Contents

Utopian Worlds Adventure Creation Guide

Components:

Adventure Design:

Distributing your adventures:

Formal Guidelines:

Using Utopian Worlds Development Suite

The Component Editors:

Scripting:

Compendiums:

Menu Commands:

Component Security:

Miscellaneous:

Information

Distribution of Program Files:

Utopian Worlds Developer Support:

Acknowledgements:

Utopian Worlds is currently available for all Windows platforms supporting Win32 including Windows 3.1 with Win32s, Windows NT and Windows 95.

Credits

Lots of great talent went into the design and development of this game and its concepts. Thanks to all who helped.

Head Programmer

James Payne

Game Graphics Designer

Wendy A. Peabody

Trademarks

List of Distributable Files

You can package several of the included adventure components in your own adventures, and distribute them in one copyrighted product. The following is a list of the files that you may include in your adventures:

- Any icon or bitmap included on the distribution disks, except those icons and bitmaps appearing in the sample adventures
- Any adventure component included on the distribution disks, except those adventure components appearing in any of the sample adventures

In ***no*** way is it permissible to distribute any dynamic-link library, executable file or help file on any online system or through any electronic media without express permission from the author. Any violation of this agreement consitutes a copyright and license violation.

Copyright Notice

When distributing your adventures, you must either include your copyright notice or add the Utopian Worlds Development Suite copyright notice.

This is Utopian Worlds Development Suite v1.0.

E-Mail Tech Support

Technical Support via electronic mail is available for all users of Utopian Worlds Development Suite. Simply address your brief bug report or technical question to:

```
uworlds@aol.com
```

E-Mail Adventure Support

You can address questions or comments about Utopian Worlds Development Suite or adventure programming in general to the same address that is used for electronic mail technical support (`uworlds@aol.com`). Please indicate that your question addresses Utopian Worlds Development Suite in general, not technically.

Bulletin Board and Real Time Discussion

Envisioned for Utopian Worlds Development Suite support includes a fully operational bulletin board system for exchange of adventure files as well as other general files. This bulletin board system may also feature real-time discussion.

However, in the meantime, real time discussions may become available on major online services. These forums will provide an outlet for discussion on adventure programming for Utopian Worlds Development Suite and also RPG gaming in general. Registered Utopian Worlds Development Suite developers will be notified of future real-time conferences.

The Utopian Worlds Registration System

The Utopian Worlds Registration System allows Utopian Worlds Development Suite developers to make contact with the game author and also allows Utopian Worlds Development Suite developers to keep in touch with each other. It is through the Registration System that developers will be alerted of real-time discussions. The files **BASREG.WRI** and **ADVREG.WRI** contain information about basic and advanced registration, respectively.

Map Editor

Map Editor is a robust map creation system. It stores a fixed array of tiles (see below) that contain all the necessary information about a map. This design is incredibly easy to use, and sports a very speedy drawing method.

Map Editor also supports a three dimensional system that is very handy for creating multi-level dungeons and towering skyscrapers. There is no limit to the number of floors that you can create, memory permitting. Remember that as you add floors and increase dimensions, memory and disk requirements rise exponentially.

To learn about the design and the use of Map Editor, select one of the following topics:

Using Map Editor

- Tiles
- Selecting Tiles with the Tile Selection Dialog Box
- Placing Tiles
- Connecting Maps with Linkage
- Specifying Information with the Header Dialog Box
- Editing Tiles
- Using Levels
- Applications of Scripts (including Traps)
- Auto Links

Tiles

Tiles are fixed-size, square blocks that retain certain characteristics about themselves. When creating a map using tiles, you piece these tiles together like a puzzle, creating a mosaic of different tiles. Tiles unleash their properties when they are stepped on, but things like items and monsters can be accessed from a short distance away. Tiles rely on an external library to process events.

You can use the description attribute of each tile to play multimedia sounds and to show dazzling bitmaps. To find out how, see below.

Below is a list of all the attributes of a tile, and a description of each attribute.

Display File

The display file attribute controls the appearance of the tile to the player. This should be a valid Windows icon, bitmap or Utopian Worlds animation.

Overlay File

This attribute defines a picture to be drawn on top of the display file. It can be any of the same three types.

Overlay Two

This attribute is identical to the Display File, but it is drawn after the Overlay File.

Manual Multi-State Tile

This attribute indicates that the tile is a **multi-state** tile. See Maps for more information.

Branch to Map

This attribute defines the map that the character will be transported to when he steps on this tile. It should be a valid map file.

Item Here

This attribute defines the item that lies on the tile. The user can retrieve this item. It should be a valid item file.

Person Here

This attribute defines the person that lies on the tile. The user can speak to the person. It should be a valid person file.

Monster Here

This attribute defines the monster that lies on the tile. The user can fight the monster. It should be a valid monster file.

Unpenetrable

This attribute allows you to restrict access to this tile. When the character attempts to maneuver into the tile, he will be denied access. This attribute, however, can be changed by multi-state tiles and devices. If you want to absolutely restrict access to the tile, use the Never Access flag instead.

Never Access

This attribute allows you to restrict access to the tile unconditionally. It differs from Unpenetrable because it cannot be changed by any means.

Drop Item Here

This attribute allows you to conditionally restrict access to this tile. When the character uses the specified item on the tile, the tile will unlock and he will be allowed access. Using the wrong item will not unlock the tile.

Trigger Event ID
This attribute will trigger the event(s) with the specified integer ID.

Linkage Value
This attribute attaches an ID to the tile. Originally it was only used for branching purposes, hence the term linkage. Now, it has expanded to be used by events and also by certain devices.

Jump To Linkage
This attribute causes the character to be teleported to the tile having the specified linkage value.

To Level
This attributes causes the character to be moved to the same position (x,y) in the given level. To move to a different position and a different level, use the Jump to Linkage flag.

Adjust Level
This performs the same function as the To Level flag, but performs a relative change in level. For example, to move up one level, specify 1. To move down a level, specify -1.

Device
This attribute allows you to select a device to inhabit the tile.

Device Info
Certain devices require information. This is where to put that information. See devices for information about the information required by devices.

Script
This field allows you to indicate a script, in the Universal Script Language to run on certain conditions, given below.

Run On Use
This attribute indicates that the script given will be run when the tile is used.

Run On Step
This attribute indicates the the script given will be run when the tile is stepped on.

Description
This is a descriptive sentence that is displayed when the user requests a description of the tile. If the first character of the description is an open bracket ([), Utopian Worlds will interpret the description as a file in any one of the supported formats.

Script Applications

Why should you create a script? There are **plenty** of reasons to do this. First of all, script-izing many of your complex tiles (ones that you find yourself using over and over) makes it easy to recreate them countless times. Second, you can port them across adventures and to new versions very simply. You can also compile large resources of scripts and upload them for other developers to purchase or use.

One important application of a script is the trap. Traps are specialized actions that take place when the user steps on a tile. With a script, you can play a highly detailed movie, a sound effect, take damage from the character, and tell him what just went on. It would only take the following four lines of scripting code to perform the above application:

```
play "trap.avi"
play "trap.wav"
modhp -10
message "You have just been struck by a magical arrow!" "Trap"
```

That's all there is to it! You now have a magical arrow trap that does ten points of damage! With the incredibly flexible scripting command language, you can perform almost any function immediately when the user sets foot on a tile.

Scripts can also be used in conjunction with the **multi-state tile** feature (see Maps). With a script, you can create a creaking door, an old noisy, rusty gate -- all elements that enhance the realism of your story.

Scripts can be used in events and in other scripts. You will begin to realize the incredible power of scripting once you begin to dabble.

Tile Selection Dialog

This dialog box allows you to edit the attributes of tiles. For information about the attributes of tiles and their functions, see tile attributes.

Placing Tiles

Now that you have selected a tile with the tile selection dialog box, it is time to put that tile pattern to work. The map editor maintains a template of the tile you created, identical in every way. Everytime you *place* a tile, the tile template is placed on the map in the position you chose. That means you do not need to select tiles every time you want to place one. The tile you most recently selected stays in effect until another tile is chosen.

To place a tile, just click the left-mouse-button anywhere inside the map boundaries. The tile will appear where you clicked. You can use the painting method to speed tile placing. To do this, press and hold the left-mouse-button in the map boundaries, and begin to move the mouse around. Tiles will begin to follow the mouse cursor. To stop, just release the mouse button.

Editing Tiles

Sometimes it is necessary to edit a tile that has already been placed. To do this, just click the right mouse button over the tile you want to edit, and select **Edit this Tile...** Then, the tile selection dialog box will appear, filled with the tiles attributes.

You can also bypass the selection dialog to choose a tile to place. Click the right mouse button over the tile you want to model the current tile (the tile to place) after, and choose **Select this Tile as Current**. You can then place that tile as you would any other tile.

Connecting Maps

You can connect maps using linkage values (for information on linkage values, see tiles). It is very easy. Tiles that branch to other maps use their own linkage values to find a tile to send the character to. For example, tile A has linkage value 7, and branches to map EX1.MAP. Utopian Worlds will search for tiles in EX1.MAP with linkage value 7. The first tile that is found with that linkage value will accept the character.

Levels

Levels are the 3-D element of your map. In the set header dialog box, you can specify the range of levels you want to include. You can define a totally new mosaic of tiles for each level, and move sequentially or randomly through them. It is a very simple concept, but provides speedy access to alternate floors. It also lets you link floors coherently in one map file.

Move to a Specific Level

To move to a specific level and begin editing, just select **Set Viewing Level...** and type the correct level number in the edit box. Then click **OK**. You will notice the display change, and the title bar alter to reflect your new level.

Move Up or Down a Level

To move up or down a level, just click on the corresponding arrow (up or down) on the toolbar, or choose the respective menu command (**Jump Up One Level** and **Jump Down One Level**).

Header Dialog Box

The Header Dialog Box is designed to get information about the size of your map and the size of the tiles that will occupy that map. Below is a description of each control and its function:

Name

This contains the name of the map or adventure, or any appropriate stamp. This is used for internal markings only.

