcanic eruption or something (you keep checking to see if it's the right time, and when it is, blammo!).

This event is also available for creatures, encounters, and tomes.

Unequip (Timing: Pre, Post)

When: The Pre timing fires just before the engine allows you to unequip an item, Post fires afterwards. An unequipped item is still being carried, it's just not being worn or held in hand.

Why: Pre timing is a place to "curse" items, scripting `Let Local.Fail = Global.True` will prevent the player from getting unequipping the item (just make sure that you check the `Cursed?` property or he really won't be able to get rid of it). See the `Dropltem` event to further mess with items this way.

This event is only available for items.

Unlock (Timing: Pre)

When: On locked doors and chests only, the Unlock event occurs just before the unlocking attempt succeeds or fails. The Pre-Unlock event fires just before the Pre-Open event.

Why: If you wanted to put a trap on a chest, this is your ticket. Just put an chest with a Pre-Unlock trigger attached, select the *Is A Trap?* checkbox on the trigger, and do something evil. Note: To alter the chance of picking a lock, use the `PickLock` event.

This event is also available for encounters.

UseInEncounter (Timing: On)

When: The On-UseInEncounter event fires when a player character selects the Use menu option. UseInEncounter does not target an item or a creature, it is the equivalent of waving a wand (or whatever) in the air.

Note that the 3 "use" events are `UseOnItem`, `UseInEncounter`, and `UseOnCreature`. You should not assign more than 1 of these to an item.

Why: The On-UseInEncounter timing is commonly used for items, and that's where you put the statements in for the item's effect.

This event is only available for items.

UseOnCreature (Timing: On)

When: The On-UseOnCreature event fires when a player character selects the Use menu option. UseOnCreature targets a creature for its effect, for good or evil intent.

Note that the 3 "use" events are `UseOnItem`, `UseInEncounter`, and `UseOnCreature`. You should not assign more than 1 of these to an item.

Why: The On-UseOnCreature timing is commonly used for items, and that's where you put the statements in for the item's effect. Time for that magical talisman that destroys golems!

This event is only available for items.

UseOnItem (Timing: On)

When: The On-UseOnItem event fires when a player character selects the Use menu option. UseOnItem targets another item for its effect, enchanting it, destroying it, or whatever.

Note that the 3 "use" events are UseOnItem, UseInEncounter, and UseOnCreature. You should not assign more than 1 of these to an item.

Why: The On-UseOnItem timing is commonly used for items, and that's where you put the statements in for the item's effect. Time for that magical wand that enchants swords!

This event is only available for items.

Encounter Trigger Events

Combat
EnterEncounter
ExitTome
Ignore
Listen
Open
Search
SearchTraps
StepEncounter
Take
TargetOfUseInEncounter
TurnCycle
Unlock

Combat (Timing: Post)

When: This fires just after the last monster is killed in an encounter, but before the Search option appears.

Why: Seldom used, but a good place to put dialog commenting on the encounter. If you put in an encounter with a terrible opponent, like a dragon or lich, you could put in a Post-Combat trigger where a character says "Well, that was pleasant. I thought dragons only breathed fire!"

This event is only available for encounters.

EnterEncounter (Timing: Pre, Post)

When: Encounters happen when the party moves onto one of the encounter's tiles. The Pre-EnterEncounter timing fires before anything in the encounter appears, even the description. Post fires after the player has chosen one of the encounter responses (Ignore, Fight, Flee, etc.).

Why: Pre is often used to put in a special, lengthy description of the encounter, or possibly show a graphic or sound effect for the encounter.

This event is only available for encounters.

ExitTome (Timing: Pre)

When: There is at least one "exit place" in a tome where the party can leave the tome, making themselves available on the roster for adventures in other tomes. The ExitTome event fires when a party goes to the "exit" tile, just before the party is actually removed from the tome.

Why: Sometimes used to remove items from party members that are particular to that tome. Suppose that you made a goblin adventure and there was an amulet that could instantly slay the Goblin King.

You might want to remove it before the party leaves. You can also prevent the party from leaving with the `Local.Fail = True` statement.

This event is only available for encounters.

Ignore (Timing: On)

When: On the response screen for encounters, players have the option to Ignore the creatures. If you do so and the creatures are hostile, you lose initiative for 1 turncycle.

Why: You may want to do something special if the party ignores a certain beastie. Perhaps ignoring the town guard gets the party instantly thrown in jail.

This event is only available for encounters.

Listen (Timing: On)

When: The Listen command appears on the right-click menu when the party is standing next to a door. Listen doesn't do anything unless you put an encounter on that space with an On-Listen trigger.

Why: A great way to give the player clues (or red herrings) about what may await them when they walk through a door. I sometimes make the active character do a Intelligence roll to see how successful they are at listening.

This event is only available for encounters.

Open (Timing: Pre)

When: The Open command appears on the right-click menu when the party is standing next to a door (whether the door is locked or not). This event occurs just before the door opens. If the door is locked, the Pre-Unlock event fires just before the Pre-Open event.

Why: Remember, locks and keys are handled automatically (see the How to make Chests, Doors, and Traps tutorial), so this event isn't used for picking locks. If you wanted to put a trap on a door, however, this is your ticket. Just put an encounter with a Pre-Open trigger on the tile(s) in front of the door, and select the *Is A Trap?* Checkbox on the trigger.

This event is only available for encounters.

Search (Timing: Pre, Post)

When: After you click the Search button, this fires before you see the list of Items (Pre) and after you end the search (Post). Note that searches are automatically prevented if live monsters are present and their *Prevent Search?* checkbox is selected.

Why: Seldom used, but you could set `Local.Fail = True` with Pre timing to prevent the search. You could use a Post-Search trigger to set off an alarm to bring in guards. If you want to put a trap on something, however, the events *Take* or *Unlock* are better choices.

This event is only available for encounters.

SearchTraps (Timing: Pre, Post)

When: The Search screen includes the Search for Traps button. This event fires just before the search occurs (Pre) and just after (Post). Traps are simply any triggers in the encounter (or on an item in the encounter) with the *Is A Trap?* property checked.

When the player searches for traps, the active character attempts to find the trap, and if that attempt fails, the trap goes off. If the trap is found, the active character automatically attempts to deactivate it, and if that fails, the trap goes off. The % for finding is (Intelligence * 4), the % for deactivating is (Agility * 4), so a character with 17 Intelligence and 13 Agility has a 68% find and a 54% deactivate.

Why: The Pre timing is often used to alter the % chance of finding or deactivating the trap. The variables are Global.FindTrapsChance and Global.RemoveTrapsChance.

This event is also available for items.

StepEncounter (Timing: Post)

When: Encounters are often painted onto multiple tiles, and when they are, this event fires just after the party moves onto another tile within the encounter. StepEncounter never fires in combat.

Why: This is a great way to have different things happen on certain tiles within the room. Use this event and within the script, check the Tome.MapX and Tome.MapY properties to see if they're on a certain space. If so, they see a description of a portrait on the wall or something.

This event is only available for encounters.

Take (Timing: Pre, Post)

When: This fires just before a character takes an item (Pre) and just after (Post).

Why: This is an excellent place to put traps on your treasure. Just use a Pre or Post-Take trigger and check the *Is A Trap?* property. Put a curse on the thief, make the item explode, have a panel of the wall slide aside as flames shoot out, oh, the imagination just wanders. Usually, you'll want to set the trigger's *Duration* property to 1 so that the trap fires once.

Besides traps, fun as they are, we often use this event for little puzzles like "you need x in order to get y." Suppose you have a magical sword stuck into a stone, and the character needs to hold the Pendragon Cloth to pull out the sword. To do this, put a Pre-Take trigger on the sword with script:

```
CopyText "Pendragon Cloth" to Local.TextA
If ItemNow.Name Like Local.TextA then
DialogShow "[CreatureNow.Name] withdraws the sword Excalibur!"
Else
Set Local.Fail = True
DialogShow "The sword is stuck fast in a stone. You cannot withdraw it."
End If
```

This event is also available for items.

TargetOfUseInEncounter (Timing: Pre, Post)

When: Items that are used can be UseOnCreature (targets a creature), UseOnItem (targets an item), or UseInEncounter (no target is chosen, but the active encounter is the implicit target). This event fires just before (Pre) and after (Post) such an item is used in an encounter.

Why: Magical rooms or areas with strange properties sometimes use this event. A good example of this might be a magical portal area. If the teleport wand is used in that room, the party teleports to a certain location, but if the teleport wand is used elsewhere, it doesn't do anything. If you set the Pre timing to Local.Fail = True, such magic items wouldn't function in that encounter at all.

This event is only available for encounters.

TurnCycle (Timing: Pre, Post)

When: In combat, this event occurs before any characters get their turns, just after the EnterEncounter event. Unlike that event, TurnCycle recurs after the all creatures get their turns. Outside of combat, this event occurs every 10 steps on the map or when a party member performs 2 actions.

Why: This is a fairly commonly used event because applies to all creatures (not just the active one), all equipped items, the active encounter, and the tome. I like to use it for occurrences that happen every so often. This is a good event to use to check "time based" events, like a volcanic eruption or something (you keep checking to see if it's the right time, and when it is, blammo!).

This event is also available for items, creatures, and tomes.

Unlock (Timing: Pre)

When: On locked doors and chests only, the Unlock event occurs just before the unlocking attempt succeeds or fails. The Pre-Unlock event fires just before the Pre-Open event.

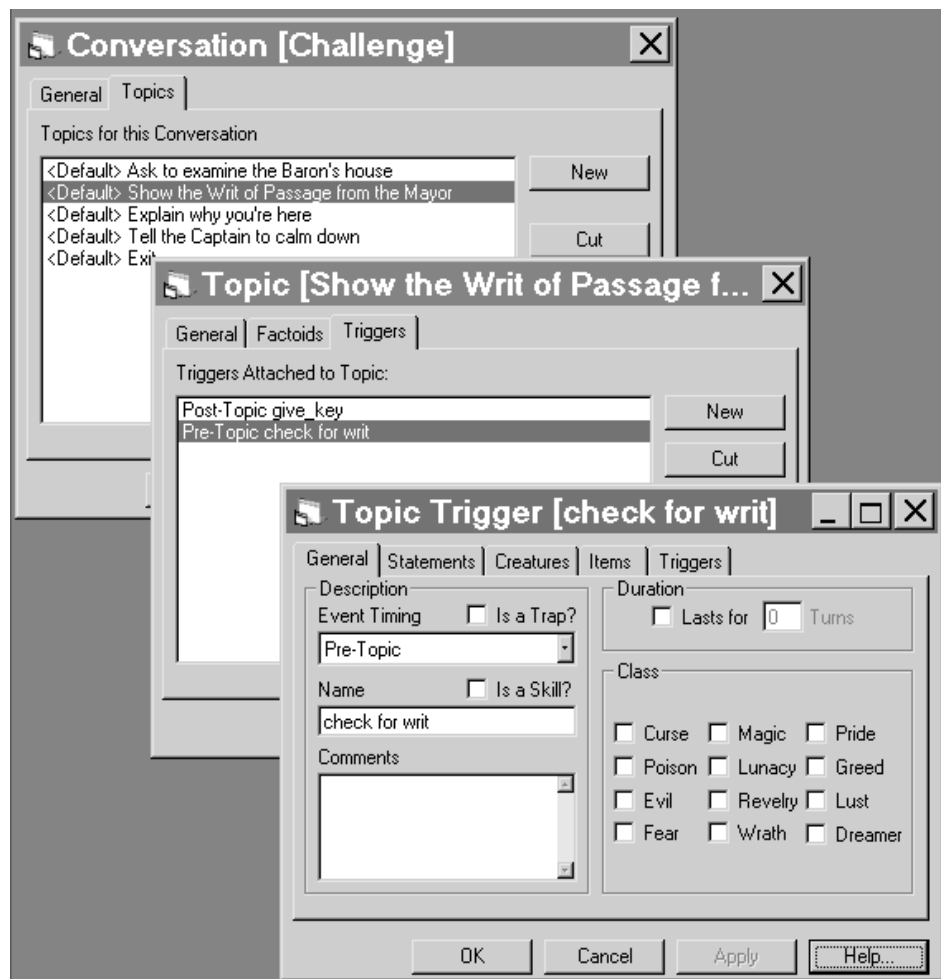
Why: Remember, locks and keys are handled automatically (see the How to make Chests, Doors, and Traps tutorial), so this event isn't used for picking locks. If you wanted to put a trap on a chest or door, however, this is your ticket. Just put an encounter with a Pre-Open trigger on the tile(s) in front of the door, and select the *Is A Trap?* Checkbox on the trigger.

This event is also available for items.

Dialog Trigger Event

There's a grand total of one dialog event, but it's a doozy.

