

ADVENTURE CREATION KIT v3.2

The Adventure Creation Kit (ACK) game engine and editor system was designed and programmed by *Chris Hopkins*.

Hearty thanks go to *Joshua Lawrence* for resurrecting interest (including my own) in ACK by creating and promoting the ACK website; for creating the ACK Launcher, making ACK available to a wider audience; for his help with getting the objects and creatures in the new kits completed; and for his extensive beta-testing work on the new release of ACK, without which the 3.0 version would not have been possible.

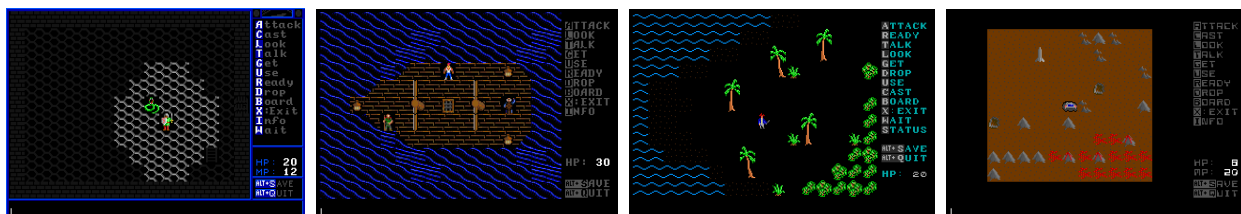
Additional thanks go to *Seth Hopkinson*, for helping me release ACK as freeware several years ago (if anyone can find him, let me know); also to *Robb Sherwin*, for helping to promote ACK and providing space on the Jolt Country forums for the ACK community; to *Mickey Callahan* (for beta-testing the original release of ACK); and to the DosBox team (*Sjoerd v.d. Berg*, *Peter Veenstra*, *Ulf Wohlers*, *Tommy Frössman*, and *Dean Beeler*) for creating an emulation platform that allows a 19 year old game (ACK was started in 1989, so parts of this game engine are old enough to vote!) to run remarkably well on modern computers.

Thanks also go to *Joshua Steele*, for his Ultima IV tileset (a couple of which are used in the kits), and to the crew at EUO (<http://swut.net/euo>) for creating some additional inspiration on graphic tiles. And in the case of the Ultima kit, credit for the content goes to *Richard Garriot*, *Denis R. Loubet*, *Kenneth Arnold*, and *Douglas Wike* (and no doubt many others) of Origin Systems.

Finally, thanks go to *Stuart Smith* for creating Adventure Construction Set, the original inspiration for ACK.

User Manual - Table of Contents

About ACK:	2
Getting Started:	3
Creating ACK Adventures:	8
ACK Editor Modules, Overview:	9
CONFIGURE ADVENTURE: Setting Global Settings:	11
IMPORT FILES, EXPORT REPORTS: Import and Export Game Data:	12
EDIT FONT: Edit the typeface used by your adventure:	13
EDIT GRAPHIC TILES: Edit graphics for your terrain, items, and creatures:	13
EDIT OBJECTS, ITEMS, TERRAIN: Manage your terrain, weapons, items, etc.:	15
EDIT MESSAGES AND DIALOGUE: Manage text content for your adventure:	19
EDIT MAPS AND REGIONS: The Map Editor:	22
EDIT PEOPLE AND CREATURES: Defining your NPCs (Non-Player Characters):	24
EDIT MACROS: The ACK Scripting Language:	25
Misc. Tips and Tricks:	30
Custom Content: Pictures, Sounds, and Skins	38
Appendix A: Running ACK in DosBox:	40
Appendix B: What are all these files in my adventure folder?	41
Appendix C: Distributing Standalone Adventures	42
Appendix D: Game Walkthroughs	43



About ACK

ACK is a complete system for making your own 80's-style RPG adventure games. Games will have an overhead perspective, and be mostly key-driven. You use a top-down perspective, where all "objects" around you (terrain, items, creatures) are represented by "graphic tiles" (16x16 pixel images tiled out into a grid.) ACK was inspired by another, much earlier game called Adventure Construction Set, by Stuart Smith and published by Electronic Arts. Those who were familiar with ACS will notice many conceptual similarities between the two games:



ACK also had other inspirations; most notably the Ultima series of games (particularly Ultima V) and in fact the newest build of ACK, v3, includes a large number of enhancements designed to make Ultima-style games even easier to implement. Those who were familiar with ACK v2 will notice that a second set of graphic tiles has been added, the "darkness" effects have been overhauled, experience and levels have been implemented, the combat engine has been improved, portal features have been enhanced (so moongates can now be implemented), music support has been improved, wandering monster features have been fixed, the conversation system has been brought to an Ultima VI level (including portraits) and support for externally-created pictures (including using them as "skins") has been improved.



Getting Started:

Installing ACK

ACK is bundled with Joshua Lawrence's "ACK Launcher" that will run (via a DOS emulator called DosBox) the main ACK menu or any of the individual ACK editors. No installation is required; simply unzip ACK to a folder of your choice. (If you are upgrading from an older version of ACK: back up your old games out of the ACK\GAMES folder, replace your ACK folder with the new version, and copy your games back in.)

Once you have unzipped ACK to a folder, just run the "ACK Launcher" program.



From here, if you want to use the normal ACK main menu, you can click the "Go!" button to load the main ACK application, after which you can close the ACK Launcher if you like. There are also check-boxes for enabling mouse support in ACK (which some people may prefer for playing an ACK game, although the editors do not take advantage of the mouse) and for changing the graphic scaling options (see the troubleshooting section below.)

Or, you can use the ACK Launcher to directly choose an adventure and play it or run the editors. To use the ACK Launcher this way, first click "Select Adventure" to choose one of the adventures in the ACK\GAMES folder. (Or if you want to start work on a new adventure, click "Create Adventure" first, then "Select Adventure" to choose the game you just created. After selecting an adventure, you can use the editor buttons to launch the individual ACK editors. The editors will launch with 2x scaling (regardless how you have the options set above) so that you can fit more editors on the screen. You can run multiple editor windows, and press TAB to synchronize them to each other.

For example, you may want to have the graphic tile editor, object editor, and map editor all open at once. While working on a map, you may realize you need a signpost. You can switch to the graphic tile editor window, draw the new graphic tile, and press TAB to immediately save it. Then switch to the object editor, press TAB to load the new tile set, and create your new "SIGNPOST" object with the new tile. Then press TAB again (to save the changes to the objects list) and switch to the map editor. Pressing TAB in the map editor will load any new graphic tiles and objects – so you will then be able to start placing your new signpost on the map!

Troubleshooting and Optimizing

Video issues: ACK runs in a 320x200 video mode, and DosBox scales that up to a more readable resolution. The default settings scale it up by 3x and perform aspect-ratio correction. Not all computers will perform the same, however, so you may wish to experiment. The images below show (in close-up) how the aspect and scaler affect the quality of the output:



Normal3x scaler, with aspect correction on (default)



Normal 3x scaler without aspect correction



Hq3x scaler with aspect correction (recommended)



Hq3x scaler without aspect correction

Sound issues: in testing the Mac version of the launcher, Josh found that sound was choppy on the iBook G4; if you encounter similar problems you may want to try reduce the CPU load (see below) or experiment with different settings in the dosbox.conf file.

Performance issues: If performance is poor (either in ACK or on your computer when you run ACK) then you can change the CPU load of the DosBox window with the Control-F11 and Control-F12 keys (to decrease/increase CPU cycles) and the Control-F7 and Control-F8 keys (to decrease/increase the frame skip.).

Keyboard issues (Macintosh): ACK often requires you to press Function keys, like F10 to exit an editor. On newer Mac laptops, the function keys are by default mapped to hardware functions, so in that case you will need to hold down Fn + F10. You will also need to hold down Fn to use DOSBox's function key controls -- for instance, to slow down cpu cycles, you will need to hold down Fn + CTRL + F11. If you want to use the function keys in ACK and DOSBox without having to hold down Fn, launch OS X System Preferences, go to the Mouse & Keyboard menu, and check the box that reads 'Use F1-F12 Keys to control software features.' Note that after doing this, pressing Fn + F1 - F12 will control hardware features, and you can uncheck the box to revert to standard function key behavior at any time.

Patching and updating ACK: If minor bug fixes are released, they may be done as a "hot fix" which includes just a collection of EXE and OVR files; to apply a hot fix, just replace your existing files with the new ones. Major new releases will be a complete copy of ACK, and you should copy out your GAMES folder, replace your copy of ACK with the new one, then copy your GAMES folder back in.

ACK Community

If you have problems running ACK, or discover gameplay bugs where features do not work as they should in certain situations, we'd like to hear about it. As of 2008, the author of ACK, Chris Hopkins, can be contacted at adventurecreationkit@gmail.com.

The author of the ACK Launcher (and expert on using ACK in the Mac environment) can be reached at jjsonick@yahoo.com – he also runs the website for ACK, at <http://www.mozomedia.com/ack>.

New versions of ACK (including PATCH.DAT files to update your copy of ACK to the most current version) can be found at <http://www.80sgaming.org/ack>

If you want to get in touch with other ACK users, you're encouraged to come visit the ACK discussion thread on Robb Sherwin's Jolt Country forums, at <http://www.joltcountry.com/phpBB2/viewforum.php?f=16>.

Playing an ACK Adventure

Every ACK adventure will play a bit differently – some might have no combat at all, while some might be primarily hack and slash; some might be set in a fantasy world with magic, while some might be set in the future. However, certain elements are common to all ACK adventures, such as the keyboard commands:

Arrow-keys: move your character about the screen.