Author

This contains the author of the map, adventure, or any other important information. This is currently used for internal markings only.

Map Width (tiles)

Specifies how many tiles will fit in one row across the map. Ten is a good size for small maps while 25 is good for large-sized rooms and fields.

Map Height (tiles)

Specifies how many tiles will fit in one row vertically up and down the map.

Tile Width (pels)

Specifies the width of each tile, in pixels. Unless you are using bitmaps with small dimensions, this should be set to 32 to accomodate the standard icons.

Tile Height (pels)

Specifies the height of each tile, in pixels. This again should be 32.

Multimedia

This specifies the multimedia file to play when the map opens. This can be set at a later time using the menu command.

Map Editor v2.0

Map Editor v2.0 introduces a new map creation paradigm, the object-oriented paradigm. Instead of using two-dimensional, fixed size tiles like Map Editor v1.0, Map Editor v2.0 introduces variable sizing, three-dimensional object layering and a new object-oriented interface. In most cases, you will find that using this version of the map editor is much faster and more enjoyable than using the old map editor.

Choose one of the following topics to learn about the dsign or the use of Map Editor v2.0:

Using Map Editor v2.0

- Objects
- Creating Objects with the Object Creation Dialog Box
- Placing Objects with Dynamic Variable-Sizing
- Placing Objects with Static Variable-Sizing
- Editing Objects

The Design of Map Editor v2.0

- Internal Structure

Objects

Objects are the foundation of the Utopian Worlds Development Suite Map Editor v2.0. Each object, like a tile, has certain attributes. In fact, each object is based on a tile, so it has many of the same attributes, with a few additions and deletions. The big advantage of objects is the flexibility obtained with dynamic sizing. Instead of having a fixed size tile, Map Editor v2.0 allows you to create objects with any size and shape (not necessarily square). It also vanquishes the limitation of a fixed map size. Now, maps can be a large as the map edit window allows.

Objects also introduce a new complexity into character maneuvering. The character will still move inside a grid of 32 x 32, just like in Map Editor v1.0. For checking tile penetration, Utopian Worlds will first check the upper left corner, then the center, and finally the lower right corner of the character icon, in that order, if no match is found.

Below is a list of the attributes an object has. You will notice that most of these attributes are identical:

Display File

The display file attribute controls the appearance of the object to the player. This should be a valid Windows icon, bitmap or Utopian Worlds animation. You can choose **Tile** or **Stretch**. **Tile** works for icons and bitmaps, but not animations. It reproduces the display file as many times as needed to fill the entire object rectangle, arranging the pictures in a standard tile layout. **Stretch**, on the other hand, simply ensures that the bitmap (the only supported type for stretching) will exactly fit the object rectangle.

At and Size

These attributes define a rectangle that the object will fit in. By default, these values are zero. If you plan to use dynamic sizing, keeping these values zero will suffice. If you want to use static sizing, you will need to indicate a rectangle for the object to fit it. These values should be in pixels. Using static sizing is generally good to ensure map alignment. The **At** values are not used when creating an object, but they can be used to move an existing object to a new position, or to ensure correct positioning of an existing object.

Branch to Map

This attribute defines the map that the character will be transported to when he steps on this object. It should be a valid map file.

Item Here

This attribute defines the item that lies on the object. The user can retrieve this item. It should be a valid item file.

Person Here

This attribute defines the person that lies on the object. The user can speak to the person. It should be a valid person file.

Monster Here

This attribute defines the monster that lies on the object. The user can fight the monster. It should be a valid monster file.

Unpenetrable

This attribute allows you to unconditionally restrict access to this object. When the character attempts to maneuver into the object, he will be denied access.

Drop Item Here

This attribute allows you to conditionally restrict access to this object. When the character uses the specified item on the object, the object will unlock and he will be allowed access. Using the wrong item will not unlock the object.

Trigger Event ID

This attribute will trigger the event(s) with the specified integer ID.

Linkage Value

This attribute attaches an ID to the object. Originally it was only used for branching purposes, hence the term linkage. Now, it has expanded to be used by events and also by certain devices.

Device

This attribute allows you to select a device to inhabit the object.

Device Info

Certain devices require information. This is where to put that information. See devices for information about the information required by devices.

Description

This is a descriptive sentence that is displayed when the user requests a description of the object.

Object Creation Dialog

It is very straightforward to create objects with the Object Creation Dialog Box. Simply fill in the appropriate values according to what attributes the object will have.

Placing Objects

Using Dynamic Variable Sizing

Dynamic Variable Sizing is a new way to place objects that allows maximum flexibility in visual placement. To place an object, you first must create one with the object creation dialog box. Then, click-and-hold the left mouse button at one corner of the objects rectangle, and drag to the other corner. A ghost rectangle will show you where the object will be. Then, release the mouse button, and the object will be placed.

Placing Objects

Using Static Variable Sizing

You can place objects with the *static variable sizing* method also. This allows you precise control over object size, and allows you to align objects on the 32 x 32 character movement grid. To use this method, fill in the appropriate object width and height in the **Size** field of the object creation dialog box. Then, click the left mouse button anywhere in the map window to place the object. A ghost rectangle will follow the movement of the mouse cursor to indicate the position of the object.

Editing Objects

Sometimes an objects attributes must be altered. To do this, just click the right mouse button over the object and choose **Edit this Object**. The object creation dialog will appear, filled with the attributes of the object.

Internal Structure

The internal structure of Map Editor v2.0 is more complex than the simple array approach taken by Map Editor v1.0. Instead of an array of tile structures, Map Editor v2.0 uses a linked list of actual windows. Each window keeps one tile structure to maintain its attributes, and a rectangle to maintain its object rectangle.

Map Editor v2.0 has shifted most of the responsibility of maintenance and editing to the object itself, rather than Map Editor v1.0s approach of external modification and drawing routines. Each tile window can edit itself, move itself, and even delete itself when asked to do so. This offers a much more object-oriented approach and is more flexible than Map Editor v1.0.

Person Editor

The Person Editor is used to edit people. It is very easy to use, but it is very powerful. To learn more about using person editor and the scripting language, choose one of the following topics:

Using Person Editor

Keywords
Response Text
Person Attributes
Manipulating Keywords

Using PersonScripting

Person Scripting Language Fundamentals
Person Scripting Command Compendium

People

People are very important and thrilling parts of a Utopian Worlds Development Suite adventure. It is important to make interesting people who have a purpose - to give the player information about the game or entertain him. Either will do.

See the Utopian Worlds Development Suite Adventure Guide for some ideas about making people.

Keywords

Keywords are words that a person responds to. Each keyword defined for a person will have corresponding response text that indicates the text the person will say when the keyword is spoken to him.

Response Text

Reponse text is the text a person says in response to a certain keyword.

Person Attributes

Each person has certain attributes that describe him. Below is a table of all the definable attributes:

Attribute	Description
Name	This is the name of the person, and is displayed in the Conversation dialog when the user speaks to this person.
Display File	This is the display file (icon or bitmap) that is drawn on a tile when the person is selected to appear on that tile.
Multimedia	This is the multimedia file played when the Conversation is started.
Cannot reach with GlobeNet	This means that the person cannot be accessed using GlobeNet (i.e. communications devices)
Cannot find with GlobeNetQuery	The equivalent of an unlisted number

Manipulating Keywords

It is very easy to add and remove keywords and response text.

Adding Keywords

To add a keyword, simply type the keyword into the keyword entry box, and click Add. The new keyword will appear in the list to the left, automatically alphabetized.

Removing Keywords

To remove a keyword, simply select the keyword you wish to remove in the list, and click Remove.

Assigning and Editing Response Text

Response text is always paired with a keyword. To edit the text that is paired with a certain keyword (for example, hello), simply select the keyword from the list, and begin typing (or editing) text in the large, multiline edit box below. To save the text you have entered, just click anywhere else in the dialog box.

Monster Editor

The Monster Editor is used to create foes that will impede the character on his way through your world. See Monsters for more information. To learn about using the monster editor, choose one of the following topics:

Using Monster Editor

Monsters
Giving Monsters Items
Using Monster Files

Monsters

Monsters are designed to oppose the character by giving him a good fight. Make sure that all monsters are strategically placed, and that you have not overcome a weakling adventurer by putting a terribly difficult monster in his path.

Every monster has certain *attributes*, and a table of them follows:

Attribute	Description
Hit Points	Hit points are a measure of the monsters life force. These points can be drained by damage, and when they reach zero, the monster is dead.
Magic Points	This is a measure of the spellcasting capability of the monster. Each time the monster uses a spell, his magic point total decreases, and when it reaches zero, he cannot cast spells any longer.
Maximum Hit Points and Magic Points	The upper limits for these two values.
Weapon	The monsters weapon skill. The weapon skill quantifies the monsters aptitude at weapon handling. The higher this value, the more likely he will score a hit.
Defend	This number indicates the monsters success in defending incoming blows.
Damage	This number indicates the monsters aptitude at successfully doing damage.
Strength	This number affects the monsters damage when using weapons.
Agility	This number affects the monsters chance to dodge attacks.
Intelligence	This number affects the monsters spell success.
Resistance	This number affects the monsters resistance to damage in general.
Experience	This is the amount of experience given to the character when he vanquishes the monster.
Money	This is the amount of money given to the character when he vanquishes the monster.
Name and Display File	These specify the name and the file to display during battle (bitmap only), respectively.
Level	This number indicates how hard of an opponent the monster is.

Giving Monsters Items

Monsters carry items with them, and choose the best of them to use. To give monsters items, just enter the items filename into the Item File box, and click Add. To delete an item the monster already has, select it in the list and click Delete.

Using Monster Files

It is very simple and straightforward to use monster files once you have them saved. To add a monster to a map tile, just enter the filename into the **Monster Here** box, and click the checkbox. You can also use File Manager to drag UWM files onto specific tiles.