Topic (Timing: Pre, Post)

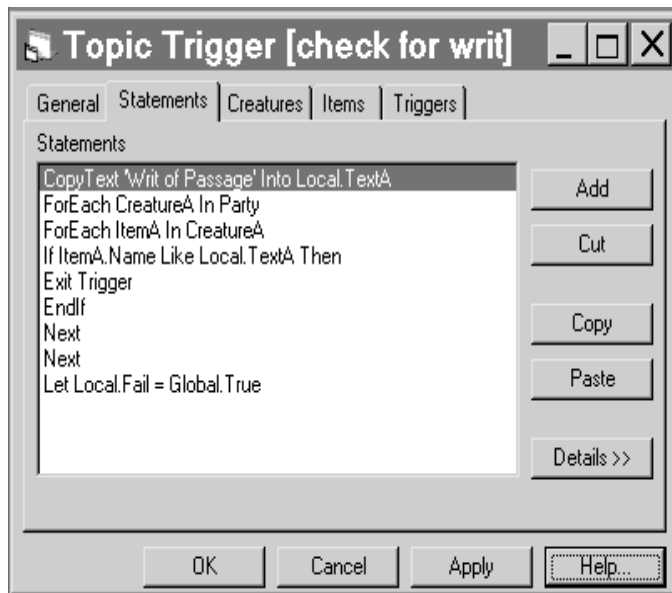


When: Computer-controlled creatures, especially friendly ones, will often engage the characters in conversations instead of trying to hack them to pieces. Conversation occurs when the player chooses the Talk button on the encounter screen, or the talk button on the right-click menu. This brings up a list of the possible characters to talk with.

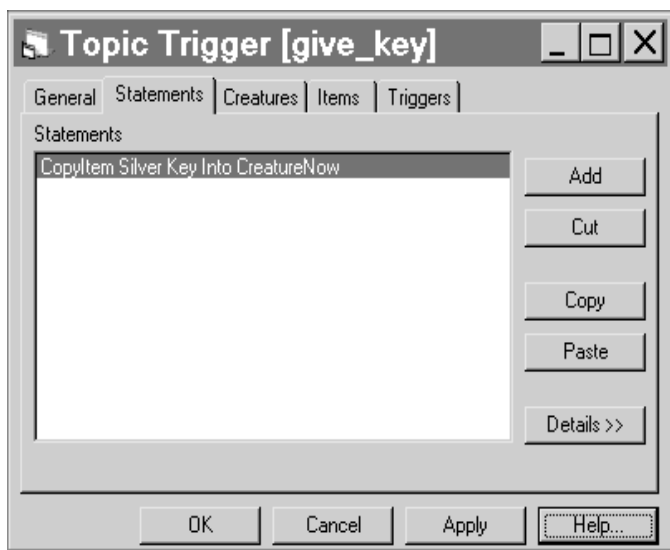
When you choose to talk to someone, the engine uses the conversation attached to that creature. The Pre timing fires before the topic is displayed, the Post timing fires after a topic is chosen.

Why: Most conversations have several "topics," which are things that the player "says" to the creature. You can attach triggers to these topics to do something. For example, the demo tome contains a Kingsmen Captain. If a player chooses the "Show the Writ of Passage" topic, I want the Captain to give him a key.

There are two triggers attached to this "Show the Writ" topic. The Pre-Topic "Check for writ" fires before any topics are even displayed for the player. This trigger looks through all the items on all the party members for the writ. If it finds the writ, the trigger does nothing. But if no one has the writ, the `Let Local.Fail = Global.True` statement prevents the "Show the writ" topic from even appearing:



So then, if the player chooses the "Show the Writ" topic, I know that he's actually got it. The silver key is an object carried by the Captain, and it's a simple matter to give it to the character in the Post-Topic trigger:



Most topics, of course, don't have triggers. When you say something to someone, they usually just talk back. But if you want something to happen based on something the character says, Pre and Post-Topic triggers are the way to go.

This trigger is only available for Dialogs.

Tome Trigger Events

EnterTome

ExitTome

TargetOfApplyDamage

TurnCycle

Note: Tome triggers are put in the Triggers folder in the Tome Window (instead of being attached to an object). Also, the tome is always considered active, so if you use the TargetOfApplyDamage of the Turncycle events, those triggers will always fire during those events. In Tosspot, for example, we used the TargetOfApplyDamage event to put in a special "Headshot" trigger that does evil things if anyone - character or monster - takes a lot of damage on their noggin. Turncycle may even be more useful here, because you could have something going on for the party no matter where they are.

EnterTome (Timing: Post)

When: This event occurs once, when the tome is started for the first time.

Why: This is a good place to put effects that occur immediately. Many tomes, for instance, may start with a cutscene that describes the background story.

This event is available only for tomes.

ExitTome (Timing: Pre)

When: Every tome has at least one location where the party can "leave" the adventure and return to the roster of available characters. This event occurs just before the engine actually removes the characters from the tome.

Why: You could use this event to remove tome-specific triggers from characters, or give them a farewell message, or whatever. You could even prevent them from leaving unless they meet certain criteria with this statement: `Let Local.Fail = Global.True`

This event is available only for tomes.

TargetOfApplyDamage (Timing: Post)

When: Unlike most events, TargetOfApplyDamage fires for the target of an attack, not the active character. The Post timing occurs after damage is applied but before the Death event (if health has been reduced to 0 or less). Because this is the Tome object, this event will fire for every creature in combat.

The TargetOfApplyDamage event fires no matter how the damage was inflicted (dice, the CombatRollDamage statement, or a Put CreatureTarget.HealthNow – xxx into CreatureTarget.HealthNow statement).

Note that the TargetOfApplyDamage event also runs for the active items of the target creature. That way, if the target is wearing a Ring of Invulnerability, you could put a Pre-ApplyDamage trigger on it to reduce the Global.DamageRoll to 0.

Why: The Post timing is sometimes a good place to put a "flee" trigger or dialog "Ahhh! I taste my own blood! Soon I shall drink yours!" We used it in Tossport to add a special bonus effect for a head blow trigger.

This event is also available for creatures and items.

TurnCycle (Timing: Post)

When: In combat, this event occurs before any characters get their turns, just after the EnterEncounter event. Unlike that event, TurnCycle recurs after the all creatures get their turns. Outside of combat, this event occurs every 10 steps on the map or when a party member performs 2 actions.

Why: This is a fairly commonly used event because applies to all creatures (not just the active one), all equipped items, the active encounter, and the tome. I like to use it for occurrences that happen every so often. This is a good event to use to check "time based" events, like a volcanic eruption or something (you keep checking to see if it's the right time, and when it is, blammo!).

This event is also available for items, creatures, and encounters.

RuneScript

Introduction

RuneScript works like a normal programming language, including such mundane things as "If - Then" conditions and looping. But you also get many special statements designed just for RuneSword. It's really an easier, special purpose language.

It probably goes without saying, but a huge feature in the Creator is that all script is created by picking from droplists. There's no hassle with syntax. You can still create scripts that don't work, or don't function exactly right, but you cannot make the game bomb with a bad script (or if you do, you encountered a bug that's our fault, not yours).

If you don't have a knack for programming, don't worry too much. You can do some amazing things with cut and paste, and tweaking the code of others.

Sections include:

- ❖ **Basic Statements** Boring statements that are universal, like IF - THEN - ELSE - ENDIF constructs. Also handy FOR-EACH and FIND statements to target stuff for you nefarious deeds as a Creator.
- ❖ **Combat Statements** Ways to make critters fight and cast spells.
- ❖ **Dialog Statements** Covers cutscenes, conversations, and other statements that get your storyline across. The most critical of these is DIALOGSHOW.
- ❖ **Miscellaneous Statements** Grab bag o junk that doesn't fit easily into any other category.
- ❖ **Weird Stuff** Commentary on unusual aspects of the language.

Note that you still need to know which event to select for your trigger, and which object to attach your trigger to. Otherwise, your wonderful RuneScript won't fire at the right time. If you want to give a +1 on the "to hit" attack roll for Sir Valiant, for example, you attach an Pre-Turn (creature) trigger to him. See the Introduction to Trigger Events for more info.

Basic Statements

The statements in this section are sort of universal, whether you're talking about RuneScript or Basic or C, you'll find conditions, IF-THEN-ELSE constructs, etc. A big discussion of programming fundamentals is (obviously) outside the scope of this document, but you'll find resources on the web for that sort of thing.

And/Or

And checks to see if all conditions are true, Or checks for any condition. Note slightly unusual syntax - the "Then" still remains on the preceding IF statement.

```
If Random 1-20 < CreatureNow.Lust Then
And Random 1-20 < CreatureNow.Greed Then
    DialogShow "Sir Valiant follows the lovely maiden upstairs and gives her all his money"
EndIf
```

CallTrigger [Sub-trigger Name]

Executes the Sub-trigger. This runs the Sub-trigger and returns to the next statement in the calling trigger. If you set Local.Fail = True in the Sub-trigger, the calling Trigger fails as well. You must create the sub-trigger (attached to the calling trigger) first.

ExecTrigger CombatSmarts

ExecTrigger [Sub-trigger Name]

Executes the Sub-trigger. This runs the Sub-trigger and returns to the next statement in the calling trigger. If you set Local.Fail = True in the Sub-trigger, the calling Trigger fails as well. You must create the sub-trigger (attached to the calling trigger) first.

ExecTrigger CombatSmarts

Exit [Trigger/Loop/Abort]

Transfers control out of Trigger (ends execution of trigger statements) or If-EndIf loop. The Abort option is unsupported.

Exit Trigger

Find [Object] in [Object] named [Text String]

This command searches for an object (item, creature, or trigger) so you can affect the object somehow. The Local.FoundIt variable returns a True or False on whether the object exists. Suppose that you want a trap to detect whether the active character is carrying the cursed Scarab Necklace.

```
Find ItemA in CreatureNow Named 'scarab necklace'  
If Local.FoundIt = Global.True then  
    'do something extra nasty  
EndIf
```

Or what if your magic sword does extra damage against goblins? On a Post-DamageRoll trigger, you could do this:

```
Find CreatureA in EncounterNow Named 'goblin'  
If Local.FoundIt = Global.True then  
    'Put Global.DamageRoll * 2 Into Global.DamageRoll  
EndIf
```

Why You Don't Use the Find Command

- ◆ If you want to affect more than one creature or item object at a time, such as a poison cloud that damages all members of a party, or spell that targets all enemy snakes, a **ForEach-Next** loop is necessary. But Find is great for affecting one object.
- ◆ Find does not work for targeting encounters elsewhere on the map. For that, use the TargetEncounter statement.

ForEach [Object] In [Collection] Next

This is the "go find something or someone" command. It enables you to do stuff like find out if one of the monsters is an elf, or if anyone is carrying a fire wand, or whatever. Note that the "EncounterNow" collection is the non-party members in an encounter (monsters, usually). The "Party" collection is party members. If you're search for, or affecting, only 1 object, use the Find command instead of a ForEach – Next loop.

'Healing all party members who are still alive:

```
ForEach CreatureA In Party  
    If CreatureA.Health > POS 0 Then  
        Let CreatureA.Health = CreatureA.MaxHealth  
    EndIf  
Next
```

```
If [Test Condition] Then  
  [Elself]  
  [Else]  
EndIf
```

The classic of flow control. The first condition that evaluates as true is the one that is executed. For a variation on how to do this, see Select Case.

```
If Random 1-20 < CreatureNow.Lust Then  
  DialogShow "Sir Valiant follows the lovely maiden upstairs"  
EndIf
```

IfText [Variable] Like [Text String]

Enables you to compare something to text (a text string).

```
IfText CreatureNow.Name Like Sir Lucky Then  
  'whatever  
EndIf
```

This command was a late addition. You will still see lots of instances where the old style comparison is used in the Tome (the old way still works). I've got a bazillion things to do, and updating old code just for appearance's sake is waaaaaay down the list. Do as I say and not as I do. This is an example of the old way:

```
CopyText "Sir Lucky" to Local.TextA  
If CreatureNow.Name Like Local.TextA then  
  'whatever  
EndIf
```

Label **Branch**

These two go together. You can put a label statement as a "bookmark" in your code. The Branch statement makes the code execution jump to whichever label you specify. For you programmers out there, this is a GOTO type statement, except it's done in an odd way. For best results, use high numbers as the labels:

```
If Local.TextA Like CreatureNow.Name Then  
  'I need to exit here
```



```
Branch POS 250
EndIf
```

```
..
```

```
Label 250
'code continues execution here
```

Put [Property] [+ - * /] [Property] Into [Property]

Evaluates and stores result into a variable. Use the Put statement to add, subtract, multiply, divide, etc.

```
Put Local.IntegerA + Local.IntegerB Into Local.IntegerC
```

```
Select [Property]
Case [X]
Case [Y]
Case [Etc.]
End Select
```

Begins a select block. Evaluates and executes statement block. When you have many possible cases, this is easier to decipher than an If-Then block.

```
Select Random 1-3
Case 1
DialogShow "Rolled a 1 so you die"
Case 2
DialogShow "Rolled a 2 so nothing happens"
Case 3
DialogShow "Rolled a 3 so you get the magic sword"
End Select
```

Set [Object] = [Object]

Assigns one object to another. You could assign a certain creature as the target creature.

```
Set CreatureTarget = CreatureA
```

TargetEncounter [Text String]

This is useful for creators who want to affect other encounters on a map based on something happening. For instance, suppose that I have several encounters where Morgarth the Vampire attacks the characters, then teleports away to attack them later. So I've made 2 encounters with Morgarth on my map, and both are active. But what if, by some stroke of fate, the players actually manage to kill Morgarth in the first encounter. On a Post-Death trigger on Morgarth, you could have this:

'Note that the 2nd encounter is named morgarth reappears

TargetEncounter 'morgarth reappears'

Let EncounterTarget.IsActive = Global.False

TargetEncounter only works on the current map, but you can get around this limitation. To affect an encounter on another map, add a factoid (morgarth dead) and check for the factoid at the first encounter in the new map.

It's worth noting that TargetEncounter is a fundamentally different type of command than TargetItem or TargetCreature. Those commands enable *players* to target stuff (with whatever). When a Creator wants to target items or creatures to muck around with, the creator uses the Find statement. TargetEncounter is also a *creator* command, except that you get muck about with encounters.