A: Attack. If you have a weapon ready, you will be prompted for where to attack.

T: Talk. Prompts interaction with others.

L: Look. Gives you the name of whatever you target, and (if looking at someone else) tells you if others are hostile. Will use the object (open a box, etc.) if you are close enough. (To use items you can pick up or are already carrying, use the Get and Use commands)

G: Get. Picks up items, if you are close enough.

D: Drop. Drops items on the ground, in case you no longer need it or it's too heavy to continue carrying.

U: Use. Uses items. (Sometimes used to ready weapons or armor, etc., if the adventure is set as such)

R: Ready. Prepares weapons and armor for use. (May be done with "Use" instead, if the adventure is set as such)

C: Cast. Cast a spell. (Might be done with "Use" instead if the adventure is set as such)

B: Board. If you find a vehicle or other conveyance, like a horse or a boat, you can board it with this command.

X: Exit. Used to dismount a vehicle you boarded with the Board command.

I: Player Info/Inventory. Gives you information about your character.

Space: Pass. Pass one turn.

Different ACK games may have other key commands as well; for example, some games may implement commands like "Wait", or "Search", etc. Finally, there are a few system commands:

Alt-S: Saves your game.

Alt-Q or F10: Quits the game. Asks if you want to save. (If you're quitting to reload a saved game, say no.)

F1: Shows a help screen.

Alt-M: Toggles mouse support.
Alt-V (or *Control-V*): Mutes sound and/or music.
+ and – keys: speed up/slow down combat messages.

Ability Scores

Common to all ACK adventures, players have several ability scores that determine combat ability, etc.:

Hit Points. Also called 'HP'. This is the amount of damage that can be sustained before the character is killed.
Magic Points. Also called 'MP', or 'Energy'. Used by items, spells, sometimes as fuel for vehicles, etc.
Strength. This controls how much weight the player can carry, and how fast the player regenerates hit points.
Intelligence. This controls how fast the player regenerates lost magic points, and how merchants deal with the player.

All of the above numbers max at 255, but typical player starting values are in the 10-20 range.

Weapon Skill. This shows how proficient the character is with short-ranged weapons.
Ranged Weapon Skill. Same as weapon skill, but used for ranged weapons.

These skills also max at 255, but typical player starting values are in the 25-75 range. Values up to 100 determine the % chance of hitting an unarmored creature; values over 100 allow the player to do extra damage, beyond what the weapon could normally do, when attacking creatures. (Note that strength does not affect weapon damage.)

Inventory

Some items in the character's inventory can be "readied" and will determine combat effectiveness:

Weapons: A character can have one weapon readied. (The Ready command must be used to switch weapons, including between melee weapon and ranged weapon.) Melee weapons can only attack the four squares directly adjacent to the player. Ranged weapons can target any square within the weapon's "range". A player cannot attack without a weapon.

Armor: Each armor has a "type", and the character can wear up to one armor piece of each type. There are seven types, so the player can wear up to seven pieces of armor. Armor will absorb a certain amount of damage, and has a percent chance of doing so. For example, a steel helmet may absorb a large amount of damage (20) but only 5% of shots will hit it; whereas a beekeeper suit may only absorb a tiny bit of damage (1) but all (100%) of shots will hit it.

Other items that can be in inventory include "misc. items" (these can sometimes be "Used" for effects, or are sometimes required for any number of other things) and magic spells (which can be "Cast", if the particular ACK game implements a magic system.)

Combat Mechanics

When an attack is made, it checks for the chance to hit: weapon skill up to 100 determines the chance (so 50 will hit 50% of the time, and 100+ will always hit.) If the attack hits, it calculates the damage: a random number between 1 and the weapon's damage value. For weapon skills over 100, there's a chance for a critical hit. Weapon skill (minus 100) determines the chance of double damage (so 150 minus 100 equals 50, a 50% chance of double damage; and 200+ minus 100 equals 100, a 100% chance of double damage). For weapon skills from 200 to 255 (the max) the same thing happens again, for quadruple damage. Once this damage is determined, the armor is then factored in: for any armor worn, there is a chance of it absorbing damage, and then an amount of damage it will absorb. If all the damage is absorbed, the attack becomes a miss.

HP/MP Regeneration and Poison

Hit points and magic points will regenerate at the rate of (regen rate times ability score) divided by 2000. For example, if the player's INT is 20 and the regen rate is set at 10, then magic points will regenerate at the rate of one MP every 10 turns.

If the player is poisoned, the poison strength will be spread over 10 turns: for example, a poison strength of 20 would do 2 damage per turn, while a poison strength of 1 would only do 1 damage every 10 turns. (The HP regen rate, of course, may counteract part of this.)

Weight and Encumbrance

Items can have "weight" settings. This limits what a player can carry: a player can carry 5x their Strength before they start slowing down. Strength scores of 10-20 are considered a typical "human range" by ACK, so assume characters can carry about 50-100 units of weight. (If the player is in a vehicle, however, then the vehicle's carrying capacity is used instead – this may be more, or less, than what the player could otherwise carry.) Creatures are not affected by weight, as they do not have Strength values.

Adventures Included with ACK

Adventure Creation Kit includes a few adventures to give you an idea of what can be done with the game system:



TUTORIAL: This demo adventure (rewritten for ACK 3.2) introduces you to the basic game engine concepts, explaining what you are seeing as you go. It serves as a sort of tribute to 80's gaming in general, using content from a few of the games that inspired ACK, such as Ultima V, Adventure Construction Set, and 2400AD. This adventure is the best example of what the newer version of the ACK engine is capable of doing.



BRIGANDS: An example of a small, short adventure: only a few regions and people to talk to. Mostly combat driven. (There's a walkthrough at the end of the manual.) This is a very old adventure – it's been part of ACK since version 1, with minimal changes – and does not demonstrate the full capabilities of the ACK engine.



CARAVAN: An example of a medium-sized ACK adventure, and one that is more puzzle and conversation driven. When you play, remember that most of the game is strategic: finding clues and solving puzzles. If you think with your fists, you will not solve the game. If you want to know if someone is hostile, use the "Look" command on them and it will tell you. The plot of Caravan, in short, is that you are a Roman citizen who wants to travel east. You must first find your way out of the city. There is no combat in the early stages of the game; everything must be done by talking. Talk to everyone. You will be told several times to seek someone else and ask them questions. Bear in mind that some information holds a price: the

bartender won't tell you anything valuable unless you give him what he wants. Talk to everyone; you will find all that you need. As the game continues, you will continue through Rome, Egypt, and then on to the Orient. There will be combat, but only when you are attacked. Be careful: killing the wrong people may make the game impossible to solve, and defiance to the wrong person may put you in danger! (There's a walkthrough at the end of the manual.)

Kits Included with ACK

Adventure Creation Kit includes a few kits to get you started with your game creation:



HISTORY: The History kit is similar to the Caravan adventure, and includes many of the same graphic tiles. It can also be used as a starting point for a generic fantasy adventure.



MODERN: The Modern kit has graphic tiles and objects to represent modern buildings, like houses and offices; as well as somewhat less modern buildings that you might still see today, like haunted houses.



ULTIMA: The Ultima kit has graphic tiles imported from Ultima V (with only minor alterations to coast and river tiles to simplify them somewhat, as shown at left). The intention is not to perfectly emulate Ultima (the ACK engine would be incapable of that feat anyway) but rather to provide a kit that would allow creation of a sort of playable Ultima fan-fiction.

For more information on the contents of the kits, check out the exported reports in each KIT folder:

OBJECTS.TXT – Contains a list of all the objects: terrain, weapons, etc.

CREATURE.TXT – Contains a list of all the people and creatures.

MACROS.TXT – Contains a list of all the macros included in the kit. Note the variables that are used by the existing macros, so that you don't accidentally use one that is already being used by one of the features already built into the kit.

Creating ACK Adventures



Starting a New Adventure

First, from the ACK main menu, go to "Select/Create Adventure". It will normally start you out in the "ACK\GAMES" folder. This is the best place to create your adventure. Press F3. Now give your adventure an 8-letter name. You have just created a folder for your adventure, and the basic files needed.

Now, back at the main menu, go to "Import Files, Copy Adventure". Press K to import a kit. Decide the kit you wish to use. HISTORY is good for historic adventures in the 300AD to 1500AD period in Europe, Egypt, China, or Japan. MODERN has a variety of modern-day and haunted-house graphics and objects. ULTIMA is of course the kit for creating games that look like Ultima V, and has a large number of graphic tiles and objects implemented. Choose the kit that is closest to what you want. You are not required to use a kit (you can create everything from scratch) but it is strongly recommended. While you're in there, you can also pick a different font to use. ACK includes seven different fonts, or you can copy one from an existing adventure.