Notice that monsters do not appear on tiles. This is to surprise the casual adventurer. But, if he is smart enough to look at clues (which you must provide) he can know where the monsters are, and avoid them or fight them.

Item Editor

The Item Editor is used to create objects that the character will take and use on his quest. See Items for more information. To learn about using the item editor, choose one of the following topics:

Using Item Editor

Items
Using Item Files

Items

Items are objects that perform functions. The function can be causing damage to a foe, healing yourself, restoring your magical power, etc. Items can also have multiple functions, too. They can raise your attack ability at the same time they heal you to full power. They can slow your opponent at the same time that they inflict critical damage.

Items are *incredibly* powerful, and they can be abused by a novice adventure designer. This is why it is imperative that items comply with the Adventure Development Guidelines version 1.1. Be aware of the fact that items **can** be carried between adventures, and this is what makes them dangerous. If very powerful items are given to a character, he can carry these extremely powerful items to other adventures, and make mince-meat out of them. It is no fun for developers or players.

When creating items, you can opt to follow the **strict** guidelines or the **lax** guidelines. The strict guidelines give you actual numbers that your items must fall in the range of. The lax guidelines give hazy upper limits, and remind you that items should not be toyed with.

Items have a specific naming convention that allows you to block parts of the name from the users view. You can do this by placing double front and back slashes in the item name, as necessary. Text after a double backslash will never be visible. Text after a double front slash will be visible only if the **Mask Item Name** flag is not set. For example, naming an item Short Sword\\My adventure will appear as Short Sword. Naming an item Short Sword//of Death and setting the **Mask Item Name** flag, the user will see the item as Short Sword. When (and if) the flag is removed, the user will see Short Sword of Death.

All items have certain attributes, and what follows is a table of these attributes:

Attribute	***Description***
User Hit Point/Magic Point Change	These describe the change in the users hit points and magic points, respectively, when the item is used in combat or outside combat.
Target Hit Point/Magic Point Change	These describe the change in the targets hit points and magic points, respectively, when the item is used only in combat.
Duration	This describes the length of time that the items effect will last. Note that this includes *all* effects, including damage and healing.
Weight	This describes the physical weight of the item. Characters can only carry 25 pounds times their strength.
Category	This classifies the item, and determines the rules of combat it follows. For example, spells behave differently than weapons in combat.
Range	This specifies the operable range of the item. The player can only use this item on tiles within the specified range. This range should be given in tiles. For example, an item with a range of five will work in a five tile radius around the character. Note that damage decreases as the distance increases.
Multimedia	This is the multimedia file played when this item is used.
Effect Flags	This is the list of all effect flags. The selected ones apply to the item. For a compendium of all effect flags, see Effect Flag Compendium
Effect Flag Information	Select an effect flag in the combo box, and then

enter data to complete the effect flag definition. For example, items that change your weapon skill need to know how much to change the skill by. This is what this information is for.

Description This is the description of the item.

Book Text File This specifies the text file to display when the **Text** effect flag is selected.

Show Filename This is the file that is displayed when the user clicks 'Show' in the Inventory window.

Animate Icon This is the icon that is used to represent an item's effect on the map screen.

Name and Display File These specify the name and the file to display on a tile when this item is selected, respectively.

Using Item Files

You can place items by checking the **Item Here** check box in the Tile Selection Dialog and entering the filename in the adjacent edit box.

Facility Editor

The Facility Editor is used to create training facilities and stores that the character can visit to improve himself. For more information, see Facilities. To learn more about using Facility Editor, choose one of the following topics:

Using Facility Editor

Facilities and Wares
Editing Facilities
Using Facility Files

Facilities

Facilities are places where the player can buy new items or improve his own abilities. Generally, facilities are very simple and straightforward, containing a list of **wares** (what the facility sells) and their prices.

Each ware has certain *attributes* that modifies what changes it performs on the character. For example, a Weapon Training ware would modify the players weapon skill. This is acheived with effect flags, just like items.

To see all of the effect flags and their data, see the effect flag compendium.

Editing Facilities

Facilities are very easy to edit. They maintain a list of **wares** (see facilities) and their attributes.

Adding a Ware

To add a ware, just click **Add** underneath the **Ware List**. A new ware called **New Ware** will appear in this list.

Removing a Ware

To remove a ware, just select it in the **Ware List** and click **Delete**.

Editing a Ware

To edit a ware, select the ware you want to edit in the **ware list**, and then modify the settings in the panel to the right. When you are complete, click OK to save the changes you have made. If you do not press OK, your modifications will be lost.

Linking a Ware to an Item

Sometimes, the purchase of a ware is directly linked to the purchase of an item. To link a ware to an item, select the **Item** effect flag (see the effect flag compendium). This indicates to the program that the ware should be linked to an item. Then type the item filename in the **Item File** box, or click **Browse** to choose an item from a list.

Setting Effect Flags and their Data

Certain effect flags require data (see the effect flag compendium). The combobox next to the effect list contains a list of all the selected effect flags. To change an effect flags data, select the effect flag in the combobox and type the data in the **Extra** box.

Setting a Wares Cost

To set a wares cost, enter the value into the **Cost** field. See the Utopian Worlds Development Suite Adventure Guidelines v1.1 for information about ware pricing.

Using Facility Files

The use of Facility Files is less straightforward than the use of other components. You can use a facility just like branching to a map. Just enter the filename into the **Branch to Map** field, and check the checkbox. Utopian Worlds will automatically detect whether the file is a map or a facility.

Animator

The Animator is used to create simple animations with frames based on icons. For more information, see Animations. To learn about using Animator, choose one of the following topics:

Using Animator

Animations
Frames
Creating an Animation
Using Animation Files

Animations

An animation is a series of frames that are played in rapid succession to give the illusion of changing through time.

Frames

A frame is the subdivision of an animation. Each frame is based on a single icon, and is completely separate from all other frames.

Creating an Animation

Animations are very simple and straightforward to create because they are just lists of frames.

Specifying an Icon for a Frame

Every frame is based on an icon for its appearance. To specify this icon, just type it in the **Frame** box at the bottom of the dialog. You can press browse to see a list of icons. Remember to trim the path information from the icons name if you want your adventure to be portable!

Inserting a Frame

When you have specified an icon for your frame (above), click **Insert Frame**. The frame you have created (above) will appear *above* the frame that is currently selected in the **Frame List**.

Deleting a Frame

To delete a frame, select it in the **Frame List** and click **Kill Frame**.

Using Animation Files

You can use animations just like icons. The entire process is transparent.

Event Editor

The Event Editor is used to run scripts and also to modify certain tiles when certain things occur. To learn more, see events. To learn more about using event editor, choose one of the following:

Using Event Editor

Events
Creating Events
Using Events

Events

Your adventure would be quite straightforward if it did not change throughout time. This is the purpose of events - to give your world a dynamic capability to change. Events can be triggered when the player steps on a certain tile, talks to a certain person or just be based on the actual progression of realtime.

The mechanism of events is based on **IDs**. Each event has a unique ID, assigned by the computer, and also a **user ID**, which the user uses to identify an event or a group of events. To perform many changes at once, just assign many events the same **user ID** and they will all be activated.

Most events can modify tiles, so there must be a way to identify which tile to modify without using arcane indices and other methods. Events use a tiles linkage value (see tiles) to determine which tile to modify.

Creating Events

All events are maintained in a master list per adventure, in a file specified in the **Event Resource** field of the adventure package. To use events, you must supply an adventure package file! This makes the addition of events transparent, and allows all events to be grouped together logically.

Creating a new Event

To create a new event, click **Add Event**, and a new, empty event will appear, ready for you to edit.

Deleting an Event

To delete an event, select it in the **Event List** and click **Delete**.

Editing an Event

To edit an event, click the **Edit Event** button, and the Event Edit Dialog Box will appear.

Event Edit Dialog Box

The Event Edit Dialog Box contains a few sentences that describe the event and its job. It is quite a straightforward concept.

An event has an **auto-activate time**. This is the roughly the number of seconds that will elapse before this event is automatically triggered. An event also has an ID. This is how you refer to the event when using trigger commands. Remember that two events can have the same ID if you want them to always be activated in tandem. Then, you choose the map filename and linkage value that the event modifies. If the event is to modify a tile, you enter the linkage value of the tile to modify and the filename of the map containing the tile. Next, describe the effect on the tile. You can either set or clear a flag of the tile. Choose the correct verb (set/clear) and the appropriate flag in the box. **Unused** flags are reserved for future expansion. If you do not wish to modify anything, choose the **None** verb. Then, choose the data member to change. If you do not want to change anything, choose **None**. Enter the new status of the data member in the next box.

Finally, an event can run a script. To do this, enter the script filename in the appropriate box.

Using Events

The Utopian Worlds program searches for events in a special resource file that is indicated in your Adventure Package file. That means you can only use events in adventures, not in solitary maps.

Triggering an Event when a tile is stepped on

Do do this, just click the **Trigger Event ID** box in the Select Tile Dialog and enter the correct **ID** (see events) into the box.

Triggering an Event with Scripting

See Scripting

Triggering an Event with PersonScripting

See PersonScripting

Container Editor

The Container Editor is used to edit containers, such as chests, bags, and the like. For more information about containers, see containers. For more information about using the Container Editor, select one of the following topics:

Using Container Editor

Containers
Using Container Files
Creating Containers

Containers

Containers are simply files that hold a list of items. Its interface is very similar to the Animators.

Players can take items from containers. In this way, containers act like very large chests. Remember that there is no limit to the number of items that can be placed in a container.

Using Containers

Containers are used in conjunction with devices.

Making a Chest

You can create a chest by activating the **Chest Device** and entering your container file in the **Device Info** field.

Making a Locked Chest

This is the same as creating a chest, except use the **Locked Chest Device** instead. For information on unlocking the chest, see Devices.

Creating Containers

Containers are very simple to create because they are just lists of items.