TriggerComment

Put in lots of comments on what your script is doing. You'll thank yourself later. Though you select the TriggerComment statement, it actually appears in the script as an apostrophe (').

'Test player to see whether he follows succubus in disguise

If Random 1-20 < CreatureNow.Lust Then

DialogShow "Sir Valiant follows the lovely maiden upstairs"

EndIf

While [condition]

Next

Begins looping block. Continues until the condition evaluates to false (= 0).

While CreatureNow.HealthNow <= 0

DialogShow "You're still dead"

Next

Combat Statements

The following commands are often used in monster On-Attack "brains" triggers for combat. Of course, you can simply put weapons in a creature's inventory, and they'll fight fine without a "brains" trigger (mindless goblins? oh yeah, we've got mindless goblins).

CombatAttackWithWeapon

With this single statement, a creature will

- ❖ equip a weapon if it hasn't already done so
- ❖ target an opposing character if one hasn't already been targeted
- ❖ move into proper range if necessary
- ❖ attack

"But Dan," you say, "Creatures that lack On-Attack brain triggers already do this by default! Why use this command?" How about a simple example:

```
Let Local.IntegerA = Random 1-3
Select Local.IntegerA
```

Case Pos 1

```
DialogShow Goblin Says [Brief Textline] "Die Intruder!"
CombatAttackWithWeapon
```

Case Pos 2

```
DialogShow Goblin Says [Brief Textline] "This one's ears are mine!"
CombatAttackWithWeapon
```

Case Pos 3

```
DialogShow Goblin Says [Brief Textline] "I'm cracking skull! Hee!"
CombatAttackWithWeapon
```

```
EndSelect
```

This trigger is just a simple way of getting a goblin to hurl random insults at his targets, but done well, it can add a lot to his "combat personality."

CombatAttackWithSpecial [Description] [DamageDice]

With this single statement, a creature will

- ❖ target an opposing character if one hasn't already been targeted
- ❖ attack using the description and damagedice specified

What if your critter doesn't use weapons to attack? Suppose a monster slaps characters around with tentacles, claws and bites, or shoots blue beams from his third eye? This is your command. Just describe the attack (this should be brief) and set the damagedice.

CombatAttackWithSpecial "The Golem chops at you" Damage=Dice 2d12+3

CombatMove [Toward/AwayFrom] [Motive]

Moves the Creature with a specific motive. Motive can be Closest, Farthest, Strongest, Weakest, Random or TargetCreature (so you can write a Trigger to set TargetCreature based on anything and then move toward it). Has no effect outside combat.

CombatMove Toward Strongest

CombatRollAttack [Bonus Variable] [DamageType]

Performs a ToHit Roll (rolls d20) on CreatureTarget and returns True for a hit, false for a miss. Don't use this if you're using the CombatAttackWithWeapon or CombatAttackWithSpecial commands, because those commands do a roll to hit if necessary. Generally, this is followed by a check to see if it hit, then a CombatRollDamage statement (see CombatRollDamage for an example).

CombatRollDamage [Dice Amt]

Rolls damage dice and applies the results against the CreatureTarget and stores a the damage in a variable. Don't use this if you're using the CombatAttackWithWeapon or CombatAttackWithSpecial commands, because those commands do a roll to hit if necessary. Generally preceded by a CombatRollAttack statement, unless you're in a situation where the attack should automatically succeed.

CombatRollAttack Bonus=Pos 2

If Global.AttackRoll = Global.True Then

CombatRollDamage Dice.3d6

EndIf

CombatStart

This puts the character into combat with creatures. Of course there must be creatures in the encounter for this to work. This statement is often used in dialog when things go awry. For example:

```
DialogShow DM Says [Reply WithPick] The Guard tells you to bow down before Emperor Suthnic. Will you:
DialogReply Fall to your knees
DialogReply Refuse, telling the Emperor that you parley on your feet
DialogAccept Into Local.IntegerA
,
'Now let's react to player's choice
,
Select Local.IntegerA
Case 1
DialogShow DM Says The Emperor awaits your request
Case 2
DialogShow DM Says The Emperor motions to the guards
CombatStart
End Select
```

CombatTarget [Motive]

Just sets CreatureTarget without actually moving. Motive can be Closest, Farthest, Strongest, Weakest, Random or TargetCreature.

CombatTarget Closest

Runes [Rune List]

Spells are On-Cast sub-triggers attached to a creatures On-Attack "Brains" trigger. In RuneWorld, the Runes command is the first statement in the script for spell sub-triggers. It specifies which runes are needed to cast the spell. In effect, the rest of the trigger does not execute until the proper runes are queued up. Spells with 1-2 runes are cast immediately, 3-4 rune spells take 1 turn, and 5-6 rune spells take 2 turns.

To actually cast the spells, a spellcasting monster needs an On-Attack "brains" trigger with a SorceryQueRunes statement.

Runes Earth Fire

Sorcery [FizzleRate = Property]

This command is only used in an On-SkillUse type trigger attached to a character, and is generally the only statement in such triggers. The FizzleRate Variable determines the % chance for spell failure. When the player selects the spellcasting skill, the engine lists all the character's spells. The engine identifies spells as any triggers attached to the character with the On-Cast event.

The actual spell effects, and the runes needed to cast the spell, are in the spell triggers themselves. Spells are On-Cast type sub-triggers attached to the Sorcery skill trigger. Spellcasting monsters and computer-controlled characters do not use a Sorcery statement, instead they use a SorceryQueRunes statement. Unless you're creating a new spellcasting skill for player characters, you won't be using this statement. Also see the Runes statement.

Sorcery FizzleRate = Pos. 5

SorceryQueRunes [Rune List]

This statement is necessary for spellcasting monsters, and it's usually found in an On-Attack "brains" trigger attached to the monster (such triggers determine what the creature does in combat). SorceryQueRunes queues up the specified runes. Then the engine looks at attached On-Cast spell sub-triggers for matches.

SorceryQueRunes Earth Fire

Dialog Statements

These statements are all about telling a story, and as such, are near and dear to our hearts. You can do all the sophisticated wizardry you want with clever triggers, but the truth is, the story counts. Characters need to be taunted, argued with, assaulted, loved, hated, etc. You can take the simplest, stupidest goblin monster there is, but if he says a couple of clever lines before he dies, he'll be memorable.

CutScene [Style] [PictureFile] [Text]

Show a picture and/or text in a window. Great for introductions, interludes, story scenes, etc. The Styles parameter includes several ways to arrange the picture & text. Note that the text doesn't appear in the line of code, but if you look at the code editing part of the window, you can access it.

Note: The BMP file must be in the [Your Tome Name] subdirectory within the Tomes directory.

CutScene Picture Left, Text Right Picture=warrior.bmp

DialogShow [CreatureName] Says [Text] [Style]

One of the most commonly used commands in the game, this displays the creature's picture, name, and text. The style choice has a dramatic effect on how this works:

- ❖ *Normal*: Shows speaker, his picture, and text in a window. The player must click to close the window.
- ❖ *Brief Box*: Same as normal, except the window automatically disappears in a few seconds.
- ❖ *Brief TextLine*: During combat, the text appears above the speaker's head. Outside combat, the text appears above the speaker's portrait.
- ❖ *Reply WithPick*: Shows speaker, his picture, and text in a window. The player is allowed to click on a text response. To show the possible text responses, immediately after this command, you must enter one or more DialogReply statements, and then a DialogAccept statement to store the response.
- ❖ *Reply WithText*: Shows speaker, his picture, and text in a window. The player is allowed to type in a text response. To store the response, you must follow this command with a DialogAccept statement.

If you use "DM" as the creaturename, the "Dungeon Master Orb" is the speaker. Use "PC" and the engine will use the first player-controlled character in the party as the speaker.

DialogShow DM says [Normal] "You have displeased me"

DialogReply [Text]

This gives a clickable "choice" for a player, and you usually have at least two choices (else it's not a choice, urrrr). It follows a DialogShow WithPick statement. To store the reply, you follow the last DialogReply with a DialogAccept statement. The first choice returns a 1, the second a 2, and so on. An example follows:

DialogShow Executioner Says [Reply WithPick] For high and grievous crimes against the state, how would you like to be killed?

DialogReply Hanging sounds fine

DialogReply Cut off my head with an axe, please

DialogReply How much gold do you need to forget about this?

DialogAccept Into Local.IntegerA

,

'Now let's react to player's choice

,

Select Local.IntegerA

Case 1

'wants hanging

Case 2

'wants beheading

Case 3

'offers us gold

End Select

DialogAccept [Property]

Pause until entry is received from open dialog box. See the DialogReply statement for an example.

DialogAcceptText [Property]

Pops up a window where the player can type in a text string, then saves it in a variable that you specify.

DialogDice [Dice] Into [Property]

Opens diceroll dialog box, rolls the specified dice when the player clicks, and stores the result in Variable. Suppose that you have an enchanted trap that looks like a pile of gold. You could put the following in a Post-EnterEncounter trigger:

DialogShow DM says [Normal] "There's a huge pile of shiny gold spilling out of a large chest. [CreatureNow.Name] feels an odd attraction to it, and must make a Greed roll of [CreatureNow.Greed] or more"

DialogDice 1d20 Into Local.IntegerA

If Local.IntegerA >= CreatureNow.Greed then
 'success so he doesn't rush in and grab it
Else
 'greedy bum failed and bad stuff is happening!
EndIf

DialogHide

This can be used to force "Brief Box" dialog windows to close before their time elapses. That's the only time this is useful. See the DialogShow statement.

DialogShow DM says [Brief Box] "Hey there"

DialogHide

DialogShow DM says [Brief Box] "The Hide statement makes sure that this appears right away instead of waiting for the previous Dialog Brief Box to time out"

DialogBuySell

Need info. This is the way merchants do business.

DialogSorcery [Learn, More, Forget, Leave]

You could use this statement to enable PC's to learn spells from a creature. This brings up a window that shows the On-Cast triggers from CreatureNow. You can only Learn spells that match your Sorcery Skills (a moon mage cannot learn Wrath spells, for instance).

Miscellaneous Statements

The statements in this section don't fit into any particular category. Grab bag time.

AddFactoid [Text]

Adds a Factoid, which you type in into tome's list of factoids. A factoid is RuneScript's way of globally sharing information on the rare occasions it's necessary. It beats using global variables for storage because globals can be overwritten easily, and globals are reset when the tome is saved anyway. In the Tossport demo tome, for example, there was a character called the Mayor who rewarded the party if they had rescued the Baron. But it was possible that the Baron might die or leave before the party got back to the Mayor's house. So when the party rescued the Baron, I had a statement like this:

```
AddFactoid "Rescued the Baron Successfully"
```

At a different location in the tome, the Mayor had a trigger with statements like this:

```
CopyText "Rescued the Baron Successfully" into Local.TextA
```

```
If Local.TextA like Tome.Factoid then  
  '(do something nice)  
EndIf
```

That "If" statement will check every factoid in the list for one that matches. When you're finished with a factoid, you can destroy it (see DestroyFactoid).

AddJournalEntry [Text]

Creates a Journal entry that you type in. Journal entries are commonly used as a way of adding notes and clues for quests. Maybe you have a side adventure such as a Farmer's request to rid his land of dire wolves. But because your tome is chock full of interesting side adventures, you want to help the player remember them. So when the farmer asks for help, you put in a statement like:

```
AddJournalEntry "Giles, a prosperous farmer north of Bone Spur, asked us to rid his land of dire wolves. He had seen the mauled remains of a dozen sheep, and said that no other animal could inflict such damage."
```

AddQuest

Creates a "quest" in the quest log that reminds players about something they're supposed to accomplish.

Addquest "Solve the murder of Giles Gentry"

AwardExperience [Creature]

Give experience points to the specified creature.

CopyCreature [Creature] Into [Encounter/Party]

Copies Creature that's attached to this Trigger into the Encounter or Party. If there is not encounter (you're outside combat) this creates an encounter. To go into combat, follow this with a CombatStart statement.

CopyCreature Orc into EncounterNow

CopyItem [Item] Into [Encounter/CreatureX]

Copies Items from this Trigger into the Encounter or TargetCreature. Creates Encounter if necessary.

CopyItem Sword into EncounterNow

CopyTrigger [Trigger] Into [Encounter/TargetCreature/TargetItem/Party]

Copies Trigger to the specified target. This is an extremely powerful capability that is discussed at length in the Sub-triggers section.

CopyTrigger Poisoned Into CreatureTarget

CopyTile [Tile] At X=[Property] Y=[Property] [Bottom/Middle/Top]

Copies Tile from this Trigger into the (X,Y) coordinate, overwriting the current Bottom, Middle or Top Tile. This is great for special effects to occur on the map, like a pit suddenly appearing or a wall appearing destroyed after something happens.

CopyText [Text] Into [Variable]

Copies text that you type in into a variable. This used to be necessary for IF THEN comparisons, but the new IfText statement has made it seldom used anymore – see the section Weird Stuff section on text strings.

CopyText Sir Valiant in Local.TextA

Destroy [Object]

Destroys the Object (Item, Creature, Encounter, Trigger, etc.) Poof. The object is not actually destroyed until all the statements in the destroying trigger are executed. You could have an object self-destruct.

Destroy CreatureA In EncounterNow

DestroyFactoid [Text]

Destroys factoids matching the text you type in. It's a built in wildcard, so the following statement would destroy all factoids beginning with the phrase "Arliss found," like Arliss found the artifact and Arliss found her child, so be careful.