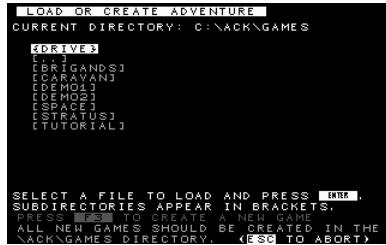
The basic flow of adventure creation is as follows:

1. Create yourself a document somewhere, where you can keep notes. Keep reminders to yourself about the general flow of your storyline. Also make notes here about what you are using variables for, if you use them (with names like A-Z and A2-Z2, it's hard to keep them straight otherwise!)
2. Create the adventure files, importing a kit if you want to.
3. Make sure you have the basic graphic tiles and objects you'll know you need – but don't go crazy with this, you may find your game idea evolve as you go, and it's easy enough to go back and make more stuff later.
4. Start making your basic maps – again, don't try to finish the job all in one pass, just get the basic structure.
5. Create the dialogues for the people that are basic to your storyline. If you need to create special objects for them (such as a key for a dialogue that gives you one) then start creating those as you need them.
6. Create creatures for those dialogues, if needed.
7. Go back and populate your map with the basic characters for whom you've made those dialogues, and add more detail to those maps – decorations, etc.
8. Think about quest-specific objects (keys, locked doors, etc.) and start placing those.
9. Start populating your maps with enemies or non-essential people. Create dialogues for some of those, if you want (look up what "small talk" dialogues are, they can be created much easier for people who have little to say)
10. As areas start looking "finished", use the config editor to start your player in specific locations with specific stuff (so you can test tricky areas of content quickly)
11. Once you start getting close to being "done", start thinking about the title and intro sequences (check out the TUTORIAL adventure for ideas here)
12. Finally, set the player starting location and equipment to what it should be at the start of the game, and playtest your adventure to make sure the quests all work, battles aren't too hard, etc.

Note that you can run multiple editor windows and press the TAB key in each one, whenever you want it to re-sync up with the others. Note also that much of your content creation may be done outside of ACK (I use Photoshop to make BMP files, and Audacity to make WAV files to convert to RAW audio files).

You should make a backup copy of your adventure folder from time to time, as ACK does not have an Undo feature. Copying the adventure is just a matter of copying the adventure folder in your operating system; it is not done inside ACK. The same applies to distribute your adventure for others to play: just copy or ZIP the folder (it's under the ACK\GAMES folder) and share it as needed. Make sure they know how to get ACK, and where to put your game folder (into ACK\GAMES). Alternately, there are instructions later in this manual on how to cut down ACK into a standalone game-launcher for your game; see Appendix C.

Editor Modules: Overview



Select/Create Adventure: Used only if you don't use the ACK Launcher. You can create new adventures and select adventures to load here.



Configure Adventure: Certain game-wide settings are set here, such as text colors. Game startup parameters, like title screens, starting location, character choices, and starting inventory are also set here. Browse through the four pages; this module is a catch-all for anything not directly manipulated in the other editors.



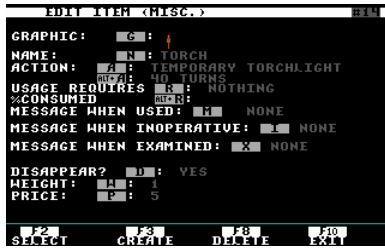
Import Files, Export Reports: Adventure Kits are imported with this utility; various features can also be copied from other adventures using this tool. The “game in progress” files can also be deleted here, which is a good thing to do to your adventure before distributing it. You can also export a full set of reports on messages, dialogue, objects, creatures, and configuration.



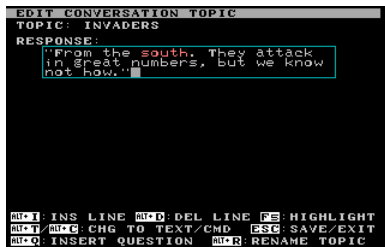
Edit Font: Allows customizing the current font. Note that ACK prompts only use uppercase; the only place lowercase will be used is in user-created messages and dialogues. This means the lowercase characters can be used for other purposes, if you wish, like runes, hieroglyphs, etc. Before creating a new font, check out the font options in the Import/Export tool; you may find something much closer to what you need.



Edit Graphic Tiles: Everything you see in ACK is represented by a graphic tile; you can edit or modify them here. You have two sets of 240 tiles (so a total of 480) with one set being the “normal” tiles (used by all terrain, items, creatures, and mosaics) and the second set being “animation” tiles used by certain terrain or creatures that toggle between multiple tiles to give an appearance of motion. There are 256 colors available for editing tiles; you can either use the palette built into ACK, or create a custom palette by creating a “PALETTE.BMP” file in your adventure folder (320x200 256 color Windows BMP format.)



Edit Objects, Items, Terrain: Everything that isn't alive is an "object". Objects fall into different categories; the category determines what features can be set. Categories are: Space, Custom Space, Obstacle, Custom Obstacle, and Portal (for terrain types); and Misc Item, Weapon, Ranged Weapon, Armor, Vehicle, and Currency (for item types.) See the section on the Object editor for more info.



Edit Messages and Dialogue: Most text content in the game can be edited here. Short messages are two-line text messages frequently used for quick messages, like "You open the box". Long messages fill the screen: they are useful for signs, books, etc. Dialogues are triggered by the "Talk" command and are interactive; they can respond to keywords (which you can highlight if you choose.) The plot of many games is driven by searching for and using these keywords; for example, person 1 tells you (after you perform some favor for him), "go ask person 2 about 'beef'." Dialogues can ask yes/no questions, terminate conversation, attack the player, give/take items, run macros, perform conditional branching, buy/sell items, and much more.



Edit Maps and Regions: A new adventure does not start with any maps. To create and edit maps, go to "Edit Maps and Regions" from the main menu. Add a new region (you can have up to 99), and choose a type (Room or Worldmap). For more information, read the section on the Map Editor.



Edit People and Creatures: New creatures can be created, and existing ones changed, in "Edit People and Creatures". Creatures have an alignment (determining who they attack; the player is considered "good"). Note that weapons and remains are separate; you can have a creature fight with a sword, but it won't drop that sword as loot unless it's also specified under "remains". In addition to dropping remains on the map, creatures can also run a macro when they die.



Edit Macros: This is a rudimentary scripting tool that can directly manipulate many ACK features, run external programs, and run complex lists of actions, with loops and conditional branches. Macros have access to certain features that aren't available any other way; this makes macros very useful for title screens and intro sequences, such as in the CARAVAN sample adventure. Macros are described in more detail later in this handbook. Note: you can dump the data from this editor to a text file called "MACROS.TXT" for reference.



CONFIGURE ADVENTURE: Setting Global Settings

This is where you decide the universal aspects of your game: What special colors does it use? What is the title screen? What are the player's ability scores? Where is the player now? etc. Always make sure that you set these before trying to play your adventure. Settings are divided across four pages:

PAGE 1: The settings on this page determine how your adventure starts.

The title sequence is a picture and long message, and/or a macro. If you use a macro, you can end it with a SHOWBMP that leaves a picture on the screen; that picture will then serve as a background for the game select/character creation menu.

The intro sequence runs only when the player creates a new character, and this happens after the player chooses the class, name, etc. Play begins immediately after the intro sequence.

Other options on this page determine the player's starting inventory, time and location, and name. The game will prompt the player for a name if the name is set to "PROMPT"; anything other than a blank value will show up in the player status, but not be changeable.

PAGE 2: If you are editing the original-copy of an adventure, then there will be three sets of ability scores. You may toggle the second and third on and off with [T]: if they are both off, then the player will not get to choose a character class. If you are editing a game in progress, there will only be one set of ability scores; these are the ability scores of the player.

PAGE 3: The level progression for the character, if you choose to use it, can be set here. The player starts at level 1, and when reaching the experience milestones for levels 2-10, the player's level is set to the new level and the corresponding macro, if any, will run. Experience is given by creatures being killed (the experience for each creature is set in the creature editor) or can be granted by macros.

PAGE 4: Some general configuration for gameplay is found on this page.

Text colors determine all the colors of text in the editors and adventure. Normally 4-6 are highlighted. The background of color 0 will determine the general background color, color of blank tiles, etc.

The "Day/Night View Sizing" option lets you globally disable the feature where the illuminated view radius changes automatically from daytime to nighttime. (Note this can be disabled region-by-region as well, in the map editor.)

Other options on this page let you set sounds and certain event macros. (If no death macro is set, gameplay ends when the player dies.)

The "extra command" allows you to create a new key command, but it must use a key that isn't already in use by the normal commands.

PAGE 5: Miscellaneous options appear on this page.

If you want to use a BMP picture as a "skin" in place of the normal in-game key list, you can select it here. You can also select whether the area above the menu is used for a clock or for an image that indicates the approximate time. (See the section on custom skins for more info.)

Other gameplay options here include whether HP and MP regenerate while resting (and whether you want to replace this with a macro); whether your adventure is password protected (disabling editors and cheatcodes).

This page also lets you change the version of an adventure. Updating an adventure to the latest ACK version will activate certain play features that would otherwise be deactivated for backwards compatibility (skins, experience/leveling, and palette colors are handled differently depending on game version.)

Note that this page gives you the option to edit a saved game. The saved game being edited is the one in slot 0 (the "quick save") slot.



IMPORT FILES, EXPORT REPORTS: Import and Export Game Data

This utility is used for copying data in from kits or other adventures and for exporting reports. It can also be used to delete the saved game. It is not used for importing custom sounds or BMP pictures; those should be placed in the game directory directly, because they are not ACK-generated content.

The options at the menu are:

IMPORT FONT: Load a new character set.

IMPORT SOME GRAPHIC TILES: Loads new graphic tiles from another ACK adventure.

IMPORT SECONDARY GRAPHIC TILES: Same as above, but with the "secondary" (also called "animation") tile set.

IMPORT RAW GRAPHIC TILES: Imports graphic tiles from certain other games, if they store their tiles in a format that ACK can decipher. Tiles can be imported from many older Origin games, including Ultima 1 (the re-release), Ultimas 3 through 5, and 2400AD. Note that if this fails, you can also take screenshots and use the Alt-B command in the Graphic Tile editor to import from BMP files. As of this writing, the ACK forums on joltcountry.com have a few threads on this subject.