Specifying an Item to Add

Type the item filename you want to add into the **Item** box. Click **Browse** to choose from a list. Remember that you need to strip pathnames if you want your adventure to remain portable.

Adding an Item

Once you have specified the item you wish to add (above), click **Add Item**, and the item will appear in the **Item List**. The items are automatically sorted.

Deleting an Item

To delete an item, select it in the **Item List** and click **Delete Item**.

Utopian Worlds Development Suite Adventure Guidelines v1.1

What follows is the official publication of the Utopian Worlds Development Suite Adventure Guidelines, version 1.1. They cover each component, and provide a set of guidelines for you to follow to insure compatibility in the future and to insure a well-behaved adventure.

Maps

Although generally free-form, maps should be less than **30 x 30** in size, to ensure quick loading time and to avoid memory problems.

Maps should not rely on specific hardware configurations, specifically the presence of a 256-color graphics adapter.

Maps should not rely on the resolution of a monitor, and should provide **32 x 32** tiles in all instances unless a particular reason exists to change it.

Maps should not rely on the runtime existence of any non-standard components in the UTOPIA directory.

People

All important nouns in response text should be defined as keywords.

Every person should have a name and a valid icon image.

If a person uses speech, the text should appear in the text box for computers without sound capabilities, and as a transcript for hard-to-understand speech.

All words defined as keywords should be underlined according to the following convention: This word is _underlined_. However, some adventures will not wish to conform to this guideline, and that is acceptable.

All multiword keywords should always be underlined.

All people shall respond to the keywords **name** and **job** as well as **hello** (the greeting keyword), or their equivalents in other languages.

People should never assume their state. In other words, people should never assume if they are being spoken to in person, or over a communications device.

Whenever possible, the use of emotional data enhances people.

Items

Strict Guidelines

The absolute value of the change of HP to the user shall not exceed 500.

The absolute value of the change of MP to the user shall not exceed 1000.

The absolute value of the change of HP to the target shall not exceed 600.

The absolute value of the change of MP to the target shall not exceed 1500.

Lax Guidelines

The absolute value of the change of HP to any target should not exceed one half the maximum hit points of the hardest monster in the game.

The absolute value of the change of MP to any target should not exceed one third the maximum magic points of the hardest monster in the game.

General Guidelines

Items such as keys or generic swords should contain the initials of the game it is from to avoid confusion with other adventures. The longer the **imprint**, the better. You can make this imprint non-printable by preceding it with two backslashes. For example, a sword from Death of Sahatra might be called **Sword\\(DSAHATRA)** or **Sword\\(DS).**

Items should never have identical names. This can happen if the above guideline is not followed. If this happens, items with differing traits can be stockpiled on another item with the same name, or the user can confuse the two differing items.

Items should never have a weight greater than 25, unless it is not meant to be taken.

Items should never have subzero weight.

Facilities

Facilities should offer items at prices generally equal to their street value. A list of street values follows:

Spells: *weak: 500-3000, strong: any price*
Weapons: *generic: 100-1000, special: any price*
Items: *generic or weak: 50-750, special: any price up to 10000*
Mystics: *any price, generally not for sale*
Armor: *generic or weak: 1000-10000, special: any price up to 100000*
Implants: *any: 5000-300000*

Monsters

Monsters should not exceed a characters capability unexpectedly. For example, the casual orc should not be able to outmatch a veteran character, but a red dragon may. The character should be prepared for death matches of this sort.

Events

Events with auto-activation time of **zero** are assumed to have no auto-activation property.

Animations, Containers, Adventure Packages

No guidelines established.

About Utopian Worlds Development Suite

Utopian Worlds Development Suite is a complete adventure development package that allows you to create your own worlds and populate them with thrilling personalities, monsters and magical relics from ages past.

All adventure are based on **components**, or pieces, of your adventure. Utopian Worlds Development Suite can create and edit many types of **components**, all listed in the contents.

Universal Scripting Language

The Universal Scripting Language is currently a somewhat limited language, but does offer some fine control over the games progress that is not possible with other means of control. Below is a complete tutorial on script programming. See the command compendium for details on commands.

The Universal Scripting Language is based on the HandyParser Engine developed expressly for Utopian Worlds Development Suite, but created for overall portability. For information on using this engine in your own programs, contact the author (see adventure support).

Program Flow

In USL (Universal Scripting Language), all program flow is sequential. This means that there are no goto commands, subroutines or functions. This is why it is a *scripting language*.

USL does, however, provide rudimentary conditional commands. While not as sophisticated as a full-fledged **if**, it does provide some control over program flow.

Command Syntax

Commands are structured very similar to DOS commands, but they use commas to separate parameters instead of spaces. The command syntax looks like this:

```
command [optional parameter] [optional parameter] [optional
        parameter]...
```

Remember to enclose your string parameters in quotes!

The parameters depend solely on the command itself, but they may be either string (characters making up a word like `Good morning Bob`), or integral (like 45 and 890 - not 54.25).

Person Scripting

Person Scripting is not as powerful or as complex as Universal Scripting Language (USL). It needs to be formatted differently because of two reasons:

- First, commands need to be implanted into text and easily differentiated from normal text by the program.
- Second, commands need to have handles so Utopian Worlds can remove them dynamically from the text.

Below is the tutorial on Person Scripting. See Person Scripting command compendium for commands.

Command Syntax

Person Scripting is based on individual commands, and they have no knowledge of any other commands that may have executed. These commands all execute in succession as they appear in the text.

This is the command syntax for a Person Script command:

```
@_commandletter_,[parameter],[parameter],...#
^        ^            ^------------^        ^
|        |                   |              end of command
|        |                   parameters, based on command
|        this identifies the command to execute
this identifies the beginning of the command
```

Remember that certain commands do not need parameters, while others need many. Also, do not include a comma after the last parameter! The # sign will alleviate the need for a comma.

Universal Scripting Compendium

This compendium is a list of all commands, their parameters and their functions. These commands are for the Universal Scripting Language (USL).

Command	Parameters	Function
play	**1 (string):** the name of any valid multimedia file	This command plays the given multimedia file. It waits for its completion before returning.
trigger	**1 (numeric):** the ID of any event.	This command triggers the given event(s).
conversation	**1 (string):** any person file	This command opens the given person file, and begins a conversation with that person.
fight	**1 (string):** any monster file	This command opens the given monster file, and starts a fight with that monster.
money	**1 (numeric):** amount of money	This command adds the given amount of money to the characters current amount.
lightall	none	This command turns every tile into a fully lit tile.
jump	**1 (numeric):** the linkage value of the tile to jump to	This command moves the character to a tile with the specified linkage value.
stairs	**1 (numeric):** the change to perform to the current level	This command moves the character to the floor plus the specified amount. To move down x floors, specify **-x**. To move upwards x floors, specify **x**.
stairs_abs	**1 (numeric):** the floor index to move to	This moves the character to the specified floor.
endgame	**1 (string):** the text file, help file or multimedia file to display.	This ends the game. See the End Game Device for more information on the parameter.
display	**1 (string):** the file to display	This displays the specified file, in any of the supported file formats.
ask	**1 (string):** the type of question **2 (string):** the text of the question box **3 (string):** the title of the question box	This asks the user the question that you specify. If he answers "yes," the following command is executed. If not, the following command is skipped. Currently, the only supported question type is **"yesno"**.

runscript	**1 (string):** the script to chain to	This chains to the specified script. When that script has finished execution, the current script will resume.
runprogram	**1 (string):** the program to run	This runs the specified program. It can be a DOS or Windows program.
modhp	**1 (numeric):** the change to apply	This applies the specified change to the character's hit points.
modmp	**1 (numeric):** the change to apply	This applies the specified change to the character's magic points.
message	**1 (string):** the text of the message **2 (string):** the title	This displays the specified message with the given title. It requires the user to press a button to continue.
messagebar	**1 (string):** the text of the message	This displays a message in the message bar. It requires no button push. It is good to use the **wait** command if you have several messages to display with this command.
wait	**1 (numeric):** the number of seconds to wait	This performs a Windows-friendly wait. The execution of the script is held for the given number of seconds, but allows background tasks to continue processing. This is good for creating dynamic storylines where it appears as if a conversation is taking place.
check	**1 (string):** the parameter to check. See below for possibilities. **2 (numeric/string):** depends on the above parameter	This command will check the condition specified. If the condition is **true**, the program will execute the next statement in the program. If it is **false**, the program will skip the next statement and continue.

Possibilities for check parameter 1

What follows is a list of parameters supported for the **check** command, above. It also lists parameters and the function of each.

Possibility	Parameters	Function
has	**1 (string):** the name of any item (the actual name [Short Sword] not the filename)	This command checks to see if the character has the specified item in his inventory.
nothas	**1 (string):** the name of any item (the actual name [Short Sword] not the filename)	This command checks to see if the character does not have the specified item in his inventory.
weapon<	**1 (numeric):** any weapon skill	This checks if the character's

	value	weapon skill is less than the specified value.
weapon>	**1 (numeric):** any weapon skill value	This checks if the character's weapon skill is greater than the specified value.
defend<	**1 (numeric):** any defend skill value	This checks if the character's defend skill is less than the specified value.
defend>	**1 (numeric):** any defend skill value	This checks if the character's defend skill is less than the specified value.
damage<	**1 (numeric):** any damage skill value	This checks if the character's damage skill is less than the specified value.
damage>	**1 (numeric):** any damage skill value	This checks if the character's damage skill is less than the specified value.
magic<	**1 (numeric):** any magic skill value	This checks if the character's magic skill is less than the specified value.
magic>	**1 (numeric):** any magic skill value	This checks if the character's magic skill is less than the specified value.
mp>	**1 (numeric):** any magic point value	This checks if the character's magic point total exceeds the given amount.
mp<	**1 (numeric):** any magic point value	This checks if the character's magic point total is less than the given amount.
hp>	**1 (numeric):** any hit point value	This command checks to see if the Characters hit points exceed the given amount
hp<	**1 (numeric):** any hit point value	This command checks to see if the Characters hit points are lower than the given amount.