DestroyFactoid "Arliss found"

MoveParty [Area] [Map] X=[Property] Y=[Property]

Immediately moves the party to a specified location in the Tome. You select an area, and a map, then specify the x,y coordinates.

MoveParty Area=Area1 Map=Clothmakers At X=Pos 1 Y=Pos 1

PlayWAV [FileName] [Que/Immediate]

Plays WAV format sound file from the /data/wav directory. Que plays at next opportunity. Immediate stops current sound and plays this one.

PlayWav Clap.wav

PlayMIDI [FileName] [Que/Immediate]

Plays MIDI format file from the /data/midi directory. Que plays at next opportunity. Immediate stops current sound and plays this one.

PlayMidi Imperial.mid

PlaySFX

Plays a SFX file (an animated spell effect). Does nothing outside combat. It has a Test button where you can see what the effect will look like without actually running the game.

PlaySFX bluemist.bmp Head Fling Fast

TargetItem [in Encounter/TargetCreature/CreatureNow]

For player-controlled creatures, this pops up a dialog box showing the items they carry. The player will use the item in hand on the item chosen. Whichever one is picked is loaded into the specified handle - either ItemA, ItemB, or ItemC.

TargetItem ItemA

TargetCreature [in Encounter/Party/All] [Range POS #]

Pops up a window for player-controlled characters to choose a target: other characters (Party Only), monsters (Encounter), or both (Any). You can specify a range of within POS # spaces, and you can also specify whether dead creatures are valid targets. If the statement isn't on a player-controlled character's trigger, a random creature is picked from the specified list.

For monster combat brains, you'll probably want to use the CombatTarget statement instead, because with that statement, there's a little built-in intelligence about what type of target to pick.

TargetCreature CreatureA Party Only Within Range = POS 15

Weird Stuff in RuneScript

We have a few sections on weird stuff, which is our way of discussing odd little programming issues that are particular to RuneSword.

Weird Stuff with "This.That"

If you're a non-programmer, you may be thrown off by the references to "This.That" in these docs. This is Object.Property syntax, or put another way, Thing.PartOfThing. Some examples:

- ❖ The name of the active character? CreatureNow.Name
- ❖ The value of the active item? ItemNow.Value
- ❖ The strength of the target character? CreatureTarget.Strength

And on it goes. The Properties document explains this stuff in more detail and lists the objects and properties for reference.

Weird Stuff with Variables

You cannot name your variables. You have to use one of the following:

- ❖ Local.TextA, TextB ,TextC for text strings
- ❖ Local.ByteA, ByteB, ByteC for numbers up to 255
- ❖ Local.IntegerA, IntegerB, IntegerC for numbers up to 32,000
- ❖ Global.TextA, TextB ,TextC for text strings
- ❖ Global.IntegerA, IntegerB, IntegerC for numbers up to 32,000
- ❖ TriggerNow.ByteA, ByteB, ByteC for numbers up to 255

The "local" variables can only be referenced within the trigger that sets them, and they are reset once the trigger ends. Globals are can be seen anywhere.

To save a value that you can reference later, use a TriggerNow.ByteABC variable (which is stored as a property on the trigger) . Other triggers can access the variable with a handle. For permanent storage of text strings, use a factoid instead of a variable (see the AddFactoid and DestroyFactoid statements).

Weird Stuff with Text Strings

One cool last minute change is the IfText statement. To compare something to text (a text string), you used to need two statements:

OLD WAY

```
CopyText Sir Lucky to Local.TextA
If CreatureNow.Name Like Local.TextA then
    'whatever
```

NEW WAY

```
IfText CreatureNow.Name Like Sir Lucky Then
    'whatever
```

Note that you will still see lots of instances where the old style comparison is used in the Tome (the old way still works). I've got a bazillion things to do, and updating those CopyText and If statements is waaaaaay down the list. Do as I say and not as I did.

A Minor Weirdness with Numbers

No big deal, but numbers are always preceded by a POS or NEG to indicate whether the number is positive or negative. If you're used to a syntax like

```
If CreatureNow.Intelligence > 16 Then
```

You have to remember that in RS, it's

```
If CreatureNow.Intelligence > POS 16 Then
```

Weird Stuff with Objects

In RS object lingo, the CreatureNow and ItemNow objects are the current, active objects. Suppose that 3 brave knights (Moe, Larry, and Curly) are fighting a blue banshee and a red banshee. It's Moe's turn to do something. Moe is the CreatureNow and the sword he's holding is the ItemNow. When Moe finishes his turn, Larry becomes the CreatureNow and Larry's battleaxe is the ItemNow. If Larry decides to attack the blue banshee, the blue banshee becomes the CreatureTarget object.

So what about the objects named CreatureA, CreatureB, and CreatureC, not to mention ItemA, ItemB, and ItemC? In a programmer's world, these are called *handles*. The RS engine doesn't ever use them; they're there so you can look at other objects.

For example, suppose that Moe gets a +3 to hit when there are banshees around (he hates banshees because of an unfortunate incident). We could write the following Pre-Turn trigger on Moe:

```
For Each CreatureA in EncounterNow
  IfText CreatureA.Name like Banshee then
    Put CreatureNow.AttackBonus + POS 3 into CreatureNow.AttackBonus
  ExitTrigger
EndIf
Next
```

The For-Next loop makes the RS engine look at each creature in the encounter, the two banshees in this case. One banshee at a time is loaded into the CreatureA handle. The name is checked to see if it's like "banshee," and if so, the CreatureNow (Moe the Knight because it's his turn) gets a +3 attackbonus. The ExitTrigger statement is necessary because otherwise, the loop would continue and Moe would get a +3 for each banshee.

Any A, B, and C type handles are reset when the trigger finishes.

The container for monsters and non-party NPC's is the encounter they're in. To look at each banshee in the preceding example, we used the EncounterNow object. If you want to look through the party members, substitute "Party" for "EncounterNow" in the first line..

Weird Stuff with Dice

As a tribute to pen & paper role-playing games, we use dice in our combat system. There is one 20-sided die rolled to see if you hit, and then multiple dice (usually 6 or 8-sided) rolled for damage. We also has "saving throws" against the vices or ability scores. For instance, to see whether a character can resist a taunt from a hobgoblin ("Your mother is a ^%#\$^\$#"), we might have the character roll a 20-sided die. If the result is lower than the character's Wrath number, the character just blew his cool. There's a standard way to refer to rolling dice. 1d20 is one 20-sided die, 3d6 is three 6-sided dice, 5d4 is five 4-sided dice, and so on.

Weird Stuff with Testing and Debugging

You cannot bomb RuneSword with your script, but you can certainly get unintended results. Alas, there is no "test bench" within the Creator program (version 2, don't you know). To test your changes, you've got to run RuneSword and play. A few things to remember:

- ❖ If you're using the Creator and playing your unfinished tome in RuneSword at the same time, any changes you make aren't reflected in the Tome until you Restart the tome.
- ❖ Debug mode (on the options screen) is your friend. Just remember that Debug mode does not report the results of randomly generated values accurately, at least not in this version.
- ❖ If you're tearing your hair out, wondering why the & %\$^%\$ some bit of script isn't working – trust me grasshopper, you'll find yourself in this dark place – use the DialogShow statement to help debug. Suppose that you cannot figure out why Curly the Knight isn't being helped by your Fountain of Healing. To ensure that the proper statements are executing, put in a temporary DialogShow statements that report the status of key variables. Keep going like that and you'll be able to tell how the code is working. When it looks good, delete the DialogShow statements.
- ❖ As your tome gets bigger and bigger, you'll find that it may take longer to get to the place you're working on. Don't forget that you can reset the coordinates where the party first appears in the property sheet for the tome. You can also put in teleports with the MoveParty statement.
- ❖ Go to www.runesword.com and ask questions. You will find help.

Sub-Triggers

Regular triggers are great for the majority of effects that happen instantly (healing, a blast spell, teleport, etc.). But what about making an effect occur at a certain time for a target? Or what about an effect that you want to last a certain number of turns for a target?

For instance, how do you make a party member +3 to hit on his next turn using a magic wand? The wand needs an OnUseOnCreature trigger to function. You might think that you need the following statements:

TargetCreature CreatureA Any Within Range = Pos 25
Put CreatureA.AttackBonus + POS 3 into CreatureA.AttackBonus

Both statements are fine, but the second won't seem to have any effect. You see, the RuneSword Engine recalculates all bonuses to hit and damage at the beginning of a each character's turn, just before the character becomes "active." Specifically, the following happens:

- ❖ The wand wielder uses the wand and the OnUseOnCreature trigger on his wand item fires.
- ❖ The first line of the wand's script executes and the wielder targets his friend the Swordsman.
- ❖ The script executes the 2nd line, givingf the Swordsman the attackbonus. Alas, on the Swordsman's turn (when he can actually *use* the attackbonus), the engine resets the bonus to 0 before he ever gets the benefit.

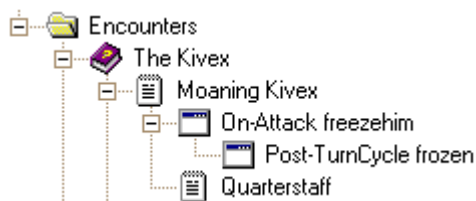
Giving Delayed Effects to a Target

The answer to this problem is making a subtrigger award the bonus, and copying the sub-trigger to the party member.

1. Make an wand item.
2. Attach an OnUseOnCreature trigger to the wand, name the trigger "Target Friend," and assign the appropriate number of charges (we will write a couple lines of runescape in the Target Friend trigger a bit later).
3. Attach another trigger (called a subtrigger) to the Target Friend trigger and name the subtrigger "Give Attack Bonus." The event for this trigger is Pre-Turn, and its runescape is:
Put CreatureNow.AttackBonus + POS 3 into CreatureNow.AttackBonus
DialogShow DM says [Brief TextLine] Power surges through [CreatureNow.Name]
4. Go back to the Target Friend trigger. Its runescape is:
TargetCreature CreatureA Any Within Range = Pos 25
CopyTrigger Give Attack Bonus to CreatureA
'Put in a special effect and a sound effect, too

Giving Long Lasting Effects to a Target

What if you want to do something to a target that lasts several turns – e.g. poisoning? This sections shows an example where we want to freeze a target for several turns.

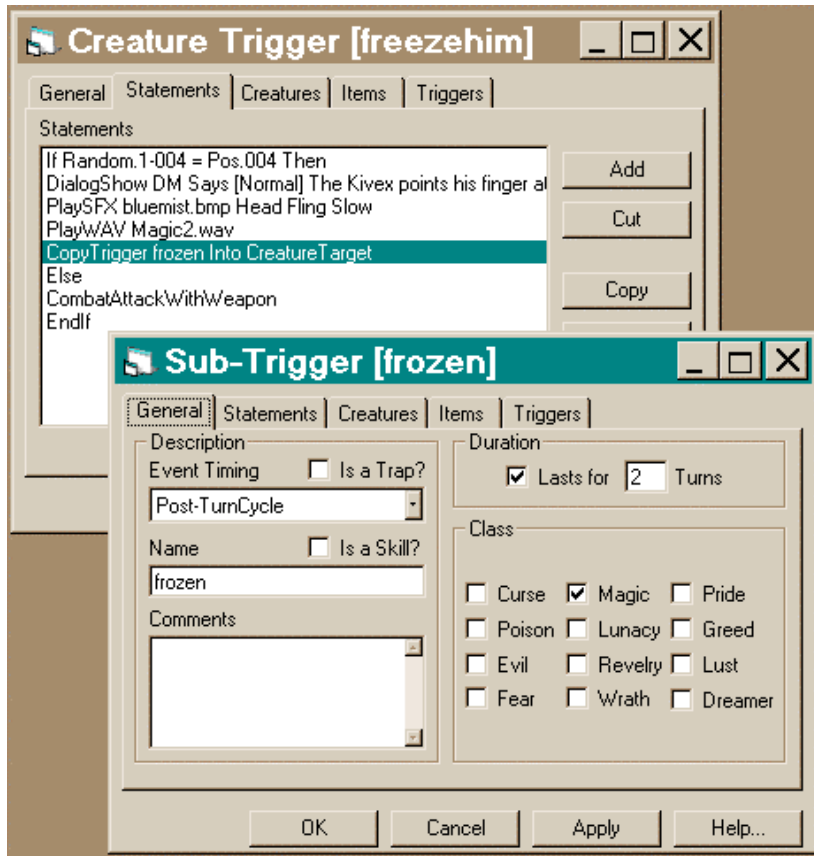


The Area Window View

Two things you can see from the Area Window screen shot:

- ❖ "Freezehim" is a normal trigger that runs when the Kivex attacks (because it has an On-Attack event).
- ❖ The "frozen" trigger is a sub-trigger because it's attached to a normal trigger. "Frozen" will never run, even after the Turncycle event, but see below for info on how it's used.

The following screenshot shows the runescript statements tab of Freezehim and the General tab of the Frozen subtrigger that will be copied.



Here's how the frozen sub-trigger can activate.

In the RuneScript code for the Freezehim trigger, there is a chance that the Kivex will employ a "freeze" attack. This is done with the line Copy Trigger frozen Into CreatureTarget. This line attaches the frozen sub-trigger to the target creature.

And guess what? Because it has become attached directly to the creature, Frozen is now a real trigger instead of a dormant sub-trigger. So on the next turncycle event, our victim will be frozen. The effect will last for two turns, then the Frozen trigger destroys itself.