IMPORT OBJECTS AND GRAPHICS: Loads objects and graphic tiles from another adventure.

IMPORT AN ADVENTURE KIT: Loads an entire adventure kit.

IMPORT 'KIT' FROM ADVENTURE: Loads objects, creatures, tiles, and messages from another adventure, as if it were a kit.

DELETE ADVENTURE: Delete the "game in progress" (saved game) to prepare your adventure to be distributed. Note that this deletes all saved game slots.

EXPORT REPORTS: Creates text reports with the contents of all short messages and long messages (to MESSAGES.TXT), dialogues (to DIALOGUES.TXT), macros and exec table entries (to MACROS.TXT), object definitions (to OBJECTS.TXT), creature definitions (to CREATURE.TXT) and game configuration (to CONFIG.TXT). Note that you can't make changes by editing these files -- you still need ACK to make any changes -- but they make it much easier to get an at-a-glance view of your adventure.



EDIT FONT: Edit the typeface used by your adventure

(Before spending too much time in the Font editor, check out the “Import Files” module and check out the selection of fonts available there. You may find something closer to what you need.) This utility is used to customize the text font used by your adventure. Operation is similar to the graphic tile editor (see below) except, since fonts only have one base color (text colors are then defined in the configuration editor) you are just turning pixels on (1-9 all act the same) and off (0). Note that there are some reserved characters used for displaying arrows, the number "10", "Pg", etc -- you can customize them, but you shouldn't change them. Note also that there are lowercase characters available. These are only used in your messages and dialogue, which you can edit for your adventure. This means that if you want to use the lowercase character set for something different (runes, heiroglyphs, etc) you can do so.



EDIT GRAPHIC TILES: Edit graphics for your terrain, items, and creatures

All things that you see during normal play - terrain, treasure, weapons, creatures, people, and even yourself - are represented by one or more graphic tiles. While you may be importing most of your tiles from another adventure or kit, it's still a good idea to make a couple of your own, to give your adventure originality. Graphic Tiles are 16x16-pixel, 256-color squares. You can have up to 480 tiles, in two sets of 240: normal and secondary. Normal tiles are used for most things: an object or creature's primary appearance, the player's icon, etc. Secondary tiles are used to animate terrain or creatures, or for spaces or obstacles that are set in “set graphic later” mode.

When you are in the graphic tile editor, you are either in Select mode (choosing a tile to edit) or Edit mode (editing a tile).



Select Mode

If the SELECT menu is present, use the arrow keys and PgUp/PgDn to move. Press [ENTER] to select a graphic tile to edit. Pressing [F10] will quit. Control-C and Control-P (or Control-V) will copy and paste tiles.



Edit Mode

From here, you can move the cursor with the arrow keys. The [1] thru [9] keys will draw in different colors, and [0] or [Space] will erase. Control-C and Control-P/V will copy and paste, just like in Select mode. You can also paste pixel-by-pixel by just hitting “P”. This will replace the current pixel in a tile with the corresponding pixel from the clipboard.

[S] (SetColors) will let you change the [1] thru [9] colors, as well as perform some color replacements. You can press [U] to find the color you were sitting on in the graphic tile, or press [R] to replace the color you were sitting on in the tile (anywhere it occurs in that tile) with the color you have picked in the palette. (Alt-R will do the same, but across *all* tiles; use this with caution!) Finally, [I] gives you an Image menu that lets you perform basic transformations: flipping, rotating, and moving.

Importing from BMP files

Pressing Alt-B while in Select Mode will replace the currently selected tile with a 16x16 piece taken from a BMP file of your choice. Just press Alt-B, and provide the path and filename to the BMP file you want to use. (It must be 320x200, 256 color, and placed somewhere ACK can see it.) Then move the selector square so that the piece you want to import is completely enclosed. Use the arrow keys to move, and press ENTER to make the capture.

If you want to import multiple tiles from the same BMP, you can just press ENTER (without typing anything) at the filename prompt the next time you use Alt-B.

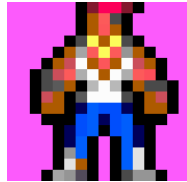
Once you import the tile, you will be placed into Edit mode where you can make any required changes. You may need to perform color replacements, depending on the palette used in the BMP file. Or, you may change the palette of your adventure by creating a PALETTE.BMP file in your adventure folder.

Note that as of this writing, the ACK forums on joltcountry.com have a few threads on the subject of importing tiles.

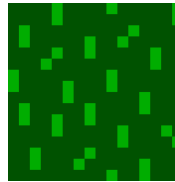
Tile Transparency

As of v3.230, ACK supports transparency for items and creatures. This is done by setting a “mask” color, which – for items and creatures only – will be rendered transparent by the engine, and any pixels of the mask color will show the underlying terrain pixels instead.

For example, if you were to use magenta as your mask, and your creature tile looks like this:



And your grass terrain looks like this:



Then during play, that creature on grass terrain would look like this:



To enable (or disable) transparency, enter the “set color” menu in tile editing mode. Select the color you want to use as your mask color (if you are enabling; if you are disabling, select any color) and then press Alt-T.



It will ask if you want to use transparency. Press “N” if you are disabling transparency, or “Y” if you are enabling transparency with the selected color. If you enable transparency, the mask color will appear in the editor with a checkerboard pattern, reminding you of its special behavior during play, as shown below:



Any desired color can be used for transparency (including black, if you find that easiest) but it is a game-wide setting and will effect all item/creature tiles. For terrain tiles, as they are never transparent, the mask color will be shown as its actual color. Note that transparency only shows during play. In the editors, you will see the mask color instead of the transparency effect.

Note also that animation does work with transparency (both animated creatures and animated terrains). However, darkened (night) tiles will not animate when transparency is enabled. Finally, the only tiles seen are the top item/creature and the top terrain. If you have a stack of items, the middle ones will not “peek through”.

These tiles will of course look better if (as shown above) you “edge” them with a border color, to separate them from the background. In the examples above, black was used as an edge color. The tile editor can auto-edge tiles: while editing the tile, press Alt-E. You will be asked which color to use for the edge color (0-9 from your pen colors) and it will then auto-draw an edge for that tile.



EDIT OBJECTS, ITEMS, TERRAIN: Manage your terrain, weapons, items, etc.

The player will spend all their time walking on two-dimensional top-view maps, which are composed completely of the objects, items, and terrain which you will create or edit here. The adventure kits include the basic objects that you will need, but you must make a few of your own if you want to have a unique game. Almost all actions and messages that you will see and do in ACK are directly related to these objects.

Objects have different features, depending on their category:

Object Categories:

SPACE: Terrain type, generally defines a passable square (grass, tile floor) but you can define it to be passable only in certain conditions (water, locked door), or to be impassable (table, counter). Has a Transparent/Opaque setting that determines if you can see on the other side of it. Space terrain, even if impassable, will allow ranged weapon shots to pass, unless set to “opaque”.

SPACE*: Custom Space; instead of having a preset action parameter, it will prompt you for one when you place the terrain on the map. Useful for things like signs, where each one will say something unique, but you don't want to have to make 10 different sign objects.

OBSTACLE: Terrain type, always impassable. (wall, mountains). Always blocks ranged weapon shots.

OBSTACLE*: Custom Obstacle; prompts for the action parameter when you place it on the map.

PORTAL: Moves the player to a different location when they step on it. Any transition (except for macros) from one map to another, or one part of a map to another (except for overlapping rooms on a room-type map) requires portals. The destination is set when you place the portal in the map editor.

MISC ITEM: An item that can be carried in a player's inventory, but is not a weapon or armor. They may be an item with an action when Used, they may be an item required elsewhere (key, ammo) or they may be trinkets (loot) that can be sold to a merchant for cash.

WEAPON: A weapon used to attack adjacent spaces.

RANGED WEAPON: A weapon used to attack non-adjacent spaces. They have the option of using ammunition, and can have different range limits and sound effects. They normally attack a single tile, but can be set to have a blast radius around the target, or to attack point-blank from the player in a stream or cone.

ARMOR: An item that can be worn to protect from damage. Each armor has a “type”, and the character can wear up to one armor piece of each type. There are seven types, so the player can wear up to seven pieces of armor. Armor will absorb a certain amount of damage, and has a percent chance of doing so.

VEHICLE: Vehicles can be used to increase your weight capacity and/or add additional weapons; they are also used to pass terrain you cannot normally pass.

SPELL/SKILL: These are inventory items that cannot be picked up or dropped, but can be used with the “cast” command.

CURRENCY: Object #1 is the “money” of the game, and appears under this category.