Person Scripting Compendium

This compendium is a list of all Person Scripting commands, their parameters and their functions.

Command Letter	*Parameters*	*Function*
e	**1 (string):** a string to display in the information area of the conversation dialog.	This command displays a string in the information area of the conversation dialog **only** if the character has Empathy active.
v	**1 (string):** same as above	This command is identical to **e**, above, except disregards the characters empathic state.
k	**1 (string):** a string that specifies a keyword **2 (string):** the name of an item (not the filename, the actual name [Short Sword])	This command branches to the given keyword **if and only if** the character possesses the given item.
s	none	Ends the conversation
t	**1 (numeric):** the ID of an event	This command triggers the given event(s).
f	**1 (string):** a valid monster filename	This command begins a fight with the specified monster, and ends the conversation
g	**1 (string):** a valid item **filename**.	This command adds the specified item to the characters inventory.
b	**1 (string):** any valid map filename	This command sends the character to the specified map
d	**1 (string):** any bitmap file	This command displays the given bitmap.
?	**1 (string):** the text of your question **2 (string):** the keyword to branch to if the response is yes **3 (string):** the keyword to branch to if the response is no	This command directly asks the user a question. Usually this function can be completed more robustly with hierarchical keywords, but this method can be used to assure a relevant and prompt response to an urgent query. This will display the given text, and branch to either given keyword, depending on the response.
x	**1 (string):** the actual name of an item (Short Sword) **2 (numeric):** an amount of gold	This command implements a trade between the speaker and the player. The user receives the amount of gold

		specified by (2) and loses the item (1) to the speaker. If the character does not own the item, the trade is unsuccessful. You should use the **?** command to ask the user if he wants to make this trade before performing it.
m	**1 (string):** any multimedia file	This command plays the given multimedia file. **This is how to implement speech.** Just record your (or someone elses) voice as a WAV file, and play it with this command anywhere in the appropriate text. Make sure to package the file with your adventure.

Maps

Maps are the representation of your world. They are essentially mosaics of different tiles, all with visual representations and attributes. You are the assembler who pieces together the different sections to create a beautiful, completed picture of your world. Your tool in this quest is the Select Tile dialog box.

You will notice that the Select Tile dialog box goes much further than just one icon. You can tell Utopian Worlds Development Suite to draw three icons by typing another icon filename in the Overlay filename box. Or, you can add other adventure elements to this tile. You can add a branch to another map or facility, a person, or an item. You do this by typing the filename of that element in the appropriate box, and clicking the check box.

When you link two maps together, you will sometimes want to implement these links as doors, or teleporters, etc. To do this, you will want to use the linkage feature. This feature links tiles with the same linkage value in different maps together. To implement a door going into a house in one map, and then the interior of the house in the other, you will want to give the doors in both maps the same linkage value, then add the branch with the **Branch to** flag. When the character steps on one door, he will appear on the other door in the other map. Be careful that these linkage values are unique. You can also use linkage values to jump between tiles in the same map. You can do this with the **Jump to Linkage** flag, and add the appropriate linkage value. This can be used to implement teleporters, staircases, pits, etc.

Maps also have **levels**. These are the Utopian Worlds Development Suite equivalent to the floors of a building or a house. The character begins on level zero, or ground level. This is where you should put all of those things that exist on the ground. The levels higher up (one, two, three, etc.) can be used to create multi-story buildings. The lower levels (negative one, negative two, etc.) can be used to create dungeons. Beware that your maps are not too large, because they can quickly eat away precious memory.

You will want your maps to be as interactive as possible. This is why tiles can be **multi-state** tiles. This feature lets you define as many different states as you want for the current tile. For example, you can place an oven on your maps. Instead of a static oven that cannot do anything, multi-state tiles allow the player to turn on the oven and receive visual (and perhaps functional) feedback. To define the different visual states, you use animations. Just define the animation with the different visual states, and then specify the animation in the tile definition. Then, you must tell Utopian Worlds that the tile is not an animation that will be changed every second, but a manual **multi-state** tile. To do this, check the **Manual Multi-State Tile** box.

When you play the adventure, you will notice a menu option that reads Use this Object. When you select this choice, the state will change and, depending on your options, a script will run. This allows you to create doors that make noise, televisions with actual moving pictures (AVI movies), radios that actually work, etc. Whenever possible, you should use this feature to your advantage, because it makes your worlds incredibly real!

Sometimes, you will want the character to have a certain item before he can pass a certain location. To do this, just click the Drop Item Here check box, type an item file in the box, and select the item name in the combo box. This will require the user to select the Use an Item... command from the popup menu, and choose the same item from his inventory. If they are identical, the tile will allow him to pass. If not, nothing will happen.

Devices are another useful feature that you can put on your tiles. See Devicesdevices for information on devices. To use a device, just select it with the combo box.

For some inspiration on creating a map, look at some of the included examples (that's what they are there for). Also, take a look at some other commercial RPGs. Look at how they implement a good story. Change their ideas around, and create your own living world. This is the power of Utopian Worlds Development Suite. No longer do you have to sit at your computer and wander around the brainchild of someone else. Manifest your own thoughts into a living, breathing world, and see what happens. Then, give it to your friends and stump them! The
possibilies and the fun are endless...

Characters

Characters are the portraiture of the player in your world. This is why they are as unique and as personal as possible. This is also why characters can be moved across adventures - so the players can develop ties with their characters over the time they play the game. This is a meaningful RPG experience.

Each character has certain attributes, and they are listed below. You will notice some are identical to monsters; this is because monsters are derived from characters.

Attribute	Description
Hit Points	Hit points are a measure of the characters life force. These points can be drained by damage, and when they reach zero, the character is dead.
Magic Points	This is a measure of the spellcasting capability of the character. Each time the character uses a spell, his magic point total decreases, and when it reaches zero, he cannot cast spells any longer.
Maximum Hit Points and Magic Points	The upper limits for these two values.
Weapon	The characters weapon skill. The weapon skill quantifies the characters aptitude at weapon handling. The higher this value, the more likely he will score a hit.
Magic	This number indicates the characters proficiency at spellcasting.
Defend	This number indicates the characters success in defending incoming blows.
Damage	This number indicates the characters aptitude at successfully doing damage.
Alchemy	This number indicates the characters aptitude at performing alchemical mixes.
Strength	This number affects the characters damage when using weapons.
Agility	This number affects the characters chance to dodge attacks.
Intelligence	This number affects the characters spell success.
Resistance	This number affects the characters resistance to damage in general.
Name and Display File	These specify the name and the file to display during battle (bitmap only), respectively.
Level	This number indicates the characters adventuring prowess, and is a yardstick for your overall progress.
Experience (Current Experience)	This number indicates how adept the character is at adventuring in general. It is a smaller resolution than levels.
Experience to reach next Level	This is the quantity of experience necessary to reach the next Level (above).
Name	This is the name of your character.

Extension Installation

Utopian Worlds Development Suite and Utopian Worlds rely on the existence of certain extensions to Microsoft Windows. If you chose not to install them when you installed Utopian Worlds Development Suite and Utopian Worlds, you can install them at a later time by following the procedures outlined below.

To install ***Microsoft Video for Windows***, simply perform the following steps:

1. Insert **Extensions Disk** into the diskette drive, and close the door.
2. From Program Manager, select **File | Run...**
3. Type **a:\msvideo\setup.exe** and then proceed with installation as instructed.

To install ***Microsoft Win32s***, following the instructions below:

1. Ensure your computer is able to run Win32s. You require a 386 or higher microprocessor and at least 4 MB of RAM. You also must be running Windows version 3.1 or higher.
2. Windows NT users need not install Win32s.
3. Insert **Extensions Disk** into the diskette drive, and close the door.
4. From Program Manager, select **File | Run...**
5. Type **a:\win32s\setup.exe** and then proceed with installation as instructed.
6. A game **Freecell** will be installed on your hard drive to test your Win32s installation.

Effect Flag Compendium

What follows is a complete compendium of every effect flag that applies to items. It also includes information about extra data and function.

Flag	Extra	Function
Activates on Use	none	This flag causes the item to activate (make these effect flags work) when the item is merely used in combat
Armor	**numeric:** the armor rating provided by this item	This flag raises the wearers armor rating.
Bag	none	This flag identifies the item as a container for other items.
Cannot be Used in Combat	none	This flag identifies the item as an item that cannot be used during combat (like a teleport spell)
Cursed	none	This flag identifies the item as a cursed item - one that cannot be deactivated without **remove curse**, below.
Death Spell	**numeric:** percent chance of success	This kills the target with the given success rate
Delete after Use	none	This flag will destroy the item once it is deactivated. To create an item that disappears when it is used during combat, just turn on the **Activate on Use** flag and this flag. That will cause the item to activate when used, and then upon deactivation, the item will be destroyed.
Detect Aura	none	This provides hazy information about the current situation
Detect Magic	none	This tells whether magic is active or not.
Dispel Magic	none	This spell turns off all magic presently active.
Empathy	none	This allows you to see the emotions of people you talk to, in certain situations.
Enemy Defend Skill	**numeric:** change to perform	This changes the targets defend skill
Enemy Intelligence	**numeric:** change to perform	This changes the targets

		intelligence
Enemy Strength	**numeric:** change to perform	This changes the targets strength
Enemy Weapon Skill	**numeric:** change to perform	This changes the targets weapon skill
Energy Disruption	none	This causes all items with the **Uses Energy** flag set to cease operation.
Examine Monster	none	This gives information about the target
Gold	**numeric:** an amount of money	This adds the given amount of money to the owners cash.
Identify Item	none	This gives information about any item the owner has in his inventory.
Impervious to Poison	none	This gives protection against poison.
Increase Agility	**numeric:** amount of change	This increases the owners agility
Increase Damage Skill	**numeric:** amount of change	This increases the owners damage skill
Increase Intelligence	**numeric:** amount of change	This increases the owners intelligence
Increase Defend Skill	**numeric:** amount of change	This increases the owners defend skill
Increase Resistance	**numeric:** amount of change	This increases the owners resistance
Increase Strength	**numeric:** amount of change	This increases the owners strength
Increase Weapon Skill	**numeric:** amount of change	This increases the owners weapon skill
Invisible	none	This makes the player invisible. Entities will no longer be able to follow him. This has no effect for entities that use it.
Magic Rating Increase	**numeric:** amount of change	This increases the owners magic rating
Mask Item Name	none	This trims the item name according to item naming conventions. See Items for more details.
Needs a Target	none	This indicates that the item requires a target to perform its duties. The program will ask for a tile on the map if the owner is not in combat, and will ask for a monster during combat.
Negate Poison	none	This destroys all poison in the users system.
Only in Combat	none	This flag indicates that this item can only be used during combat.
Poison Victim	none	This poisons the victim. Poison takes off a certain amount of hit points each turn.