Do I need a subtrigger? Questions and Answers

Magic ring that gives wearer a +2 to defense when worn? - No Trigger or Sub-trigger needed. You'd simply put a +2 in the defense property for the ring.

Magic ring that can be used on others to give them a +2 to defense for 1 turn? - Yes, the wand would have a On-UseOnCreature trigger with a Pre-Turn subtrigger. The script in the On-UseOnCreature trigger would copy the sub-trigger to the target.

+5 Sword? - Heck, that item doesn't need any triggers at all. Just set the +5 on the property sheet.

Poison Needle trap? - Yes, indeed, to copy a "Poisoned" trigger to the victim... uh, target.

Reduce target creature's strength for 3 turns? -- Yes, almost any effect that lasts multiple turns involves copying a subtrigger to a target.

Executing Sub-triggers for Convenience

We've discussed the most powerful ways to use sub-triggers: copying them to targets for delayed effects or multiple turn effects. There is a third way to use subtriggers: executing them from a main trigger. This can be quite handy.

A standard trigger can simply run a sub-trigger with a ExecTrigger statement. You'll occasionally see this within a complex On-Attack trigger for creatures. For example, an Eye Of Nuros creature has 3 different attacks that it can use. Although it's possible to put the code for each of the attacks in the On-Attack trigger, it's easier to make 3 sub-triggers and call the appropriate one from the On-Attack trigger.

See the ExecTrigger statement in the RuneScript section for an example.

Variables

There are two types of variables in RuneSword: Local and Global. Local variables are valid only within their trigger. Once that trigger ends, all locals are reset. Global variables, on the other hand, are always available to all triggers.

In *Weird Stuff with Variables*, there's a helpful explanation on how to store information for your own uses (because you aren't allowed to create your own variable names). Usually, you'll use the Local ByteABC, Local TextABC, or Local IntegerABC variables. You can use the globals, but be careful, because if other triggers can affect global values as well.

There are, of course, other variables besides those used for personal storage. The global variables True and False are the most commonly used, but many others are helpful.

The syntax is VariableType.Variable. For example, you could have statements like

```
Let Local.IntegerA = POS 10
If CreatureNow.IsFrozen = Global.True Then
...
```

Local Variables

Abort

If Abort is set to True, the Trigger stops executing and also sets Local.Fail to True (the effects of this are discussed under the Fail variable). The statement would be Let Local.Abort = Global.True. The StopExit variable and the ExitTrigger statement, on the other hand, stop the trigger from running but do not set Local.Fail to True.

ByteA, B, C

Temporary variables saved only while the Trigger is running, and can be set to anything from 0 to 255. Don't confuse Local.ByteA, B, C with TriggerNow.ByteA, B, C. The latter are actually properties of the trigger object, and their values are permanently saved.

Fail

Impacts how the triggering event will react once the Trigger has completed. Usually, the Pre timing triggers are the ones where you might set `Local.Fail = Global.True` to prevent the triggering event from occurring. Let me give you some examples to save you from more of this tortured definition. Good reasons for setting the `Local.Fail` to true:

- ❖ On a Pre-Topic trigger, this prevent the topic from appearing. Bertram the Merchant won't discuss diamonds if the PC hasn't found the diamond.
- ❖ On a Pre-Examine trigger, this prevents the description from appearing. Maybe the character must understand an ancient language to decipher the runes.
- ❖ On a Pre-Turn trigger, this prevents the character getting his turn. Perhaps he found your boggy mud a little too boggy.

FoundIt

Returns True or False based on whether the last Find command successfully found an object.

```
Find CreatureA in EncounterNow with name 'goblin'
```

```
If Local.FoundIt = Global.True Then
```

```
    'a creature with the word goblin in its name is in this encounter
```

```
EndIf
```

IntegerA, B, C

Temporary variables saved only while the Trigger is running, and can be set to anything from -32,000 to 32,000.

StopExit

Setting `StopExit` to true stops the trigger.

TextA, B, C

These are the variables you use to store text. Like other Local variables, they only last for the duration of the Trigger. For example:

```
CopyText "Golden Chalice" Into Local.TextA
```

```
Let Local.TextB = CreatureNow.Name.
```

There are no variables for permanent text storage, but you can use factoids instead (see the Tome section of the Properties document).

Global Variables

False

This is equal to 0. You can't change this value. The true and false global variables are checked with this sort of syntax:

If CreatureNow.IsFrozen = Global.False then....

True

This is equal to 1. You can't change this value. The true and false global variables are checked with this sort of syntax:

If CreatureNow.IsSpellCaster = Global.True then....

InCombat

This is equal to true (1) if the Party is currently in combat, else, it's false (0). You can't change this value.

IntegerA, B, C

Like local integers, except these can be seen by other triggers.

TextA, B, C

Like local text variables, except these can be seen by other triggers.

FindTrapsChance

This is the % chance for finding a trap in the encounter. The base chance is the character's Intelligence x 4. This is usually altered in the Pre-SearchTraps event.

RemoveTrapsChance

This is the % chance to removing a trap in an encounter. The base chance is the character's Agility x 4. This is usually altered in the Pre-SearchTraps event.

DieRollType

Contains the type of dice last rolled. Valid values are 4, 6, 8, 10, 12 and 20. Several different events roll dice. In combat, the to hit roll is one 20-sided die and the damage roll is X number of X sided dice. You can't change this value.

DieRollCount

How many dice were last rolled. Valid values are 1 to 5. You can't change this value.

ArmorRoll

The location of a successful combat hit. There are 8 possible locations for all creatures, so this returns a value of 1-8.

AttackRoll

AttackRoll contains the bonus to be applied to the next roll to hit. AttackRoll is set to 0 at the start of a Creature's turn, and then the bonuses are applied before the RollAttack event.

DamageRoll

DamageRoll contains the bonus to be applied to the next damage roll. After the RollDamage event, the DamageRoll variable contains the final roll including bonuses. DamageRoll is set to 0 at the start of a Creature's turn.

Offer

This contains the last offer made by the Creature who is buying or selling. You can use this in your Merchant's remarks during the haggling: "You must be joking! [Global.Offer] is insulting!"

RunePool [Blood, Bile, Oil, Nectar, Fire, Earth, Water, Air, Time, Moon, Sun, Space, Insect, Man, Animal, Fish, Twilight, Abyss, Eternium, Dreams]

The current number of Runes of the specified type available. For example, to check if there are enough Blood runes left:

If Global.RunePoolBlood > Pos 2 Then...

The RunePool is replenished at the start of every combat --- RuneSword copies the Map's RunePool amounts to the RunePool. Any change to these values are thus only effective for the duration of combat.

SkillLevel

This is the Skill level of the last skill used.

IsAttack [Blunt, Fire, Holy, Magic, Sharp, Evil, Cold, Mind]

This returns true (1) or false (0) if the attack is of the specified type. Useful if you want to do something special if a creature is attacked via a certain method. If you simply want to alter the damage sustained from certain attacks, however, you alter the target creature's Protection property. IsAttackTrigger [Timed, Curse, Dreamer, Greed, Lunacy, Lust, Poison, Pride, Skill, Revelry, Trap, Wrath] This returns true (1) or false (0) if the attack trigger is of the specified type.

PickLockChance

This returns an integer giving the active character's chance of opening a container (a chest or whatever). This is usually altered on a Pre-PickLock event on attached to a creature or item. For example, suppose that you want a locked chest to be 20% tougher to pick than normal. You'd put a Pre-PickLock trigger on the chest with the following statement:

Put Global.PickLockChance - POS 20 Into Global.PickLockChance

TimeDayName, TimeMoonName, TimeTurnName, TimeYearName

By default, these are "Turn: X", "Day X", "of Moon X", "in the Year X" respectively. You can change these to whatever you want by using the Post-TurnCycle Trigger attached to the Tome. For example, this lets you name each Moon.

Elegance in Dialog

Here's a few observations based on my preferences. Take them for what they're worth; you can do dialog any way you please. I'll ramble a little.

Note: For info about how to do dialog, see the interface for conversations is on Creatures Part III in the Creator Guide. Another way to do dialog is via RuneScript, using the Dialog Statements. You'll find that both techniques are used extensively in Tossport.

Which Technique is Best?

Conversations are best for conveying information and doing simple things when the player picks a certain topic. Stuff like this includes giving the player an object (such as a key, or wand, or whatever), or going into combat if the player says the wrong thing. Usually, however, conversations are best when you want the NPC to jabber on about this and that when the player picks topics.

Dialog statements are more flexible. You can enable the player to enter text, or you can show graphics via cutscenes, etc. Even in the middle of a conversation, I sometimes switch to using Dialog Statements to do what I want (see the Pub in Tossport for examples of this).

Checking Abilities and Vices

One reason for getting the right spokesman is 1d20 (20-sided dice rolls) that add a little craziness to the proceedings:

```
DialogDice DM says [CreatureNow.Name] must make a Intelligence Roll (1d20 store in Local.IntegerA)
If CreatureNow.Intelligence >= Local.IntegerA Then
    DialogShow "[CreatureNow.Name] notices that the Merchants are well armed"
End If
```

The Intelligence ability is good for general intelligence tests. Vices can make things much more interesting:

```
DialogShow DM says "The guard insults you and several generations of your ancestors"
DialogDice DM says [CreatureNow.Name] must make a Wrath Roll (store in Local.IntegerA)
If CreatureNow.Wrath >= Local.IntegerA Then
    DialogShow "[CreatureNow.Name] loses his composure and attacks"
Else
    DialogShow "[CreatureNow.Name] keeps his cool."
End If
```

You've got Lust, Lunacy, Wrath, Pride, Greed, and Revelry to work with. Use them! Remember that vice rolls succeed when the dice roll exceeds the vice value, whereas ability rolls succeed when the dice roll is less than the ability value. This is because it's best to have low vice numbers, but high abilities.

Also, it's perfectly fine to check against modified values, such as CreatureNow.Lust - 5 or whatever when the situation warrants.

And Don't Forget Skills!

A few skills depend on good creators to make them work really well. You need to check for these skills in the right situations, and give the character that has them a big advantage.

- ❖ **Fine Arts** The character has mastered music, etiquette, and diplomacy. This won't do much good in a conversation with a troll, but it should be terrific when the spokesman is dealing with nobles, or even royalty. In receiving a quest from a Duke, for instance, a character with this skill might get better terms in the deal. Or maybe the Duke would impart additional information. Or maybe he could tell that the Duke was lying about something. Also check for Fine Arts for using magical instruments.
- ❖ **Acrobatics** The character is excellent at tumbling and jumping. Maybe there's a better chance to scramble out of a trap or something. If I had a big stone ball rolling down a corridor, Indiana Jones might get a slight bonus for high agility, but he'd get a major bonus for having acrobatics.
- ❖ **Athleticism** Like acrobatics, except the character specializes in climbing and swimming.
- ❖ **Arcane Lore** Knowledge of history and magic. This character can make sense of those dusty old books.

Code Examples

For Each TriggerA on CreatureNow

IfText TriggerA.Name Like Fine Arts Then

DialogShow DM says "You remember how to howl and growl before the Morean Baron. He looks on approvingly"

AddFactoid "BaronLikesParty"

End If

Next

If you want to, you can also check the skill level of the trigger. I don't usually bother with this, because I like to reward them for even having the skill. But perhaps you have an especially difficult situation that requires the spokesman to be a Master of diplomacy:

For Each TriggerA on CreatureNow

IfText TriggerA.Name Like Fine Arts Then

And TriggerA.SkillLevel > POS 3 Then

DialogShow DM says "You remember how to bow before the Morean Baron. He looks on approvingly"

AddFactoid "BaronLikesParty"

End If

Next

Sometimes, you may want to check to see if *any* party member has a particular skill, not just the active character. Suppose that the group sees a skull carved into a stone wall, with strange writing below it. On a Post-EnterEncounter event, you may want to see if anyone has the Arcane Lore skill.

```
CopyText Arcane Lore into Local.TextA
For Each CreatureA in EncounterNow
  For Each TriggerA on CreatureA
    If TriggerA.Name Like Local.TextA Then
      DialogShow DM says "* Arcane Lore * [CreatureA.Name] says It's a Curse! Don't touch the skull!"
    Exit Trigger
  End If
Next
Next
```