Object Actions:

Objects make extensive use of “Actions” that can be triggered in many ways; for terrain, it could be triggered by passing or bumping, while for items it may be triggered by using. The possible actions are:

- | | |
|---------------------------|---|
| 00. DO NOTHING | (default action – does nothing) |
| 01. DISPLAY SHORT MESSAGE | (display short message # in a pop-up) |
| 02. DISPLAY LONG MESSAGE | (display long message #) |
| 03. DISPLAY PICTURE | (display BMP # and wait for a keypress) |
| 04. PLAY SOUND EFFECT | (select sound) |

- ... CHANGE H.P., M.P. (+/- amount, temporary chg.)
- ... CHANGE HPmax, MPmax, STR, INT, SKILL (+/- amount, permanent chg.)
- 13. CHANGE ALIGNMENT (Good, Neutral, Evil, Animal)
- 14. TAKE FROM YOU ONE OF... (Remove 1 Item from inventory)
- 15. TAKE FROM USER ALL ITEM... (Remove all Item from inv.)
- 16. TAKE FROM USER ALL TYPE... (i.e. Take away all WEAPONS)
- 17. GIVE TO PERSON ONE OF... (Give 1 Item to player)
- 18. TAKE ALL THESE FROM AREA... (Remove all Item from local map)
- 19. PUT ONE OF THIS HERE (Place Item on map)
- 20. KILL USER IF NOT HOLDING... (Easy way to kill the player)
- 21. ACTIVATE ALL UNDERNEATH (Triggers a stack in RoomMaps)
- 22. HIT PLAYER WITH WEAPON (good for traps, etc)
- 23. LET PLAYER SHOOT WEAPON (weapons that you can't carry)
- 24. SUMMON CREATURE (Place creature on map)
- ... SET VARIABLE ... TO: (Sets the dummy variables)
- ... INCREASE/DECREASE VARIABLE... (Changes the dummy variables)
- 37. EXECUTE MACRO (Triggers a MACRO)
- 38. MOVE CHARACTER ONE SPACE (Acts as if player pressed an arrow key, if player does not press any other key for "DELAY" seconds. DELAY=0.0 is instant.)

- 39. SET VIEW RANGE TO... (Sets distance player can see. Gives darkness effect. Set to above 16 for normal.)

- 40. SET ICON TO: (Sets basic player icon. Overridden by Terrain, Weapon, and Vehicle icon changes.)

- 41. PROVOKE NEARBY CREATURES (Provokes on-screen creatures of specified alignment)
- 42. TEMPORARY TORCHLIGHT (Provides a temporary moderate circle of illumination)
- 43. TEMPORARY DAYLIGHT (Provides temporary daytime-level illumination)
- 44. SET HP REGEN (Sets a new HP regen rate)
- 45. SET MP REGEN (Sets a new MP regen rate)
- 46. SET POISON (Sets a new "Poison" HP decay rate)
- 47. RUN DIALOGUE (Runs the specified dialogue)

(If you want to short-cut to a given action in the action selector, enter the number)

Different object types have different configuration options, depending on their category:

Object options: Space, Obstacle, or Portal:



Visibility: determines if you can see through the terrain ("transparent") or if spaces enclosed behind it are hidden from view ("opaque"). Opaque squares also block ranged weapons. Opaque squares can still be set as passable, however.

Lightsource: determines if the square (and the squares around it) appear lit at night, even if they are out of the player's circle of illumination.

Graphic: Set the appearance. If you select an animation tile (Alt-G), it will animate between the two. The "Icon" graphic, for spaces, is what a player/creature will look like if they are on the space. Note also that on non-custom spaces and obstacles, you can choose a graphic of "set later". This disables animation, but allows you to choose a unique graphic tile each time you place the object on the map (note that the graphic tile is chosen from the secondary set)

Action: Performs an Action; see the "Actions" table above.

Triggered By: Determines how the action is triggered:

- PASSING: Walking over the spot.
- DROPPING ITEM HERE: Dropping specified item on this spot.
- PASSING, HOLDING: Walking over the spot, holding or riding ITEM.
- PASSING, REDEEMING: Same as HOLDING, but item is lost.
- PASSING, WITHOUT: Walking over the spot WITHOUT Item.
- EXAMINING, FAR: [L]ooking at the spot from anywhere on the screen.
- BUMPING: Trying to walk into the spot, and being blocked.
- BUMPING, HOLDING

BUMPING, REDEEMING

BUMPING, WITHOUT

TAKING ITEM FROM HERE: Taking specified (or any) item from this spot.

EXAMINING, NEAR: [L]ooking at the spot from an adjacent spot.

GATHERING FROM HERE: Trying to [G]et when there is no ITEM.

ATTACKING: Attacking this spot with any weapon (or with a specified weapon).

Passable To: For spaces and portals, determines what is required to walk onto the space:

NOTHING: Nothing can pass (except ranged weapon attacks if the visibility is 'transparent')

ANYTHING: Anything can pass (except vehicles set to require certain conditions)

ONLY IF HOLDING... : Only if holding or riding Object.

ONLY IF REDEEMING... : Same as HOLDING, but item is lost.

ONLY IF W/OUT ITEM... : Only if NOT holding (or riding) item.

ONLY IF W/OUT TYPE... : i.e. Only if NOT holding any WEAPON.

IF STRENGTH > N ... : Only if strength exceeds given value.

ENERGY DRAIN ... : Only if sufficient ENERGY is given.

UNBURDENED ONLY : Only if player is not overburdened with weight.

IF VARIABLE A > N, etc: Only if specified equation is true.

ANYTHING AT ALL: Anything can pass, including all vehicles, regardless of settings.

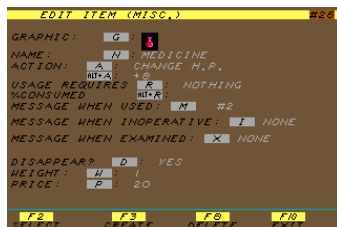
Message When Blocked, Examined: Short messages displayed in those situations.

Disappear: Determines what will cause the terrain to disappear, if anything. The special case is "While impassable" – this does not actually remove the terrain, but makes it appear as its "replaced" terrain whenever it is currently impassable. This can make locked doors appear locked without having to perform otherwise complex scripting. The "use macro" option is the same as "if destroyed", but allows a macro to run when this happens.

Structural Strength: determines the damage required to destroy the terrain; N/A can not be destroyed by weapons. If you set a value here, you can optionally set the "Disappear" option to "use macro" to give the object a macro to run when destroyed.

Replaced: If the terrain is destroyed, or disappears by other means, it will be replaced by the object of your choice.

Object options: Misc Items, Spell/Skills:



Graphic: For Misc Items, sets the appearance. (Spell/Skills have no graphic.)

Action: Action performed by using the item or casting the spell, if any.

2nd Action: Spells can also perform a 2nd action. This action can be suppressed by having the primary action call a macro, and having the macro setting SUCCESS to 0.

Usage Requires: If using the item or spell requires another object, set it here, and set the chances of consuming it.

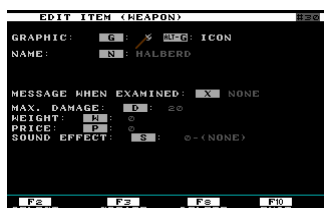
Message: Short message to be displayed when the object is used, inoperative (required object missing) or when the item is examined.

Disappear: Sets whether the item is consumed on use.

Weight, Price: Determines how heavy the item is and how much it would cost from a merchant.

Magic Points Used: For spells, determines how much MP to consume.

Object options: Weapon, Ranged Weapon



Graphic: Set the appearance. Alt-graphic determines the player's appearance when the weapon is readied.

Ammo Required: If using the weapon requires another item, set it here, and set the chances of consuming it.

Messages: Short messages displayed when the player has no required ammo, or when the weapon is examined.

Max Damage: The normal maximum damage of the weapon. Remember that weapon skills over 100 have a chance for extra damage.

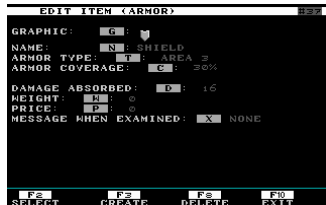
Range: Determines how far (in squares) the weapon can be targeted. You can also set whether it is a pinpoint attack (targeting a single square), or if it has a blast radius from that targeted square, or if it is a point-blank weapon that fires in a single direction in a stream or cone.

Sound Effect: VOC sound played when the weapon is fired.

Icon: Visual representation of the weapon's attack; can be dots or lines, or can be tiles from the graphic tile editor.

Weight, Price: Determines how heavy the item is and how much it would cost from a merchant.

Object options: Armor



Graphic: Set the appearance. Alt-graphic determines the player's appearance when the armor is worn.

Armor Type: There are 7 types, and only one of each type may be worn at a time.

Armor Coverage: Each incoming shot will be checked against each piece of armor; this is the percent chance that the armor will absorb damage.

Damage Absorbed: The amount of damage subtracted from the incoming attack.

Message: Short message displayed when the armor is examined.

Weight, Price: Determines how heavy the item is and how much it would cost from a merchant.

Object options: Vehicles



Graphic: Set the appearance. Alt-graphic determines the player's appearance when using the vehicle.

Built-in Weapon: Some vehicles, like ships, may include cannons the player can fire in addition to their regular attacks.

Counts As: If you require a certain vehicle to pass a terrain type (like water may require Ship) this allows you to have other types of boats (like Sloop and Brig and Frigate), but have them counted as the basic type of "Ship" when the terrain checks.

Can Go: For ships, you don't want them cruising on land, so you would set them to go only on terrain that requires them, like water.

Fuel Required: If traveling in the vehicle requires an item as fuel, set it here, and set the chances of consuming it on each step.

Messages: Sets the short messages shown when the vehicle is out of fuel, boarded, or examined.

Weight Capacity: Using a vehicle may increase the amount of weight the player can carry.

Macros: You can set macros to run when the player boards and exits. You can also have the boarding macro prevent boarding, if needed, by having it set SUCCESS to 0.

Object options: Currency

Only the graphic, appearance, and weight can be set for currency (of which there can only be one object, #1.)

Map Colors

All objects have a "map color" (shown in the upper-left corner of the object properties window) that can be set with Alt-M, and will determine the color used to represent that object on a mini-map (such as in the map editor, or with the SHOWMAP macro command).