Remove Curse	none	This allows the deactivation of a **cursed** item.
Simulate Full Light	none	This makes all of the tiles in the current map have a full light status.
Simulate Normal Air	none	This allows the owner to ignore air conditions.
Telekinesis	none	Allows access to devices, etc. and allows the player to take items from any distance.
Teleport Effect	none	This transports the owner to a specific place on the current map.
Text	none	This displays the text file in the **Book Text File** field.
Slowness	none	This **slows** the target. This means that he takes more damage.
Resist Slowness	none	This resists any **slowness** launched against the user.
Negate Slowness	none	This cancels the **slowness** effect.
Fire Attack	none	This labels the item as a fire attack. This renders it vulnerable to immunity by **Fire Resistance**.
Resist Fire	none	This resists any attacks labeled as **Fire Attacks**.
Extend Duration	**numeric:** the factor to increase duration by	This increases the duration of any other items used after this, until this item's duration expires.
Uses Energy	none	This indicates that the item requires energy to operate. The **energy disruption** flag uses this flag to determine which items to shut off.
Damage Up vs. Fire, Water, Air, Earth, Undead	**numeric:** the factor to increase the damage by	These flags indicate to multiply the actual damage by the given factor if the target is of the appropriate type.
Wind Servant	none	This acts similarly to **telekinesis**, but does not allow you to take items from any distance.

Facility Flag Compendium

This is an index of the flags used to modify wares in facilities.

Flag	Extra	Function
Increase Agility	**numeric:** amount of change	Increases the buyers agility
Increase Armor	**numeric:** amount of change	Increases the buyers armor rating
Increase Current HP	**numeric:** amount of change	Increases the buyers current HP count
Increase Current MP	**numeric:** amount of change	Increases the buyers current MP count
Increase Current Weight	**numeric:** amount of change	Increases the buyers current weight
Increase Damage Skill	**numeric:** amount of change	Increases the buyers damage skill rating
Increase Defend Skill	**numeric:** amount of change	Increases the buyers defend skill rating
Increase Intelligence	**numeric:** amount of change	Increases the buyers intelligence
Increase Magic Rating	**numeric:** amount of change	Increases the buyers magic rating
Increase Maximum HP	**numeric:** amount of change	Increases the buyers maximum amount of hit points
Increase Maximum MP	**numeric:** amount of change	Increases the buyers maximum amount of magic points
Increase Resistance	**numeric:** amount of change	Increases the buyers resistance
Increase Strength	**numeric:** amount of change	Increases the buyers strength
Increase Weapon Skill	**numeric:** amount of change	Increases the buyers weapon skill
Item	none	This indicates that the purchase of the ware gives the item in the **Item** field to the buyer.
Unused A	none	Unused

Adventure Design

Getting Started

Designing an adventure is a monumental yet rewarding task. You are given the opportunity to weave a complex story filled with living people, monsters and ancient temples that leap from the pages of mythology texts. But in order to create a powerful adventure, you first must create the world that your story will take place in. Here are a few guidelines to customizing your world.

Your World

Your world is a place that previously existed only in your imagination, but now you have the chance to breathe life into it. These are ideas about making your world interesting.

1. First, create the ancient history of your world. This should always involve the religion of the ancient peoples. Create stories of the gods aiding the tribes when they were warring, or jealous gods who gave the advantage to one side. Use the popular Greek and Roman mythological tales for ideas, or look at other cultures and their beliefs (Native Americans, Norse, etc.). Or you can depart from earths development totally and create something radical. Remember, this is **your world**.
2. Next, you need to make a map of the important places in your world. Try to tie these places into the myths you have created. Try to give the player one piece of an association (like a story about a city) and then when he actually goes there, he experienced the thrill of realization and connection. Use bone-chilling stories to help set the mood.
3. The people of your world will be a very important part. They should tell the player stories that evoke some sort of emotion in the human player of the game. Dont just make a little running around quest where the player goes and fetches items and then gets sent to fetch more items. The player gets bored with this type of game **very** quickly. It may take some practice to get the creative juices flowing - just dont give up. Just as writers sometimes encounter creative dry spells, you will also.
4. Create intense relationships between the people of the world also. It will seem very artificial to the player if the people in his world do not interact. Make some people despise each other, while others are in love.
5. Use books to tell stories as well as people. Books are often a very overlooked feature, but they can pack a mean punch. Dont forget to hook your audience in the first line! If this book has a major clue, the player may not read it because it bores him. Remember, this is to entertain the player, so do not let up for one second!
6. Overall, make your adventure **fun**! Everyone has a sense of exploration and adventure. When the fear of personal harm is overcome, (usually only subconsciously - the player still fears for his characters life) the player can let his explorer side loose. Let him flourish by giving him the usual suite of great adventure components - living people, intense places, myth, adventure, clues and red herrings. If you dont like your first try, start over with a new story and a new world, or try to repair it. Usually, starting over will give you a fresher perspective than a tedious repair job.

Adventure Directories

The first step in starting a large adventure project that will be distributed is to make a separate directory for your adventures. You can make this directory **anywhere**, but there are a few procedures to follow.

First, make sure that any non-standard components are in your directory, not the main program directory.

Second, make sure that none of your custom made components have pathnames. Relational pathnames are fine as long as they proceed down the directory hierarchy. For example, **..\sample3.ent** is not acceptable because it relies on a directory structure that may not be present on the end users machine. This notation is, however: **subdir\sample3.ent** as long as the directory structure is echoed on the end users machine.

Also, when you create components, you may want to use the Options | Change Directory... command so the browsing dialog boxes browse the right directory.

Specifying Filenames

When entering a filename into any field, make sure it does not contain an absolute path to the file or a relative pathname that goes up the directory hierarchy (like **..\sample.map**). In most cases, you should use just the name. This will ensure portability. Most fields support NTFS, HPFS and VFAT length filenames (up to 255 characters), but some support only 128 in order to save space.

Registration Information

Utopian Worlds Adventure and Expansion Advanced Registration

To register your adventure or expansion package with advanced privileges, simply fill out the form below completely and accurately, and mail to:

James Payne
Utopian Worlds Adventure Registration Code B
5 Daniel Road
Hopkinton, MA 01748-2434

Please enclose a check made out to James Payne for the amount of $49 US for adventures and $69 US for expansion products. Expansion products are classified as any products that cannot be directly played by Utopian Worlds and are designed to accompany the use of Utopian Worlds Development Suite or Utopian Worlds. Instead of the one-year catalog listing term of adventures, expansion products will receive three-year terms.

These are the benefits obtained by advanced registering your adventure:

- All benefits of normal Registration
- A listing in the official Utopian Worlds Adventure catalog (See Catalog note)
- A listing in the help of the next version of Utopian Worlds
- Free upgrades to the next major version of Utopian Worlds & Development Suite)
- Optional advertisement space on the distribution disk (See the Benefits Described note)
- ResourceLink system for finding development partners

If you are a charter developer, you receive these benefits in addition to those above:

- Free adventure registration ($4.95 for confirmation and processing)
- 10 free catalog listings
- Free upgrades until version 3.0

You will receive confirmation via E-Mail or via US Postal (no SASE required) which will include your registration access number. This will give you access to the benefits listed above. For security purposes, your access number may be used only once per offer.

----- Registration Form -- Cut Here -----
Name____
Address 1____
Address 2____
City_______________________________________State________ZIP__________________

E-Mail Address (Internet)____
Check here for US Postal confirmation: [ ]
Check here for information on optional adventure review and distributed testing: [ ]

Check here for information on advertisement space on the distribution disk: [   ]
Check here to include this adventure in the catalog listing: [   ]
Check here to include the following description in the catalog listing: [   ]

____
____
____
____
____
____

Note: This service requires an extra $20 US.

In the catalog, you can either have a price quoted, and let the user order through us, or you can provide one line of text directing the user to you. The former is free, however we add a small 5% surcharge to cover postage, etc. The latter will cost an extra $200 US, to protect our catalogs reputation. We may also attempt to verify the address and phone number given to us.

Check here for quoted price: [   ] The price is: ______________________________________
Check here for text direction: [   ]

____

Adventure Title:____
Adventure Author:____
Adventure Version:_____________________

Total Amount Enclosed: $____

----- Cut Here -----

Benefits Described

Optional Advertisement Space on Distribution Disks:

This option will place a text banner on the distribution disks that can be displayed during the user's installation process. This will provide a good form of advertisement for your adventure, and can contain a good amount of text. This same banner will be contained in ADVERT.TXT. The cost depends upon the length of the ad and current conditions. This advertisement will only be placed on one release of disks.