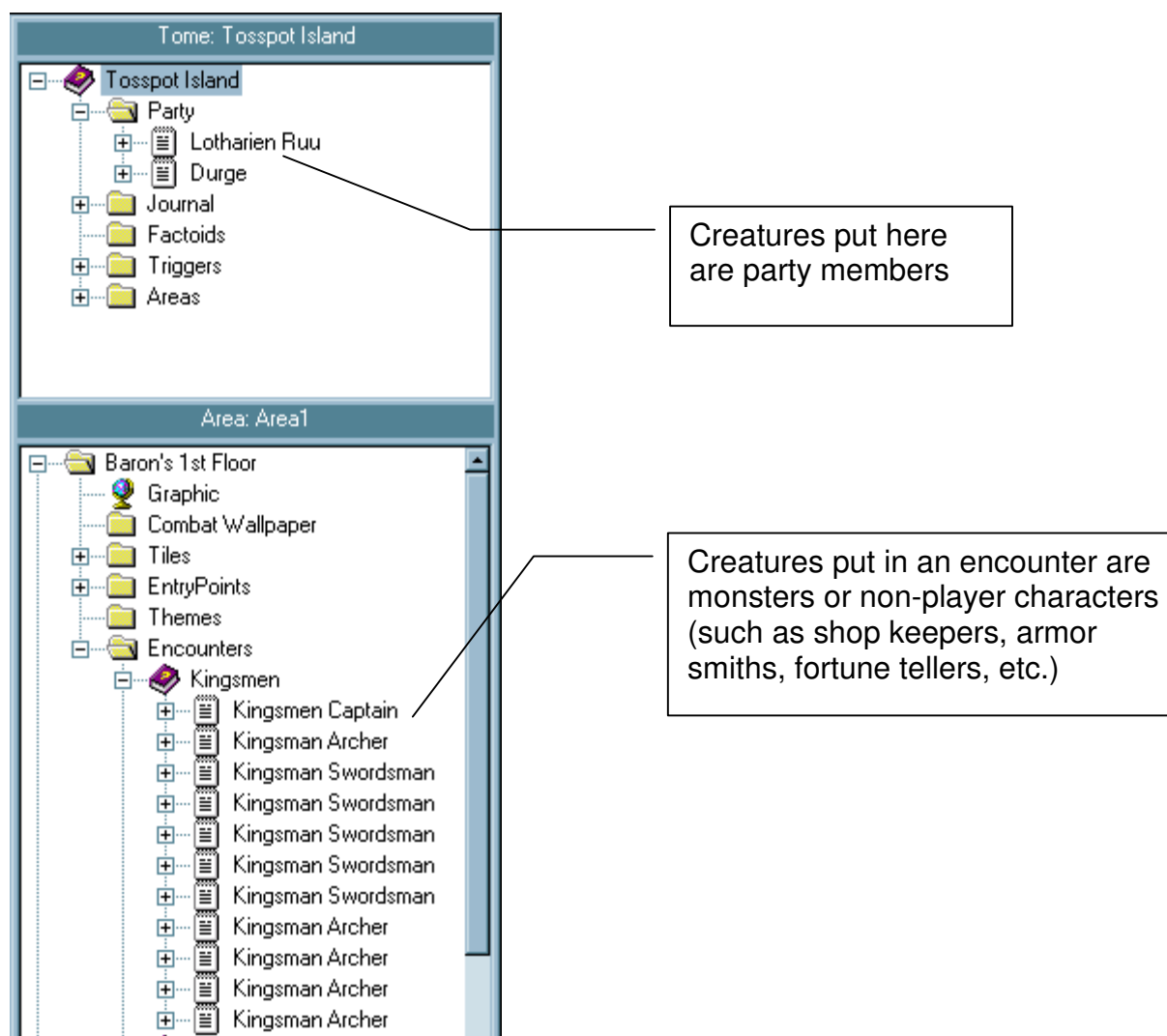
Notes: This trigger is an example of how you can easily put one For-Each loop inside another to look at more stuff (every trigger on every party member will be checked). The Exit Trigger statement ensures that you won't see this message twice if two characters happen to have Arcane Lore. Also, I put the * Arcane Lore * bit in to give players feedback, so they know that their skill came through for them.

Encounter Reactions

Cooking Up Sages, Skeletons, and Shopkeepers

I'm defining "Monsters" as living creatures that are not part of the player's group of adventuring characters. That would include shopkeepers and sages as well as skeletons. Monsters and characters are really the same in RuneSword. That is, they are Creature type objects. So what makes a monster a monster?

In a word, location. In the tome *The Land Beyond*, Durge and Lotharien are automatically part of the player's party. Note that they are located in the Tome window's Party folder. Any characters that the player gathers for this adventure are also put in the Party folder. Monsters, on the other hand, are attached to encounters. Note that the kingsmen captain, archers, and swordsmen monsters are in the Area window's Kingsmen encounter (within the Baron's 1st Floor map).



Talk, Fight, Flee, Ignore, and Search

Every encounter brings up the Encounter Reaction popup. This gives you an encounter description and gives you 1-5 of these reaction choices: Talk, Fight, Flee, Ignore, and Search. A few rules:

- ❖ If the encounter has no creatures, only the Search option appears. Search also appears if the *Prevent Search* checkbox is left unchecked on all creature property sheets. If even one creature has *Prevent Search* checked, the Search choice is unavailable. That is, it's unavailable until all such creatures are dead. In that case, the Search option appears immediately after that last creature in the encounter dies.
- ❖ Talk, Fight, and Flee appear only if the *Can Talk*, *Can Fight*, and *Can Flee* are checked on the encounter property sheet.
- ❖ The Ignore choice appears if all creatures have the *Can't Ignore* box unchecked on the creature property sheet. If even one creature has *Can't Ignore* checked, the Ignore option is unavailable.
- ❖ If there is a description, but no creature or item in an encounter, a Done button is the only reaction choice. If there's no description, creature, or item, the reaction screen will not appear and the player will be unaware that he's even in an encounter (see Traps for a reason why you might want to do this).

Computer-Control, Brains, and Friendliness

If the Computer Controlled checkbox is selected for creatures - the default - the computer runs the creature's turns during combat. Computer Controlled is always checked for monsters and usually unchecked for party members.

Computer-controlled creatures are usually brainless - another way of saying they lack an On-Attack trigger. This is fine as long as the creature fights only with weapons. If you want an enemy skeleton to fight with a sword, simply attach a sword item to him and you're done.

If you want the monster to shout insults during the fight, or use a claw attack, or immobilize foes with tentacles, they need to have brains. Most such On-Attack triggers are simple. The spellcasting On-Attack trigger is more complicated, but fortunately, we have included sample enemy spellcasters, and lots of other creatures, in the Creatures folder. There's even a video that shows how easy it is to adjust them to suit your taste.

For computer-controlled creatures, the Is Friendly checkbox determines whether the creature fights for or against the player's party.

How It's Done

Creating complex creatures from scratch can be difficult. Making a shopkeeper to sell weapons and armor from the ground up, including the dialog, takes time. Fortunately, there's no reason to do it the slow way. Use the Insert Creature from Library command or the Insert Encounter from Library command. Go to the Creatures and Encounters folders within the library and find examples that are similar to what you want. Pasting and adjusting these examples is far quicker. And once you've customized a shopkeeper the way you like, save him in the library as the mold for future shopkeeper variations. Better yet, upload him to the web site for the good of the community!

RuneSword Mantra: Cut and Paste is gooooooooood.

Locks

Just when a players think they're making headway, there's nothing like a locked door, a booby trapped chest, or a pressure plate that kicks off a hidden flamethrower to let them know who's in charge. And that would be you, the Creator (Mwa ha ha!). This section is on locks. You will find examples of these techniques in the Items directory in the library.

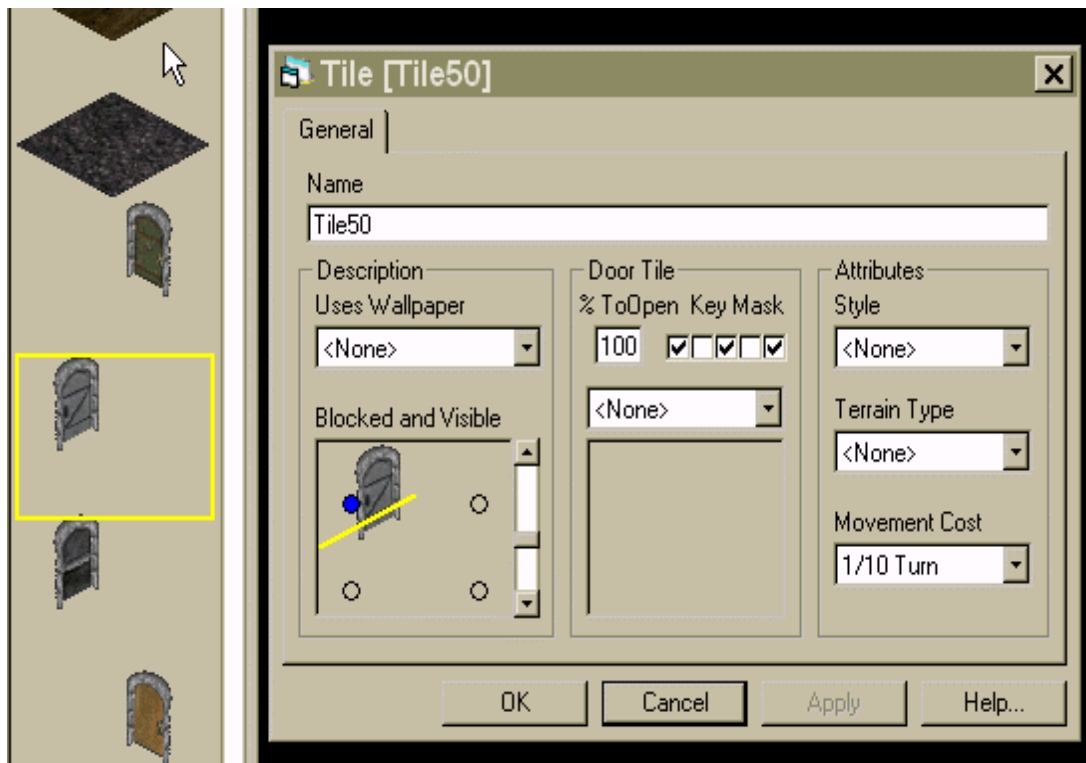
Locked Doors

Locks are generally put on special doors. RuneSword makes this especially easy. Right click to get the properties for a door tile. The yellow line on the Blocked/Visible graphic means that the door is shut. To the right of that, you can specify another graphic that is shown when the door is opened -- we don't usually use this feature in Eternia, but the capability is there.

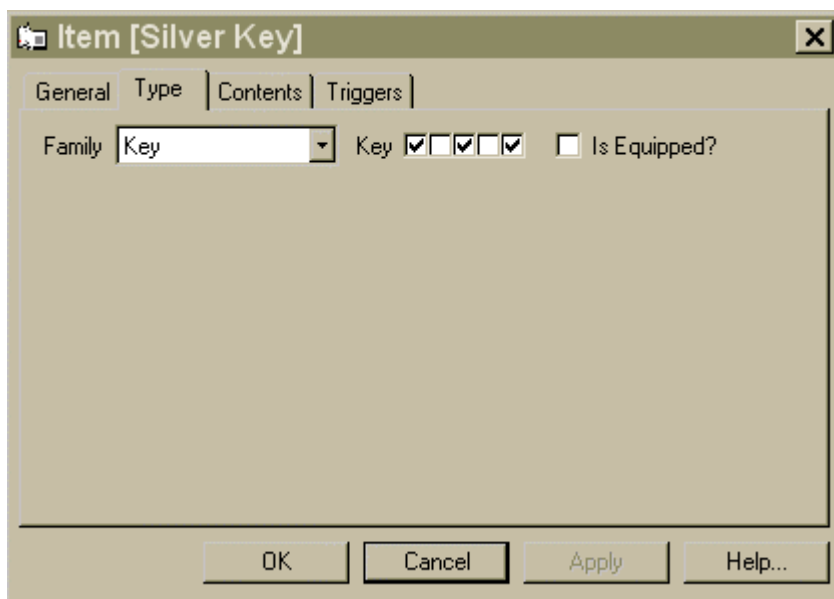
Opening Doors: Using a Key - Notice the Door Tile properties in the center. Any tile can be a "door" graphic. You can click various checkmarks on the Key Mask to make the door open to a key-type item with a matching Key Mask. If these checkmarks are set, the door cannot be bashed open (no matter what the % To Open is set to). Key-type doors must opened by keys.

Opening Doors: Bashing - If the blocked door has no checkmarks in the Key Mask, and the % To Open is less than 100, the door is considered "jammed." This is just an annoyance, because jammed doors will eventually open. Of course, if have a wandering monster encounter that may regenerate every turn or so, or a persistently-damaging trap, that annoyance could prove fatal. Or it could be a handy explanation as to why 13 goblins are waiting on the other side with arrows notched! The % To Open is the success number; the attempting character rolls a d100 with a strength bonus. For instance, suppose Sir Valiant with 17 Strength attempts to bash a 40% To Open door. His Strength adds 17 to the % chance, so he has a 57% chance of succeeding on each attempt.

Opening Doors: Other Means - If the Key Mask has checkmarks, and the player lacks the key (or the key doesn't exists, you fiend!), so another way must be found. See the Items directory in the library for examples of such items. Note that you cannot "pick" locked doors without special lockpicking tools -- the Pick Locks skill is not enough.



Properties for a typical locked door.



Properties for a key item. This key works for doors or chests that match the checkmarks in the Key Mask.

Locked Containers

Any item can be a "container." On the item's property sheet, a positive value in the Capacity property defines how much bulk the container can hold. A zero means the item is not a container.

Locks are commonly put on chests, especially the large heavy ones that contain quality loot. RuneSword makes this easy as well. Right click to get the properties for the container.

Opening Chests: Using a Key - If you want a chest to be openable with a key, make the chest's Family property Locked. As with doors, chests unlock for matching keys. If you choose the Jammed Family type, no key will work on the chest.