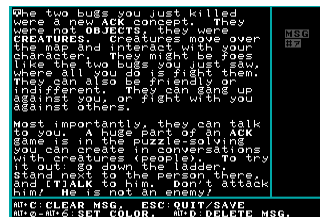


EDIT MESSAGES AND DIALOGUE: Manage text content for your adventure

This editor provides a central place to export (to a text file for review) or manage the "short messages" (2 line text messages used frequently through the adventure for responses like "you need a key to open this door) and the "long messages" (22 line messages used for books, plot-advancing text, etc.) You can also export and edit dialogues for creatures/NPCs here.

Mosaics can also be edited here. Mosaics are arrangements of graphic tiles that can be displayed by a macro. They are good for making cutscenes and having them look consistent with the regular game graphics.

Short/Long Message Editors



Short messages are used all over an adventure, mostly by objects: for example, terrain can have a short message used when passage is blocked, or when the object is examined.

Long messages are used in the introduction and title sequences, or as needed by objects and macros: for example, a road sign might use a long message to show the player a large amount of text.

Editing works the same for each: use the arrow keys to move the cursor. Alt-0 thru Alt-6 will change the text color (colors are defined in the configuration editor). Alt-C will clear the message, and Alt-D will delete the message. Press ESC to quit and save. You can have up to 255 of each message type.

Mosaic Editor



Mosaics can only be used in macros. You can have up to 255 mosaics. They can be displayed normally (MOSAIC x) to fill the screen by themselves, or transparent, to overlay them onto something else like a long message. (TMOSAIC x). The introduction sequence in the CARAVAN adventure was done with macros placing transparent mosaics onto long messages. Draw with the [A] thru [G] keys. Configure each key with [Alt-A] thru [Alt-G]. Erase with the [SpaceBar] or [Del].

You can use the [PgUp] and [PgDn] keys to flip through the separate mosaics. [Control-PgUp] and [Control-PgDn] will copy the current mosaic to the previous or next mosaic; if you are making an animation sequence, this makes it easy to copy the current mosaic forward.

Dialogue Editor



When you enter the dialogue editor, you start out editing the list of topics for a dialogue. Topics determine how the speaker reacts to the player. If there is a topic called "NAME", then asking "NAME" during conversation will run that topic. There are a few preset topics: "(INTRO)" is run at the beginning of conversation, the first time the player runs it. If it exists, "(HELLO)" is used instead of (INTRO) on future meetings. "(BYE)" is run at the end, unless there is a TERMINATE command in a topic. "(?)" is run when the player gives an unknown query. You can also have topics that can be called by other topics; *1 through *255 can be called by the GOTO command.

You may have up to 255 dialogues. Each may have up to 40 topics, each being as long as you like (as long as the entire dialogue is no more than 1000 lines.) Topics can scroll, although you will want to add WAITKEY commands to longer responses to control where it pauses (it will not do so automatically).

While this is not controlled in the editor, dialogues can also have "skins" -- BMP files that are used as backgrounds for the conversation. This allows you to have portraits for the characters the player meets. (Refer to the BRIGANDS demo adventure for an example.) Dialogue skin files are named TALKx.BMP, where x is the dialogue number. Dialogues can also manually call other skins (and they can be numbered up to 9999) if you want even more control over the portrait (to show expressions, or to simulate having other characters interrupting or joining the conversation.) You don't need to match the game palette perfectly in your dialogue skins, but at least the text colors and the background color need to match. So if your text is white (#15) on black (#0) then colors 0 and 15 in the BMP color palette need to be black and white, or things will look funny. If in doubt: just make sure your dialogue skins use the same palette.

Dialogue Topics

You can insert or delete lines with [Alt-I] and [Alt-D]. You can highlight keywords in a topic (to hint to the player that they could ask about that word to continue the thread) by pressing F5.

To insert a yes/no question in the topic, press [Alt-Q]. The question must be on or before the "?" line. The "Y" line begins the "YES" response, and the "N" line begins the "NO" response. (Questions can also be used with the ITEM TAKE command – see the section below.)

Dialogue Commands

Dialogue responses may also have "commands", which can trigger certain events in the conversation. To change a line into a command line (or to change the command setting) press [Alt-C]. To change a command into text, press [Alt-T].

MACRO: Runs a macro, then returns to the flow of the topic.

MERCHANT: Allows the player to purchase and sell items. A merchant can have up to 16 items to sell; press [A] thru [P] to set these. You can also change the exchange rates for how they sell things to you and how they buy things (refund) from you. Regardless of the exchange rates, a merchant will not buy back his own wares for a higher price than he sold them for. If he sells you a sword for 10, he will never give more than 10 for it, even if his exchange rates would ordinarily have him give you 20 for it. The prices on items are set from the Object Editor. They are also modified by the player's INT ability score; be sure to playtest your adventure, to make sure that prices are fair.

SKIN: Loads a different dialogue skin BMP (or, if you use SKIN 0, loads the original one).

TERMINATE: Ends the conversation.

END TOPIC: Ends the topic response.

ATTACK: Ends the conversation, and begins to attack a specified alignment (GOOD being the player)

ITEMGIVE: Gives a specified item to the player.

ITEMTAKE: Tries to take an item from the player; if followed by a yes/no question, this will be automatically answered by the outcome of the attempt (i.e. if the player does not have the item, the "no" response will be used)

GOTO*: Will jump to another topic, specified by variable or number. The new topic will have a name from "*0" to "*255".

SETV.: Will set a variable (A-Z) to a value (0-255).

BEGINQUEST: Makes the person start using topic (BEGINQ) instead of (INTRO), if it exists, when greeting the player.

ENDQUEST: Makes the person start using topic (ENDQ) instead of (INTRO), if it exists, when greeting the player.

Using Variables and Macros in Dialogues

You can use macros in dialogues with the MACRO command. To have the dialogue flow determined by a macro, use the GOTO* command with a variable. The variables can be A thru Z. When making topics, the topics can be *0 thru *255, and a *? topic that is used when the correct special topic is not found.

Use the "Export MESSAGES.TXT" feature at the main menu of the message editor to export your dialogue to a text file for troubleshooting.

How to use Dialogues:



Since ACK is designed for adventures in clue-finding and problem-solving, the player's interactions with other people and creatures can't be limited just to fights; there has to be peaceful interaction as well, where the player can converse with others and find information. Any person or creature on the map can have a "dialogue" assigned to them, which describes how they will respond to queries and what they will say and do.

The basic idea of a dialogue is simple; you type in one-word queries, or "topics". If the person has something to say about that topic, they will. Some standard topics for everyone might be "name" and "job". If the player types "job", the person might respond "I am a guard." You might also make a topic for that person called "guard" so the player can ask more detail by typing "guard". This can continue, with one topic leading to another, or leading the player to ask some other person about a topic. This clue-hunting can be a large part of your adventure.

Beyond this, however, dialogues can do more than just display text. Dialogues can ask the player questions, they can offer to sell the player items, they can prematurely halt the conversation, give items, set variables, trigger attacks, jump to topics automatically, give random responses, and run macros.

Some of the dialogue commands are self-explanatory. MERCHANT will give you a menu where you can set what items the

person will sell. (Prices are set in the object editor) All merchants will also offer to buy back items at a lower rate. VAR-MENU is similar; with a menu, you give settings like "Set B to 7 for \$15." TERMINATE can end a conversation immediately; maybe you had offended the person. ATTACK can be used to stop conversing and attack the player (if you said something really bad). GIVEITEM will simply give the player an item for free -- just one item, with no menu or price. SETV will simply set a variable -- no menu, no price.

The various branching options, however, aren't quite as straight-forward. GOTO* will jump to another topic. The topics must be named in a special way. They must be named with a star and a number, anything from *0 to *255. The GOTO* command, then, can jump to a specific topic (GOTO*7) or to one determined by a variable (GOTO*B would jump to topic *2 if B=2, or topic *1 if B=1, etc.) With this technique, a person could remember talking to you before by setting and using a variable. Or it could appear as if other people are talking about you to each other: one person could do SETV:B=4 and another person could have GOTO*B.

The final option, MACRO, is rather straightforward; it simply runs a macro. You could have a macro set a variable to a random number, and then use GOTO* to jump to that random setting; this would give truly random responses. Or you could have a macro do things that the other commands can't; a trainer might have a macro that increases your weapon skill and charges you some money. Someone might demand a certain item from the player, and use MACRO and GOTO* to determine if they have it and respond accordingly. A travel agent might even have a macro that changes the player's map location entirely. (Note: the QA macro command won't work from dialogue macros.)

“Small Talk” Dialogues

Sometimes you want NPCs that just respond by making “small talk”, random lines like “Some weather we're having!” that aren't plot-critical, but allow you to have extra people running adding to the atmosphere without being mute or requiring the tedious work of creating dialogues and macros to generate random quick responses. To make a “small talk” dialogue, just create a text file in the adventure folder: TALKxxx.TXT, where “xxx” is the dialogue number. For example, to make dialogue #201 into “small talk”, create a text file called TALK201.TXT, and put as many lines into the file as you like:

```
Hey, how's it going?  
Nice weather we're having?  
Stop me if you've heard this one,/a troll and an orc go into a bar...  
What are you looking at?
```

Note the slash in the third line: this splits the response into two lines (the maximum.) Each time you talk to an NPC with this dialogue, it will pick a random line from the text file, and use it as a quick response. The response will be displayed in the bottom area of the regular screen; the screen does not have to clear for small talk dialogues.

Small talks override regular dialogues; if there were a dialogue 201 already, it would be ignored in favor of the small talk file.