Catalog:

The catalog is perhaps the best area of inexpensive advertisement for your adventure. The catalog will be distributed on online services, in promotional flyers and on distribution disks. It will be an attractive RTF and TXT document that provides users with information on your adventures and how to order them. Here is a sample entry:

> *Sample Adventure* by Bob McNally
> Trek through the unknown halls of BobLand, where the unexpected happens on every turn. It's a maze of intrigue and a grapevine of secrets.
> Order direct at 1-800-xxx-xxxx $19.95

or, you can direct your customers to you:

> *Sample Adventure* by Bob McNally
> Trek through the unknown halls of BobLand, where the unexpected

happens on every turn. It's a maze of intrigue and a grapevine of secrets.
Order from Bob 1-508-xxx-xxxx $19.95

If you choose to enroll yourself in the catalog, your term will last for 1 year. Your first term will cost $55 US. After that, it must be renewed for $25 US. Each renewal lasts 1 year.

File Menu

New >>	Gives a menu of the component editors (see contents)
Open...	Opens an existing file with the correct editor
Save	Saves the current file without altering its name
Save As...	Saves the current file with a new name
Exit	Leaves Utopian Worlds Development Suite

Options Menu

Registration >>	Provides access to registration choices
Basic	Basic registration
Advanced	Advanced registration
Tools >>	Provides access to frequently used tools shipped with Windows
Change Directory...	Changes the current directory.

Window Menu

This provides functions to control the windows open in Utopian Worlds Development Suite. This is identical to the Window Menu in many Windows applications, including Program Manager.

Help Menu

This menu accesses this help information.

Set Information Dialog

This dialog lets you set the attributes of all tiles placed after this dialog is closed. You can set information about **lighting** and **air quality**.

Lighting Settings

Setting	Description
Full Lighting	This has no effect on the appearance of the tile.
Sunlight	This puts a yellowish haze on the tile, to simulate the appearance of sunshine.
Bright	This makes the tile appear that it is under extremely bright white light.
Mildly Bright	Puts a slight haze on the tile.
Dim	Puts a thick haze on the tile.
None	Puts an extremely thick haze on the tile.

Air Quality Settings

Setting	Description
Breathable (Normal)	No effect.
Mildly Toxic	Takes off 3 hit points when the tile is stepped on.
Highly Toxic	Takes off 10 hit points when the tile is stepped on.
Poisonous	Poisons the character.
Unbreathable	Takes off 50 hit points when the tile is stepped on.
Deadly	Automatically kills the character

Basic Registration Dialog

This dialog simply has fields that you would normally see on any mail in card. It is self-explanatory. After you are finished filling out the form, you will be prompted for a filename. This is the filename where the information you entered will be stored. Print out this file and then mail it, following the registration guidelines to register your adventure.

Devices

Devices are **manipulatable objects** that the character can use to serve a specific purpose. There are a certain number of **device types** currently supported by the game, and the following is a list of them.

Device Types

Type	Description
Door Device	This device has a special function. It searches for all tiles in the map with the linkage value specified in the **device info** field. It then switches the **unpenetrable** state of each tile. This is good for constructing lever puzzles.
Comm Device	This allows the character to talk with people when they are not near him, or even on the same map. The character searches for the persons name to make the call.
Book Device	This opens the text file in the **device info** field, and displays it. It can support any of the Utopian Worlds file formats. (See File Formats)
Multimedia Book Device (MM Book Device)	This opens the multimedia file in the **device info** field and plays it.
Open This Door Device	This toggles the **unpenetrable** state of the current tile.
Torch Device	This changes all the tiles **lighting** to **full**.
Toll Device	This asks for payment, and upon receiving proper payment, it allows passage. The amount required for passage is in the **device info** field.
Password Device	This is similar in function to the **Toll Device**, except that it allows passage upon receipt of the correct password. The correct password is given in the **device info** field.
Chest Device	This places a container (the file given in the **device info** field) on the current tile. The character can open it at will by using the **Open Container** command.
End Game Device	This device ends the game when the character steps on the tile containing this device. It prompts the user to save his character, and then it displays a closing file, given in **device info**. The file can be any of the supported file formats.
Locked Chest Device	This has the same function as the **Chest Device**, but the chest is locked. This means that the user has to use the item given in the **Drop Item** field in order to gain access to the container.

Locking Components

The Utopian Worlds Development Suite Locking capability is a feature that lets you have a modicum of security when distributing your adventures. Locking discourages modification to your work. While the locking capability is somewhat weak, it will deter the casual user.

If you are using Windows NT, you can use the **LOCK32.EXE** and **UNLOCK32.EXE** programs instead.

Locking your Components

To lock a component, use the **LOCK.EXE** DOS command-line program. Pass your component filename as a parameter. The program will attempt to identify the component as a valid component, and then it will perform the necessary modifications to lock your component. You will be prompted for a password.

Unlocking your Components

To unlock a component, use the **UNLOCK.EXE** DOS command-line program. Pass your component filename as a parameter. The program will attempt to identify the component as a valid, **locked** component. If this test is successful, you will be prompted for a password. If your password is correct, the file will be automatically unlocked. Note that no feedback is given for password correctness.

Command Line Options

The Utopian Worlds Development Suite command line is very simple. Just pass the component to open as a parameter. For example, to open **CMDLINE1.MAP**, run Utopian Worlds Development Suite like so:

```
devsuite cmdline1.map
```

Creating Windows Help

Microsoft Windows v3.0 and higher comes packaged with an incredibly robust and easy-to-program help engine. Previously, help was used only to accompany software packages, but many users and programmers alike are realizing its benefits as an information medium.

Utopian Worlds Development Suite and Utopian Worlds fully support Windows Help files. Generally, wherever a text file is allowed to be entered, a Windows Help file can be used in its stead. Make sure you use the proper conventions to accomplish this.

To create Windows Help, you need to have a background in creating RTF files and HPJ help projects. If you are familiar with this, you are ready to create help. If not, just follow along with the bare basics, and refer to magazine articles for more in-depth information.

First, you must create your RTF file. This is a word processor file, saved in Microsoft Rich Text Format. If your word processor cannot support this, you are pretty much out of luck. You can use fonts, colors, sizes, complex formatting, tables, graphics, whatever. Windows Help will probably support it as long as it does not occupy too much memory (graphics are usually the culprits in those cases).

Next, add the topic ID to the top of your document. This will be a footnote using the # mark. Put a string like **contents** or **main_index** in the footnote space. Windows Help uses this internally to navigate between topics. But that is beyond the scope of this mini-tutorial.

Now, create the HPJ file. It should look like this:

```
[OPTIONS]
contents=(the text you entered at the # footnote)
title=(the title of your document. Include spaces, etc.)
warning=2

[FILES]
(the name of the RTF file you created)
```

That is all there is to it! You are now on your way into the world of portability. This document can be read on **any** computer with Windows! There is one last step - compiling your help document. This is done with the HC31 help compiler, available on many online services and bulletin boards. Run it like this:

```
        hc31 (the HPJ file you created)
```

You will see periods flashing by, indicating progress. Sometimes, a warning or two will flash by, but you can generally ignore them. Your new file will be the same name as the HPJ file, but with an HLP extension.

You can view this file with the WINHELP.EXE program included with Windows.

Entities

Entities are alive. This is the best way to describe them. They resemble *people* in their ability to conduct conversations, but they are far more than this. They can speak to the player and they can also try to attack him. Even friendly entities can be killed for their goods or money. They are governed by a sophisticated artifical intelligence engine and they feature seamless integration into Utopian Worlds.

The power of entities comes in the seamless integration of entities into Utopian Worlds maps. Instead of static people and monsters who are tied in one position, entities can move around the map. This provides an effortless combination of battle and adventuring. No longer are battles in a different window with archaic commands. This method also provides the opportunity for long-distance attacks and more strategy. Utopian Worlds was never meant to be a realtime hack-and-slash adventuring system.

Remember that entities are living beings. They are not just blind killing machines, lashing out at any sign (or smell) or living flesh. They need to live and reproduce, just like all animals on earth. In dungeons, entities tend to make themselves at home. They can't rely on the fact that adventurers come tramping through the dungeon constantly, giving a fresh supply of meat each day. They set up a base and begin to scavenge. When you are creating a dungeon, keep this in mind. Your characters will probably chance upon the living quarters of these kinds of beasts often. Friendly entities need living quarters as well, but they will probably be more civilized. Remember to keep the dwellings of your creatures compatible with their personalities.

See People, above, for more details on the more personable aspects of entities.

Each entity has a list of attributes, shown below:

Attribute	Description
Hit Points	Hit points are a measure of the entitys life force. These points can be drained by damage, and when they reach zero, the entity is dead.
Magic Points	This is a measure of the spellcasting capability of the entity. Each time the entity uses a spell, his magic point total decreases, and when it reaches zero, he cannot cast spells any longer.
Maximum HP and MP	The upper limits for the monsters Hit Points and Magic Points, respectively.
Experience	The experience given to the player when he defeats this entity.
Money	The amount of gold given to the player when he defeats this entity.
Weapon	The entitys weapon skill. The weapon skill quantifies the entitys aptitude at weapon handling. The higher this value, the more likely he will score a hit. This value should not exceed twenty.
Defend	This number indicates the entitys success in defending incoming blows. This value should not exceed twenty.
Damage	This number indicates the entitys aptitude at successfully doing damage. This value should not exceed twenty.
Strength	This number affects the entitys damage when using weapons. This value should not exceed twenty.
Agility	This number affects the entitys chance to dodge attacks. This value should not exceed twenty.

Intelligence This number affects the entitys spell success. This value should not exceed twenty.

Resistance This number affects the entitys resistance to damage in general. This value should not exceed twenty.

Name and Display File These specify the name and the closeup display file, respectively. The closeup display file is shown whenever the user requests it.

During This is the multimedia file played when the Conversation is started.

Cannot reach with GlobeNet This means that the person cannot be accessed using GlobeNet (i.e. communications devices)

Cannot find with GlobeNetQuery The equivalent of an unlisted number

Every entity you create has an **attitude**. This defines the demeanor of the entity and the result when he is attacked. There are three attitude types, listed below:

Attitude	Definition
Friendly	Friendly entities will not attack the character in their default state. When the entity is attacked, however, he will become hostile towards the attacker. They will naturally roam around the map.
Neutral	Neutral entities, like friendlies, will only attack the character when provoked. However, they do not roam the map like friendlies.
Hostile	Hostile entities always attack the character.