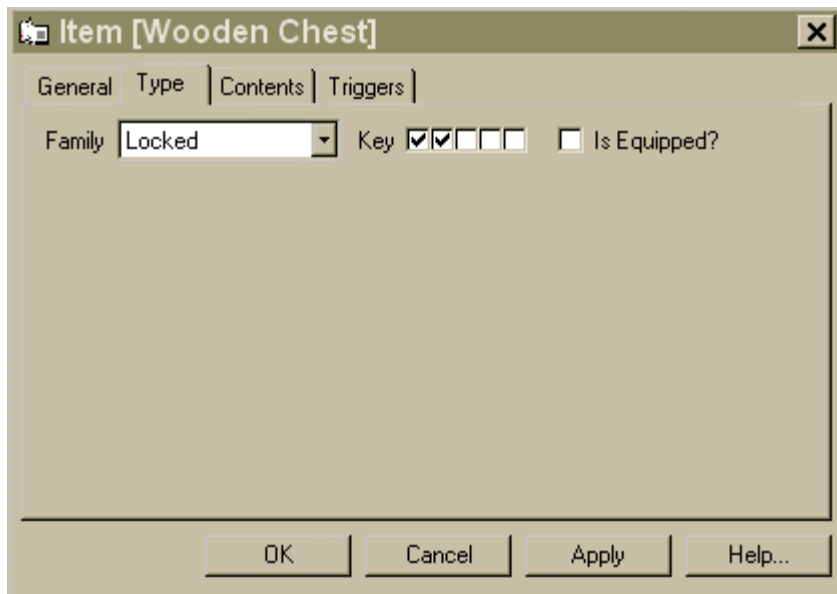
Opening Chests: Bashing - Bashing open chests will work as a last resort, but it has drawbacks. The success chance for bashing open a chest is the character's Strength x 2. Sir Valliant's 17 Strength gives a 34% success chance. Whether the bash attempt succeeds or fails, each item in the chest may get damaged:

- ❖ *Fragile* family items (i.e. potions in glass vials) have a 50% breakage chance
- ❖ *Durable* and *Money* family items (i.e. iron bars, gold coins) have no breakage chance
- ❖ All other item types (i.e. most things, including weapons and armor) have a 10% breakage chance.

As with doors, bashing will eventually succeed. But if it takes 5 attempts to open a chest containing 6 potions and a ceramic amulet, you may not be happy with the results.

Opening Chests: Picking Locks - Unlike doors, characters can pick the locks on chests without thief's tools (many chests have ingenious catches, knobs, and levers), though thieves tools may boost chances considerably. Alas, such tools are rare and generally not available for purchase. Most characters have zero chance of picking a normal locked chest. The Pick Lock skill gives characters a good chance, however. If the picking attempt fails, there is a chance that the lock will jam. A jammed lock cannot be picked or opened with a key, so you'll have to resort to bashing or other means. Jamming a lock does not harm the contents of the container.

Opening Doors: Other Means - As with doors, some magical items can open chests, even ones that are jammed shut.



A locked wooden chest.

Locking a chest and making the key difficult to find or simply unavailable is a good way to go, in our humble opinion. It makes the Pick Locks skill worthwhile and makes each chest an adventure. Succeeding at picking a lock is a significant achievement. We also like to put potions in locked chests because it makes sense to put them out of harm's way, and it's fun to penalize brawlers who bash open every chest they find.

Easy and Difficult Chests - You can alter the chance of picking a locked chest. Attach a trigger with a Pre-PickLock event to the chest. You need one statement, similar to this one:

Put `Global.PickLockChance + POS 20` Into `Global.PickLockChance`

This example would make it 20% easier to pick the lock, a worthwhile concession to make for low-level characters.

Trapped Chests - Putting a nice trap on a chest, whether it's locked or not, really makes it an adventure. See the Traps section.

Traps

Traps are fun, fun, fun to make. You can put traps everywhere, in hallways, rooms, on items, on chests, wherever. They can be triggered by all sorts of things: failing to answer a riddle, failing to pick a lock, or maybe just for walking around and breathing. There's a bazillion different ways to do traps, but we'll cover a few of the classics.

A trap is a trigger that has the *Is a Trap?* checkbox selected. Most trap triggers are either attached to an encounter or an item in an encounter. If the *Is a Trap?* checkbox is not selected, the trap will still function, but characters have no chance of either finding it. I'm as cruel as the next Creator, but fair's fair.

The player must choose the Traps button on the Search screen to have any chance of finding a trap. Even then, the Remove Traps chance is only Intelligence x 2, but this chance can be improved by the Remove Traps skill.

Item Trigger [Trigger1]

General | Statements | Creatures | Items | Triggers

Description

Event Timing ☒ Is a Trap?

Pre-Take

Name ☐ Is a Skill?

Poison Needle

Comments

Duration

Infinte

Class

☐ Curse ☐ Magic ☐ Pride

☒ Poison ☐ Lunacy ☐ Greed

☐ Evil ☐ Revelry ☐ Lust

☐ Fear ☐ Wrath ☐ Dreamer

OK Cancel Apply Help...

Do you suppose they'll notice that the sword hilt has a poison needle?

Traps on Items

In this example, the trap will go off when a character picks up the sword (Take event). Note that you don't have to make an item trap fire with the item is taken. Other logical events include Examine, Turn, and Equip. For trapped chests, the PickLock and Unlock events are appropriate.

On the Statements tab, this trigger would have something like:

Let CreatureNow.Health = NEG 1

**DialogShow [Normal] Upon grasping the sword, [CreatureNow.Name] feels the prick of a small needle.
[CreatureNow.Sex] falls over dead.**

Okay, that trap is a bit harsh. But you get the idea. You can be a bit nicer by putting a description on the sword that mention a small needle of the hilt. That way, if they examine the sword before picking it up, they just might think to look for a trap. Hopefully.

Traps in the Area

Suppose I wanted to make a nasty dungeon corridor. If characters stepped on a pressure plate, a panel in the wall slides aside and liquid flame bathes the party. I'd just put a Post-StepEncounter trigger on an otherwise empty encounter and paint the entire hallway with the encounter.

Because the encounter has no creatures, items, or a description, the player will be unaware that he is even in an encounter. The StepEncounter event fires every time the party moves from one tile to another in the encounter -- in this case, all along the hallway.

Unlike Bum, if you have empathy for players, you may be thinking "If it fires with every step they take, how will they even have a chance to search for traps?" The answer is the script:

If Random 1 - 5 > POS 1 Then

Exit Trigger

End If

DialogShow "A panel slides aside in the hall! Liquid flame belches forth, spraying the party!"

'and so on

So even though the trap fires with each step, it only has a 20% chance of truly activating. You could also make the trap activate when they reach a certain tile by checking the CreatureNow.MapX and CreatureNow.MapY properties. Or you could activate it when they take a certain number of steps within the room, by counting the number of times it fires, adding 1 to TriggerNow.ByteA until it reaches a certain number, and then activating the trap.

Under the Hood

Disclaimer This section covers advanced stuff that messes with the engine, so be forewarned. Before you go on, find the ROSTER directory (if you took the default install location, it's CROSSCUT GAMES\RUNESWORD\ROSTER). Backup the ROSTER directory to a nice, safe spot on your hard drive. You'll be cooking with gas.

Treeworld has a unique map with 7 lands, all with various types of Elves, and I've got 5 new forest type skills in mind Hey, you've come to the right place. Lunatic. A few things have come up. By that, I mean that during the beta test of RuneSword, people would say, "But can it do this?" The "I want to" stuff includes:

- ❖ New Races
- ❖ New Worlds
- ❖ New Spells
- ❖ New Skills

And on it goes. The answer is usually yep. Most creators will not want to mess with things on such a fundamental level -- making a tome is enough work -- but for others, souping up an engine is how you begin on a sports car.

You're certainly not limited to high fantasy either. Science Fiction, gothic horror, and even superhero genres are fair game.

The Secret Sauce

You'll be concerned with 3 files in the ROSTER directory:

- ❖ RUNE.INI (a critical text file)
- ❖ RUNE.TOM (a unique "Race/Skill" type of tome that you can load into Creator)
- ❖ RUNE.BMP (a graphic of the map).

You have backed up this directory, haven't you? Use your favorite text editor (Wordpad is fine) to open up the RUNE.INI file.

Here's a sample of what you'll see:

[World]

Name=Eternia

Kingdoms=12

PictureFile=rune.bmp

Tome=rune.tom

[Kingdom1]

Name=The Northlands

Location=10,2,90,200

Template=Northlander

PictureFiles=ranger1.bmp|67|27,ranger2.bmp|28|45

NameSet=4

[Kingdom2]

Name=Xal

Location=98,4,94,66

Template=Xal

PictureFiles=thief6.bmp|29|0,thief7.bmp|39|0,thief8.bmp|11|14

NameSet=7

(And on it goes for 10 more kingdoms.)

The [World] section contains:

- ❖ the name of your world
- ❖ how many kingdoms it has (each kingdom is a different race)
- ❖ the picture file for the world (players click the BMP to select where their characters are from)
- ❖ the name of the "Race/Skill tome" (more on this later)

Each [Kingdom] section has

- ❖ name
- ❖ location coordinates on the BMP picture
- ❖ template (the name of the race archetype creature in the Race/Skill tome)
- ❖ picturefiles (creature BMP files for characters from that kingdom)
- ❖ nameset (the number determines the type of names suggested for that kingdom's inhabitants)

Note that the PictureFiles line has numbers between | symbols. The numbers determine where the "close-up" box for the character's face appears. To check where it should be, bring up that creature BMP within the Creator and move the close-up box around - the coordinates appear. The limit is 18 picture files per Kingdom.

There are 12 namesets numbered 1 thru 12. Experiment with them to see the appropriate name suggestions.

Another Ingredient (the Strange Race/Skill Tome)

There is one special tome that really isn't a true tome: *RUNE.TOM*. It's located in the ROSTER directory instead of the usual TOME directory. Fire up creator and open that odd tome. Notice that it has no maps or anything like that. In fact, all it has is 12 characters in the Party folder. But these aren't regular characters - they're "archetypes" for the kingdoms in the RUNE.INI file. These archetypes don't have a graphic, and their names match the Template lines in the [Kingdoms] sections of RUNE.INI.

The skills attached to the archetypes are the skills that are offered to characters from that kingdom in the Create Character process. Characters don't get the skills automatically, of course, they still need to buy them.

Changes you make to these archetypes define how such characters come out. Change the stats, and everyone from there will get the same stats. Give them a sword, and everyone from there will start with a sword. Make them immune to fire-type damage, and everyone from there will be immune. Give them a curse, and (you get the idea).

There is a special trigger event that's often attached to these archetype creatures called Post-CreateCharacter. It fires after the character is created, and it's a good way to equip a starting character with appropriate equipment. For instance, you could give the character an ordinary sword if he has the sword skill.

The Map

And don't forget that you can make your own BMP graphic map file in you favorite paint program.

A Bit of Guidance

The name of the RUNE.INI file must remain the same. Within that file, you can define the name of your Race/Skill tome and your map, which you must also place within the ROSTER directory. This is powerful stuff, and you can see how easily you could capsize game balance in favor of super characters - "let's see, in my version of the Xal Kingdom, everyone starts with 50 strength, 75 Agility, 44 Intelligence, a vorpal sword, and has 400 skill points to spend." But then, you're above all that visceral satisfaction, right? Hey, where are you going?

And Just for Fun

Some people like random rolls for the Strength, Agility, and Intelligence attributes. Moreover, some have wondered why we cannot accommodate more attributes, because they have something in mind that SAI doesn't cover.

Ask and ye shall receive. Sometimes. When the planets align and I'm in the right mood. Anyway, I wrote a trigger called New Abilities that you'll find in the LIBRARY\TRIGGERS folder. It's a tome-level trigger that randomly rolls up Strength, Agility, and Intelligence for party members when you start a tome. It also adds 3 new abilities I made up, called Leadership, Charisma, and Something I Forget, and randomly rolls them (to check these new abilities, each character also gets a faux "skill.") These additional abilities scores are saved permanently in a trigger on characters, so you could check them or change them or whatever.

If you're interested in rolling up stats, add the New Abilities trigger and season to taste.

Sequels

As long as we're on the subject of advanced topics and feedback, a few of you ambitious types have asked about sequels. Remember the D&D days of a module series, where Module 1 was for character levels 1-3, and Module 2 would continue the storyline for character levels 2-5? Suppose that there's an evil Spider Queen in Module 1, and you gave players several ways of dealing with her: banishing her to another plane, killing her, imprisoning her in a magic lantern, transforming her into human form, or maybe Spidey got away and was still wreaking havoc somewhere at the finish. How do you know how Module 1 ended when you're creating Module 2?

Easy. The characters go on, as well as anything attached to them. You could copy a trigger named Spider Queen Result to all party members. Store a number on that trigger's ByteA, B, or C variables that means something to you - i.e. a 1 means Spidey turned into a woman (the ByteABC variables on a trigger are actually permanent *properties*). At the beginning of tome 2, you can check for the trigger value and react accordingly.

Appendices

Appendix A: Armor

- Don't wear body armor unless you have a high enough Battle Armor skill
- Spellcasters should not wear shields or helms
- Martial Artists should not wear any type of armor
- Ambidex warriors should not wear shields

Armor Name	Absorbs	Type/Coverage	Defense Adjust	Notes
Leather	25%	Body/50%	-1	Need novice level in Battle Armor skill to avoid -4 Action Point penalty
Chainmail	50%	Body/50%	-2	Need adept level in Battle Armor skill to avoid -8 Action Point penalty
Platemail	75%	Body/50%	-3	Need expert level in Battle Armor skill to avoid -12 Action Point penalty
Firefoe Plate	80%	Body/50%	-1	Need master level in Battle Armor skill to avoid -18 Action Point penalty. 0 damage from fire-based attacks.
Swordturner Plate	80%	Body/50%	-2	Need master level in Battle Armor skill to avoid -18 Action Point penalty. Half damage from edged weapon-based attacks.
Helm	50%	Helm/12.5%	-1	Reduces chances of being stunned or knocked unconscious
Helm of Drann	60%	Helm/12.5%	+1	Reduces chances of being stunned or knocked unconscious. Half damage from mind-based attacks.
Buckler Shield	75%	Shield/25%	0	Light encumbrance
Large Shield	100%	Shield/25%	0	Heavy encumbrance
Paladin Shield	100%	Shield/25%	+2	Heavy encumbrance. Half damage from evil-based attacks
Cloak	0%	Backpack/0%	0	Reduces cold-based damage by 30%

Absorbs How much damage is absorbed IF the armor is hit (see coverage)

Type/Coverage Type classifies the armor – you can wear only 1 of each type. Coverage shows the % of attacks that hit the armor.