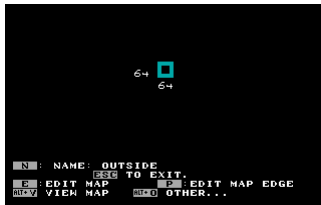


EDIT MAPS AND REGIONS: The Map Editor

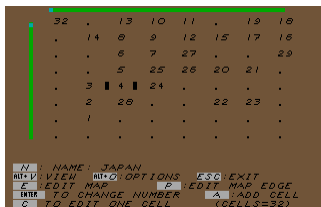
Most of your time in making an adventure will be spent here. When you have created the objects and creatures you want to place on the map, this is where you will actually draw the map and place the objects. You may have up to 99 maps. There are three types of maps used in ACK:



Room-type: Up to 16 separate “rooms” where the entire room is shown on the screen, and does not scroll. If room edges overlap, the player can walk directly from one to the next. Rooms can be up to 16x11. These maps allow unrestricted stacking of objects, which is one option for triggering complex events. (For more info, look in the Tips and Tricks section.) Ideally, these rooms are well suited for towns or dungeons where transitioning from one discrete space to another is appropriate.



Worldmap-type: A single large grid, normally up to 240x240, where an 11x11 viewport is shown on the screen, and scrolls as you move. These maps do not allow stacking of terrain onto terrain (though you can place items onto terrain) so they don't have the same automation abilities that room maps have. However, by giving an expansive, continuous surface, they give a much more “outdoors” feel. (Note: you can use 256x256, but it will have a notch missing from one corner). You can define how map edges are handled: whether they are an endless field of a certain type of terrain, a portal to somewhere else, or a wrap-around of the map.



Custom Worldmap: A worldmap mode where the technical workings are more exposed to the user. All worldmaps are composed of 16x16 “chunks” that are loaded in and out of memory as needed. Normally these are cleanly arranged in a simple grid, but a custom worldmap lets you do layouts where the map is an odd shape (such as a giant L or T shape) or where certain content (like open ocean) is repeated in order to make a map up to 512x512. Use this map type if you need to cheat and create a map larger than the standard worldmap type allows.

You can convert back and forth between custom and regular worldmap at any time by pressing Alt-C (to convert to custom) or Alt-W (to convert back to normal worldmap).

From the main menu you can add or delete regions, edit them, or select whether you are editing the default maps or the maps from the currently saved game. If you edit a map, you will enter either the Room Map Editor or the World Map Editor.

Both editors have an [Alt-O] “Other map options” page that includes region configuration, such as the option to disable or replace the “step macro” (which, if set in the config editor, would run on each step), the “step time” (controlling how fast the game clock gets incremented) and a “visibility” option which can set how the region handles darkness or view obstructions, disabling nighttime darkness for illuminated indoor regions, or creating darkness for dark indoor regions like dungeons. Macros that run when the player enters and exits the region can also be set here.

Room Layout Editor

When you first enter the Room Editor, you will be looking at a plan view showing the layout of all the rooms. Before creating any rooms, this will be blank. You have the option to name the map, add or delete rooms, and edit rooms. When you add rooms, you will lay out the upper-left and lower-right corners. You can have a total of 16 rooms. If you want the player to be able to walk from one room to another, you can lay them out with overlapping edges, and the edge squares will be shared; if those squares are passable the player can walk from one room to the next. Otherwise, you can use portals. Once you lay out some empty room shells, you will want to edit them.

Room Editor



In the room editor, you can name and draw the room. The A-D keys will paint objects onto the room at the cursor location. (Alt-A through Alt-D set the objects used.) E will erase, and L will place creatures. The arrow keys will move the cursor around the room.

In room maps, everything can be stacked, even terrain. If an object with an “ACTIVATE ALL UNDERNEATH” action is placed on top, it can trigger all terrain actions in the stack.

Worldmap Layout Editor

When you first enter the Worldmap Editor, you will be looking at a diagram showing the size of the map, and a list of options where you can name the map, edit it, or set options like map edge settings. Map edges can be endless terrain, where you can set the type -- if a space, you can walk out to infinity, and if a portal, you can have the entire edge be a portal to somewhere else. Or map edges can be wrap-around. Note that you do this, you may want to cap the north and south edges of the map with some impassable terrain like mountains, or you will have a "paradox donut" map where you can walk north forever. Or you may want this; the Ultima games, for example, seemed to love this model. Unfortunately you can't use a different portal for each direction and create one huge supermap; to do that, you will have to have constrictions in the map where it's feasible to place individual portals manually.

When you press 'E' to edit the map, you will select the part of the map to start with, and you will enter the Worldmap Editor.

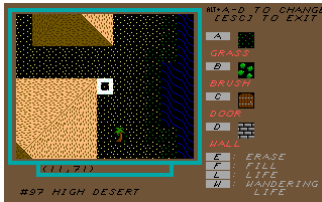
Custom Worldmap Layout Editor

If you are entering a custom worldmap instead of a regular worldmap, you start out as a 16x16 worldmap and you can add additional 16x16 cells, or "chunks", to create one large worldmap of an odd shape. To place cells on the map, use the "A" key to add more cells, then use the arrow keys and the Enter key to place cell numbers on the spreadsheet. Note that you can repeat cell numbers if you want, but be careful doing so: if the player were to drop an item (or you were to place an item) on one of these repeating cells, it becomes part of the cell and would be cloned onto the other cells of the same number.

A better idea for repeating terrain on custom worldmaps is to leave the cells undefined, and set the options for the map edges to "endless terrain". With this trick, you could make a custom map that just had a few cells to edit, but appeared like an infinitely large map with vast empty oceans and a couple of tiny islands. This avoids the item dropping problem; items dropped onto off-map locations simply disappear.

You can convert the map back to a normal worldmap with Alt-W before editing the terrain, or you can edit directly by choosing a cell location and then pressing 'E'.

Worldmap Editor



Just like the room editor, you paint objects with the A-D keys, set the objects to paint with Alt-A through Alt-D, erase with E, and use L to place creatures. There is an extra command, Fill, which will paint one of the A-D terrain types over the whole viewport. (Regular "Fill" will just replace any visible squares with the terrain you are currently positioned over. Alt-F will fill every visible square in the viewport.)

Unlike editing a room, the display scrolls. The size of the viewport matches what the player sees during play, but you have map coordinates shown at the bottom.

There is also an extra "Wandering Life" feature where you can paint a creature that has a random chance to appear in that spot. The determination is made when the current 16x16 map chunk is loaded into memory, which means that it only happens when the player enters the map or when the player walks a significant distance away and then comes back. They will not spontaneously appear right next to the player.

Note that unlike room maps, terrain cannot be stacked and the "ACTIVATE ALL UNDERNEATH" is not meaningful. You can, however, place items and vehicles onto world maps.

Placing Portals

No matter what kind of map you are using, sometimes you will need to place portals. A portal is like a space, but stepping onto it will teleport the player to another location. When you place the portal, you will be asked to select the region to go to, the room or the part of the worldmap, and then the location. Note that this does not create a return portal. Portals are one-way, and if you want a two-way path, you will have to place a corresponding portal at the other end later.

If you later change your mind about the destination, erase the portal and create a new one.

Placing Creatures

When you place a creature (or person), you will be prompted for its mobility level (Stop will be immobile, no matter what happens to it; Move will move randomly; and Wait will stay in one spot unless attacked, at which point it will move freely.) You will then be asked if you want the creature to have a dialogue (be able to talk if spoken to) and if so, which one. You can use an existing one or a blank one, and then you can then edit the conversation. (Refer to the instructions for editing dialogues in the Message Editor section.) If you later change your mind about the mobility or dialogue, erase the creature and create a new one.



EDIT PEOPLE AND CREATURES: Defining your NPCs (Non-Player Characters)

All NPCs in the game, whether good, evil, or indifferent, are created in this section. You may have up to 128 creature definitions. The adventure kits will include a number of creatures already made; you do not have to create more if you do not wish to. Note that you can export a list of all your creatures in the "Export Reports" module.



Alignment

GOOD creatures will attack EVIL creatures on sight. NEUTRAL creatures will ignore others, unless attacked. EVIL creatures will attack GOOD creatures on sight. ANIMAL creatures will attack everything except their own kind. The player is of GOOD alignment, but treated separately. If the player attacks a NEUTRAL, that creature will attack only the player, not other GOOD creatures. If the player attacks one of his/her own alignment, the player temporarily becomes a "criminal" and ALL creatures of that alignment will attack the player, until the player goes to another region.

Invulnerable creatures (if you set their max HP to 0, they are treated as invulnerable) are handled somewhat differently; their alignment determines who they will attack, but attacking them will not make them "angry". For example, if you had a plot-essential "Good" NPC you didn't want anyone to be able to kill, you could make them invulnerable; if the player tried attacking them, nothing would happen, and the attack would be ignored. If the player attacked another "Good" NPC (putting the player in a temporary "criminal" mode) then the invulnerable NPC would attack the player. In other words, be careful about mixing invulnerable and normal creatures.

Differences between Creatures and Players

Creature ability scores are simplified from player ability scores. Creatures do not use STRENGTH, INTELLIGENCE, or Magic Points (ENERGY). Creatures do not require ammunition for their weapons. Creatures are not affected by ACTIONS. They may be affected by some MACROS, and by DIALOGUES, however. Creatures cannot use PORTALS -- they are stuck in the room or the map they are in. They also cannot pass many types of terrain: only PASSABLE BY ANYTHING and PASSABLE IF HOLDING (with Vehicle) will allow a creature to pass.