An **attitude** setting is also added to the entity record. This value can be **Friendly**, **Neutral** or **Hostile**. Hostile entities will always try to attack the character, while Friendly and Neutral entities will not. The **No Respond** flag will determine if the entity responds when talked to. Note that even Hostile entities can carry conversations with the character.

Entities use a sophisticated artifical intelligence engine that allows you to choose the entities personality. Unlike other artifical intelligence engines where monsters always attack you regardless, the entity can have up to 16 different personalities. Below is a list of each choice that you can make.

Note: the dominant choices, marked with a (D) are favored during entity reproduction. Entity reproduction occurs when two entities are close together. They will create a new entity based on the fathers and the mothers traits.

Entity Personalities	Description
Offensive (D)	This personality type defines the entity as an offensive being, meaning that he will attack the player in preference to healing himself. It is the dominant trait.
Defensive	This personality type indicates that the entity will tend to his own wounds rather than attack the player.
Aggressive (D)	This personality type indicates that the entity will prefer close combat over long range, strategic combat. It is dominant.
Hesistant	This personality type indicates that the entity will prefer long range, strategic combat over close range combat.
Smart	This personality type indicates that the entity knows about itself, its weaknesses, and is able to formulate a strategy

of attack.

Stupid (D) This personality type indicates that the entity is governed by animalistic instincts. This is dominant.

Strong (D) This personality type indicates that the entity is strong. It is defined for the sake of smartness - if he is smart, then he will know he is strong. This is dominant.

Weak This personality type is the opposite of strong.

Entity types are also used when calculating damage against the entity. For example, the player may have an Ice Sword - which would be particularly effective against Fire beings. You can set any combination of the five types available.

For details on creating entities, including guidelines and ideas, see People.

Entity Editor

The Entity Editor is used to create entities. To learn about them, see Entities.

Adding Keywords

To add a keyword, simply type the keyword into the keyword entry box, and click **Add**. The new keyword will appear in the list to the left, automatically alphabetized.

Removing Keywords

To remove a keyword, select the keyword you wish to remove in the list, and click **Delete**.

Assigning and Editing Response Text

Response text is always paired with a keyword. To edit the text that is paired with a certain keyword (for example, hello), simply select the keyword from the list, and begin typing (or editing) text in the large, multiline edit box below. To save the text you have entered, just click anywhere else in the dialog box.

Adding Items

To give the entity an item or two, simply type the item filename into the appropriate box. It is the edit box inside the **Items** group. Then, choose **Add**. The item will appear in the list. Note that you can choose the **...** button for a list of items.

Removing Items

To remove an item, just select it in the item list and click **Delete**.

Setting Personality and Type

To set the entitys personality and type, click **Genetics Laboratory**. You will see a dialog box containing personality choices. See Entities for information on the personality choices.

Setting Attitude

To set the entitys attitude, just select the appropriate choice in the combo box.

File Formats

This is a list of all the supported file formats and their uses.

Utopian Worlds Development Suite has powerful information handling capabilities. It can interpret and display many different file formats, each having distinct advantages and disadvantages.

Utopian Worlds Development Suite interprets each file based on its extension, which will be given in the following information. Since some file formats do not have distinct identifiers, this method is the only feasible one that provides maximum flexibility. This is why your files **must** have the correct extension to be properly interpreted.

You can use any of the following formats whenever a file is to be used for display purposes. These occasions include book devices, script commands (**display**) and tile descriptions. Note that tile descriptions will be interpreted as files when the description begins with a [.

Text Format (TXT)

The text format is very straightforward and easy to create. You can use the built-in Text Editor for this purpose, or you can use your favorite editor (like Notepad).

Windows Write Format (WRI)

The Windows Write format is the format of word-processing files output by the **Write** program bundled with Microsoft Windows. Since Utopian Worlds does not directly interpret and display these files, the **Write** program will be started with the appropriate file displayed.

Windows Help Format (HLP)

The Windows Help format is a completely portable hypertext document format supporting graphics, fonts and advanced linking capabilities. This format is recommended for **all** documents because of its robustness. Also, Windows Help files cannot be easily modified by the user. For assistance on creating Windows Help files, see the online documentation. Also, an example of this process is included on the distribution disks.

Universal Script Language Format (USL)

The Universal Script Language is used to run scripts that you create using Utopian Worlds Development Suite's built-in scripting language.

Windows/DOS Executable (EXE)

Windows and DOS both use the same extension and format for their executables. Win32, the new 32-bit Windows supported by Windows NT, uses the same extension, but not the same format. You will only be able to use Win32 executables in Windows NT, Windows 95 or Windows 3.1 running Win32s. This way, you can run programs instead of just displaying files. Under Windows NT, you can also run executables supported by any of its numerous subsystems such as OS/2 1.x character mode programs.

DOS Command Files (COM)

Most DOS Commands come shipped in this format.

DOS Batch Files (BAT)

DOS Batch files are useful when you have several DOS-based operations to accomplish.

Universal Scripting Language (USL)

This is a standard Scripting file.

Windows Bitmap Format (BMP)

The Windows Bitmap format is a simple, uncompressed graphics file that is used for many Microsoft Windows operations. While the Windows Bitmap format is certainly not the most compact or efficient format, it is very popular with many paint and conversion programs. Bitmaps will be displayed in a private window. Bitmaps are sometimes favored over compressed formats like PCX because of their fast display time.

PC Paintbrush Format (PCX)

A popular DOS Paint program, PC Paintbrush, exports this type of file. While it is not common on modern Microsoft Windows machines, this type of graphic abounds on older MS-DOS machines and on bulletin board systems. It is also widely used in clipart collections. PCX Format has the overhead of compression, so displaying PCX graphics is noticeably slower than BMP format.

Targa Format (TGA)

The Targa format is a powerful graphics format designed for the professional because of its 24-bit color capabilities.

Windows Waveform (WAV)

MIDI Format (MID)

Windows Audio-Video Interleave Video Format (AVI)

AutoDesk Animation Format (FLI and FLC)

These multimedia formats are supported with the appropriate drivers, included with Utopian Worlds Development Suite.

Alchemy Rules

The Alchemy Editor is designed to help you define alchemy in your adventures. Utopian Worlds looks for alchemy rules in the file indicated in the adventure package **Alchemy Resource** field.

Creating a new Rule

To create a new rule, click on the **Add New Rule** button.

Deleting a Rule

To delete a rule, click on the **Delete Rule** button.

Editing a Rules Attributes

To edit a rule, select it in the **Rule List**. Then, type the filenames of the two items that will be **ingredients** in the appropriate boxes (to the left and right of the plus sign). Then, type the filename of the **result** of the mixture in the appropriate box (below the plus sign).

Alchemy

Alchemy is the outdated science of transmuting certain objects into other objects, usually in strange, arcane brews. Mostly, alchemy was used to turn lead and other worthless metals into gold. However, in Utopian Worlds, alchemy takes on a much broader meaning. Your players can mix two items that they have to create an entirely new item. The player will have no advanced knowledge of the success, or even the possibility, of the mixtures that you define.

General Rules

There are several general rules that you must understand when creating Utopian Worlds adventures.

First, you must understand **Display Files**. You will see this phrase sprinkled throughout the documentation. A display file is any type of file that can be shown, read or run and that is supported by Utopian Worlds Development Suite and Utopian Worlds. (See File Formats). Whenever Utopian Worlds Development Suite asks for a display file, you can enter the name of any file of the supported types.

Second, you must understand **Icons**. Icons are usually only asked for in certain instances. When you are asked for an icon filename, you can actually give the name of any of three iconic graphics formats (Windows Icon, Windows Bitmap and Windows Metafile), but you must understand that the picture will be shrunk to fit the tile size.

Also, there are certain fields that have the name of a certain occurrence in them, such as **Activate**. These fields should contain the name of a file to play/show when this event happens.

Genetics Laboratory

The Genetics Laboratory allows you to specify a personality for the entities that you create. See Entities for information on available personality types.

Auto Links

Auto Links allow you to easily connect maps to create a large, panoramic terrain. You can specify a map to link to in each of the four directions: up, down, left and right. When the character tries to move off the map in any of these directions, he will be taken to the appropriate map.

For example, you can define the Up Auto Link as **sample2.map**. When the character steps on the uppermost row of tiles, he will be immediately taken to **sample2.map**, positioned at the bottom and at the same horizontal position.

Adventure Packages

Adventure Packages are files that simplify playing an adventure for the user and indicate important information about the adventure itself. If you want to use alchemy or events in your adventures, you must use an adventure package so Utopian Worlds can find the appropriate event and alchemy files.

Field	Description
Adventure Name	The name of the adventure. This is displayed in the title bar when the adventure is played.
Adventure Author	The authors name.
Copyright Info	A copyright statement for the adventure.
Description File	A standard text file to be displayed when the adventure begins.
Initial Map File	The map file that the adventure begins on.
Event Resource	The name of the event file that will be used in this adventure.
Alchemy Resource	The name of the alchemy file that will be used in this adventure.

Browse Dialog

The Browse Dialog helps you choose components from a list of files. It is a very simple dialog box, resembling closely the Microsoft common dialog boxes. However, it does not allow you to freely change directories. This is because Utopian Worlds adventures must retain portability. So, this dialog allows you to access only specific directories.

The top combo box contains a list of directory names. It contains the Utopian Worlds directory, the current directory and several Object subdirectories in the main Utopian Worlds directory. You can choose from any of the listed directories.

The next box down is the file type box. This is where you can choose different file types to select from. The listed file types are the ones supported by the current operation you wish to perform. If you wish to choose a file without a standard extension, you can choose All Files, available in all situations.

The large list box is the file list box. When you select a file and it is a graphical image, a thumbnail preview will appear to the right. This will help you choose display files for many situations. To choose a file, double click it or press Ok. To abort, press Cancel.