Defense Adjust The defense number, or what others need to hit you, is 13 for normal sized characters. Most armor slows you down and makes you easier to hit, so it has a negative adjustment. Some magical armor may give a bonus, however.

Notes Special info about the armor.

Appendix B: Weapons

Weapon Name	Range	Damage Dice	Damage Bonus	Attack Bonus	AP	Notes
Dagger	Short	1d6+1	0	0	6	
Short Sword	Short	3d4	0	0	8	
Long Sword	Short	3d6	0	0	10	
Club	Short	2d4	1	0	6	Blunt type damage
2-Handed Battle Axe	Short	3d6	2	1	11	Cannot use shield or Ambidex
2-Handed Sword	Short	3d6	4	0	11	Cannot use shield or Ambidex
Battle Axe	Short	3d6	1	0	11	
Wooden Staff	Med	3d4	0	0	9	
Crossbow	Long	*notes	0	0	7	*uses short arrows that do 2d4 damage. Require a quiver to hold arrows. Cannot use shield or Ambidex
Long Bow	Long	*notes	0	1	10	*uses arrows that do 3d4 damage. Requires a quiver to hold arrows. Cannot use shield or Ambidex
Spear	Long	2d8	5	0	12	One use
20 Darts	Long	1d6	0	0	7	
Harklon Long Sword	Med	3d6	5	3	8	
Vorpal Battle Axe	Short	4d6	4	4	9	Instant kill on a neck hit
Bane Dagger	Short	1d6+1	3	3	6	Successful hit poisons foe

Range Short is 1 space, Medium is 1-2 spaces, Long is 3+ spaces

Damage Dice How many dice are rolled for damage and what kind, i.e. 3d6 is 3 six-sided dice

Action Points How many action points are used with 1 attack. If the overswing option is off, the character cannot attack with less than this many action points. If overswing is on (the default), the last attack is penalized by the shortage. Consider a warrior with 14 action points. With a dagger (6 AP), he gets 3 attacks, and the last one has a -4 to hit penalty. With a long sword (10 AP), he gets 2 attacks, and the last one has a -6 to hit penalty.

Appendix C: Spells

Level 1

School	Spell	Time	Effect
Greedspell	Numbfinger	Instant	Single target suffers 9 damage per caster's skill level. May (+15% per caster skill level) force target to drop item in hand.
Greedspell	Snap Trap	1 Turn	Gives target +15% chance per caster's skill level of finding traps. Lasts 3 turns.
Greedspell	Covetous Eye	1 Turn	Turns random item in room into gold approximating item's value.
Lunaspell	Bravado Delusion	Instant	Single target believes he is invulnerable and thoroughly wonderful, sometimes wastes time by striking poses. Short-range attacks against target get +5 to hit. Lasts 5 turns.
Lunaspell	Thought Loss	1 Turn	All enemies lose 1d4 Strength, Agility, and Intelligence (and also Action Points) for a number of turns equal to the caster's skill level.
Lustspell	False Eros	Instant	All sentient male opponents in room must save vs. Lust or stare in slack-jawed rapture at an illusion. Lasts for a number of turns equal to the caster's skill level.
Lustspell	Betrayal	Instant	Single target believes his friends have turned on him. Lasts 2 turns.
Pridespell	Holistic Healing	Instant	Removes poison, curses, and fear from single target.
Pridespell	Radiant Glory	Instant	Removes health from 1 undead target, killing it, and adds the health to the spellcaster.
Pridespell	Righteous Anger	1 Turn	90% chance that single target's weapon damage increases 1d10+6, but 30% chance that target gets -2 to hit. Lasts for 3 times as many turns as the caster's experience levels.
Revelspell	Essence Touch	1 Turn	Single target within 1 space gains 2d6+3 times the Caster's skill level in health.
Revelspell	Phase Self	Instant	Caster's current health will be the same 4 turns in the future, unless the caster dies.
Wrathspell	Firefingers	Instant	Single target within 3 spaces suffers 1d10+4 damage. 10% chance that the caster is affected instead.
Wrathspell	Hotmetal	1 Turn	Single target wearing chain or plate armor suffers 4d8 damage each turn until armor is successfully shed.

Level 2

School	Spell	Time	Effect
GreedsPELL	Daran's Trick Box	1 Turn	Creates an illusionary box in the hands of the target, who must resist a strong urge to "open" it. If opened, the box explodes for 2d10+5 damage with area effect. There is a 5% chance that a few genuine gold pieces will be left behind after the explosion.
GreedsPELL	Aether Palace	1 Turn	Target lies in a dreamlike state. Lasts 3 turns.
LunaSPELL	Shadowguard	1 Turn	Caster and party members within 2 spaces get +2 Defense. Monsters who come within 2 spaces of caster must make an Intelligence check or run in fear.
LunaSPELL	Finger Of Death	2 Turns	Kills one target within medium range (1-2 spaces). This spell does not work against animals or the undead.
LustSPELL	Nightmare	Instant	Target quails with fear for 2 turns. Only works against intelligent opponents.
LustSPELL	Infatuation	Instant	The caster gains health as the target loses health. The amount is up to 3d4 per skill level of the target. This spell works better if the target is intelligent and a member of the opposite sex.
PrideSPELL	Giant Strength	Instant	Doubles target's strength for 5 turns.
PrideSPELL	Heroic Resolve	1 Turn	Target gains that "in the zone" feeling and cannot miss his next 2 attacks.
Pride SPELL	Hardiness	1 Turn	Give +2 Defense to target for 2d4+1 turns. Bonus +1 Defense per skill level of caster.
RevelSPELL	Sybyl's Dance	Instant	Target suffers 1d6+3 to Agility for at least 2 turns. The duration increases for the Revelspell skill's level.
RevelSPELL	Wind of Hope	Instant	All party members gain 4 Health and 2 Action Points per the caster's experience level.
WrathSPELL	Jumpfire	1 Turn	Chaotic wildfire strikes a foe and bounces around unpredictably.
Wrathspell	Unthinking Fury	1 Turn	Single target gets +15 to damage and +15 to Action Points for 1d6+1 turns. 20% chance each turn that the target acts irrationally. Very irrationally.



Why cast spells if you can wield a club such as this?

Level 3

School	Spell	Time	Effect
GreedsPELL	Dreamwealth	1 Turn	Creates illusion that may render sentient beings motionless until they make an Intelligence roll.
GreedsPELL	Midas Murder	1 Turn	Damages target. The amount depends on luck and how much gold the caster is carrying.
Lunaspell	Ice Storm	1 Turn	Dagger-like icicles rain down. Damage increases with the caster's experience level.
Lunaspell	Stricken Soul	Instant	Target either misses turn or gets a -5 on attack rolls during his turn. Target also may drop item in hand. Lasts 3 turns.
Lustspell	Betayal	Instant	Target may decide that his allies are really his enemies. Lasts 2 turns.
Lustspell	Smolder	Instant	Target is slowly covered with flames for 6 turns. By the last turn, the flames do 5d4 damage
Lustspell	Globe of Desire	1 Turn	All enemies must make Intelligence rolls every turn, for the duration of combat, or suffer damage from a magical sphere. Damage is slight, based upon the caster's experience level.
Pridespell	Lunar Eclipse	Instant	Party is protected from effects of evil, fire, and sharp-based attacks for many turns.
Pridespell	Rally	1 Turn	Increases your party's Action Points by 2d6 for 2 turns.
Revelspell	Magestorm	1 Turn	Chaotic spell that strikes random foes, draining Action Points and inflicting damage.
Revelspell	Transmute to Ale	1 Turn	Creates a magical beer in your hand, but no remote control. The effect of the beer is rather random.
Wrathspell	Desolation	Instant	Drains 3d4 strength and agility for 1d6 turns.
Wrathspell	FireBlast	1 Turn	Target suffers 4d4+5 fire damage. There is a 3% chance that the spell will backfire on the caster.

Level 4

School	Spell	Time	Effect
GreedsPELL	Five Finger Discount	1 Turn	Silently steals 1 item from target creature and hides it among the caster's equipment. This spell only works once against a creature in its lifetime.
Lunaspell	Frenzied Mortis	1 Turn	Temporarily brings a dead target creature back to life. It is quite agitated and usually fights by the caster's side for a few turns before lapsing into sweet repose again.
Lustspell	Summon Djinni	1 Turn	Summons an elemental from the fabled City of Painful Remembrance, to fight by your side for a few turns.
Pridespell	Continuation	Instant	Adds 2d6 turns to most spells that are affecting the target creature. This can also cause unpredictable side effects in spells.
Revelspell	Disease	1 Turn	Target creature becomes progressively more vulnerable to more types of attacks. Eventually the creature becomes defenseless against all sorts of attacks.
Revelspell	Divine Bow	1 Turn	Gives an enchanted spirit bow (+10 to hit) to the target. The bow disappears after 3 turns.
WrathSpell	Scorching Suns	2 Turns	Obliterates target in a column of fire.

Appendix D: Skills Available to Races

Skill	North	Xal	Isle	Gaul	Tal	Andor	Jafra	Skandar	Zur	Kypar	Morea	Argos
Acrobatics	X	X		X	X			X			X	
Ambidex	X	X		X	X		X	X			X	
Appraise			X	X		X			X	X		
Arcane Lore			X			X	X		X	X		X
Athleticism	X	X			X		X	X		X		X
Axe	X	X	X				X		X		X	
Backstab		X				X		X			X	
Bardsong	X		X					X		X		
Battle Armor		X	X	X	X		X	X			X	
Bloodshield				X	X		X	X	X		X	
Body Building	X		X	X	X		X	X	X		X	X
Bow	X	X		X			X	X			X	
Charge			X	X			X		X			
Cleave	X		X		X	X					X	
Club	X		X									
Critical Strike	X	X	X		X				X			X
Dagger		X		X		X	X	X	X	X	X	X
Deflect Arrows	X				X	X				X		
Distract	X					X	X		X			X
Evasion	X	X		X	X			X		X	X	
Fencing		X			X		X	X				
Fine Arts			X			X	X	X		X		X
Greedspell				X								
Healing Hands	X		X		X	X						X
Health	X		X	X	X	X	X	X			X	X
Heroism	X				X			X			X	
Leadership				X	X		X		X			
Lunaspell			X							X	X	X
Lustspell				X						X		X
Mad Rush					X						X	
Martial Arts		X			X						X	X
Perfect Hit	X	X			X			X				
Pick Locks			X	X		X		X				
Pridespell			X							X		
Remove Traps		X		X					X			
Revelspell						X				X		X
Revive			X	X	X	X	X		X		X	X
Staff		X				X	X	X		X		X
Sword		X		X	X	X	X	X			X	
Taunt				X		X		X	X			
Whirlwind	X				X		X				X	
Wizardsry			X		X	X		X	X	X		
Wrathspell			X							X	X	

Appendix E: Designer Notes

Notes and opinions from Dan and Adam. Mostly opinions.

- II Despite the 20-sided die roll to hit, the damage dice, and the turn-based combat, RuneSword II departs from the D&D model in significant ways (no matter which edition you're using). RS II is an original system. We love pen and paper role-playing games. RuneSword II has its look and attitude as a tribute to those RPG's – but the computer is a different medium.
- II We like strategic, fun combat systems. We've never had patience for the kill-the-hordes powergaming RPG's, or emphasizing the save-reload cycle. RPG combat tactics offer all sorts of intriguing possibilities and unique occurrences.
- II In book publishing, a tsunami of me-too, Tolkien-inspired novels hit the bookshelves years ago. Most were awful, a few were good, but the sheer size of the wave burned many out on fantasy fiction. The successors emulated everything about Tolkien except his imagination. If you use elves, they'd better be *different*. The world of the included tomes is Eternia, not a generic fantasyland. Eternia has an impact on the rules, from ability scores to skills. Not all tomes made by creators will be set in Eternia – or even classified as fantasy. Get out your plasma rifle.
- II If you're into story construction, go to a library and read chapters 2-4 of Orson Scott Card's *How to Write SF & Fantasy*. Some of Card's novels are quite good and this book has 75 pages of clarity and depth.
- II Your real world setting is important during play. Darkness, candles, incense, or maybe just a good set of headphones. Get some MP3's from the Net for fighting and travelling in Eternia. Classical always works, but use whatever puts you in the adventuring trance. Put the MP3's in these directories:
 - ❖ Fighting Music **Program Files\Crosscut Games\RuneSword\Data\Music\Combat**
 - ❖ Travelling Music **Program Files\Crosscut Games\RuneSword\Data\Music\Adventure**
- II Most of the rules can be altered or bypassed with a little creativity. If you have any questions about particulars, drop us a line at www.runesword.com

*All copyrights, trademarks, whatever, on D&D (Dungeons and Dragons) are property of Wizards of the Coast. Or Hasbro. Whoever. Translated to the neutral evil alignment tongue of lawyers, this would be....