Creature Equipment

These items determine how the creature attacks and moves, but the names of these items are never revealed to the player, and the items are not dropped as loot for the player.

WEAPONS: The first weapon is a short-range weapon, the second is a ranged weapon. The creature will prefer the weapon that does more damage. If the creature has no weapons and is put into combat, it will try to run away.

ARMOR: The armor that protects the creature. Only one piece may be used. Note: Since the names of the items are never given to the player, you could use normal weapons for animals.

VEHICLE: This setting determines how the creature can move. It can be set to a terrain (confining the creature to that terrain) or it can be set to an item or vehicle (allowing the creature to walk on "passable if holding" terrain, or forbidding passage if the vehicle settings forbid it). You can also press 'F' here to make the creature "fly" (pass over any SPACE terrain)

Creature Remains

This is a stack of items that can be added to the map (for items, you specify a % chance of this happening) when the creature is killed. Terrain objects can also be used, if you want permanent remains (but be careful with this.) These items are not usable to the creature.

Macros and Experience

The "Death" macro is run when the creature is killed in combat. For v3.0 and higher adventures, the "Touch" macro runs when the creature scores a melee hit on the player, and setting an EXP value will grant the player the amount of experience whenever one of these creatures is killed. Note that it doesn't matter who kills it -- NPCs killing each other will still give the exp to the player, if exp is set. You may therefore want to set "good" creatures to give zero experience.



EDIT MACROS: The ACK Scripting Language

Overview

The Macro editor is a crude scripting language, similar to the BASIC language as it was usually implemented in the '80s. The ACK Macro language uses line numbers, GOTO's, IF-THEN statements, simple variables, and several specialized commands. The CARAVAN and BRIGANDS demo adventures both make use of macros. You are encouraged to view and experiment with these macros – use the "EXPORT REPORTS" module to dump the macros to a text file for easy reading, and then look around the editors to see where the different macros are used.

Here would be the ACK Macro equivalent of a FOR-NEXT loop in BASIC:

ACK Macro:	And in BASIC:
1: SET Z=1	1 FOR Z = 1 TO 10
2:	2
3: (your commands here)	3 (your commands here)
4:	4
5: SET Z=Z+1	5 NEXT Z
6: IF Z>10 THEN 8	
7: GOTO 1	
8: STOP	

Examples of ACK macros:

To generate a random number for use in a dialogue: The dialogue could first do SETV:L=2 and SETV:U=7. This would set a lower and an upper value. Running this macro would then set variable R to a value in the 2-7 range. You could then do GOTO*R to jump to a topic from *2 to *7. This macro is very flexible and can be used many times in many dialogues because L and U can be set to whatever you want.

```
1: SET U=U-L
2: SET U=U+1
3: SET R=RND(U)
4: SET R=R+L
```

To make a weapons trainer: Have a dialogue trigger this macro. It will increase your weapon skill and set R to 1 if the player had enough money. Otherwise it will set R to 0. With a GOTO*R, the dialogue can then give a response describing the training, or a response that tells the player that he can't afford the service. (Note that the cost is 50, so if the player has more than 49, he must have the 50. Also be careful with the first line; IF-THENS and GOTOs get renumbered automatically as you insert lines. Try inserting all 7 blank lines before typing the macro.

```
1: IF MONEY>49 THEN 4
2: SET R=0
3: STOP
4: SET R=1
5: SET MONEY=MONEY-50
6: SET WSK=WSK+20
7: STOP
```

To display a title screen: If you created Mosaic 12 as a scene to go with Long Message 5, you could use this macro:

```
1: LMSG 5
2: TMOSAIC 12
3: WAITKEY
4: STOP
```

To display a BMP picture file: There are two commands that will do this. PICTURE handles everything automatically and behaves just like the "show picture" action – displays the picture, waits for a keypress, and then fixes the color palette. For more complex customization, you may want more control; you would then use the SHOWBMP command, which simply puts a BMP on the screen and continues. If the BMP uses a different palette than ACK, you must use a FIXPAL command before the end of the macro to set it back to normal.

This example simply shows picture 1, waits for a keypress, shows picture 2, and waits for a keypress:

```
1: PICTURE 1
2: PICTURE 2
```

This example fades the screen to black, shows picture 1, shows picture 2 a second later, then displays mosaic 5 over it for a second, and finally restores the screen back to the way it was before.

```
1: SAVESCR
2: FADEOUT 10
3: SHOWBMP 1
4: DELAY 10
5: SHOWBMP 2
6: TMOSAIC 5
7: DELAY 10
8: CLRSCR
9: FIXPAL
10: LOADSCR
```

To make an animated cutscene: You can show short movie-like scenes by displaying a bunch of mosaics in sequence. First draw the mosaics in the "Edit Messages and Dialogue" section. Then make this macro:

```
1: SET F=1
2: MOSAIC F
3: DELAY 5
4: SET F=F+1
5: IF F=15 THEN 7
6: GOTO 2
7: STOP
```

For more examples, browse through the CARAVAN adventure; that game makes extensive use of macros.

How to use the Macro Editor:

Press ALT-I to insert a line, ALT-D will delete a line. Note that the line numbers of GOTO and IF-THEN statements will be renumbered for you (although ESCKEY settings will not.) Press DEL to delete the current line, or Control-C, Control-X, and Control-V to copy, cut, or paste the current line. Press ENTER to edit the current line. Use the Up/Dn arrow keys to move the pointer. (The display will scroll) ESC will exit the editor, saving the current macro.

Macro Variables:

General-purpose:

A through Z: general-purpose variables which may be used however you like. The A thru D variables can be used by objects/terrain as well. A-G can hold values of 0-255, and H-Z can hold values 0-65534.

A2 through Z2: like A through Z, but a second set. The same value restrictions apply. Can only be used in macros.

RND(x): returns a random number from 0 to x.

KEYPRESS: returns an ASCII code if a key has been pressed, or 0 if no key was pressed. (Rarely used; there are several commands normally used to handle user input)

ESCKEY: setting this variable will make the "PAUSEK" command act as a GOTO if the ESC key is hit. Very useful for cutscenes or intro sequences where you want the player to be able to skip through.

SUCCESS: setting this variable to '0' will cause a spell's 2nd action not to cast (if used in the spell's first action), or will prevent the player from boarding a vehicle (if used in the vehicle's boarding macro), or will prevent a portal from transporting the player (if used by the portal in a macro, run by an action, triggered by passing)

PORTAL: Gives the portal number of the portal that was just stepped on; can be used by the SETPORT command (for example, the command *SETPORT PORTAL ###*) to perform a last-minute destination change for a portal, if the portal has a macro that runs when it is stepped on (via an action).

WMUSIC, CMUSIC: The walking and combat music settings for the current region.

Player stats:

HP, HPMAX, MP, MPMAX, STR, INT, WSK, and RWSK: Player ability scores.

HREGEN, MREGEN, POISON: Player ability score modifiers.

ALIGN, CRIM: Variables that determine what creature alignments will attach the player.

LEVEL, EXP: Variables for the player's level and experience. Setting the level (in a macro) will set the experience accordingly based on the table in the config editor.

MONEY: Amount of money the player is carrying.

INV[n]: Amount of that object# the player is carrying in inventory. (Check the numbers in the Object Editor.)

WORN[n]: Returns 1 if the player is wearing object #n (it must be armor), returns 0 if not. If you set this variable, it will cause the object to be worn (if it's armor) and will add it to the player's inventory if needed.

WEAPON: Returns the object number of the currently readied weapon, returns 0 if no weapon is readied. If you set this variable, it will add the object to the player's inventory if needed.

VEH: Player's vehicle.

LOC[...]: Player's location. Use LOC[RG] (region), LOC[RM] (room, if applicable), LOC[X], LOC[Y] (X,Y location). Setting these variables will teleport the player.

Other game variables:

VIEW: The current un-lit view-radius. Is automatically changed by time of day (if enabled) or by entering a region.

TORCHL: The current torchlight view-radius.

TORCHD: The duration of the current torchlight view-radius.

ICON: The graphic tile currently representing the player.

DAYS, HOUR, MINUTE: Time in the game world, in 24-hour time.

AGGRO: The number of visible creatures currently hostile to the player. (Read-only)

FLY: If set to 1, the player will go into "fly mode". In fly mode: only ranged weapons work; the player can travel over anything (even obstacles and other creatures); portals are ignored; any walking or bumping action triggers are ignored; and "opaque" settings on terrain are ignored. Setting FLY to 0 will turn the mode off. Vehicles can use board/exit macros that turn FLY on and off, if you want to simulate flying vehicles. Note that room-type regions ignore fly mode.

INVISDUR: If set to a non-zero value 1-254, will determine for how many moves the player will go into "invis mode". In invis mode, creatures will not attack the player. Attacking, talking, or boarding/exiting a vehicle will break invis mode immediately. Note that you should set INVISICO first (see below.) Also note that setting INVISDUR to 0 will turn invis mode off, and setting it to 255 will make it last until the player breaks it by one of the actions listed above.

INVISICO: Sets the graphic tile to be used for the player when they are invisible. Note that this is overridden by any other temporary condition that would affect the player's appearance, like terrain, vehicle, weapon, etc.

LASTMOV: Returns the last direction the player tried to move. (values 1-4 representing N/S/E/W). Useful if you have something like a portal that would behave differently if you entered from different directions.

HELLO[d], where d is a dialogue number. Allows you to set or check the "hello" setting on that dialogue, which determines how the conversation starts. Typical values are 0 for (INTRO), for someone the player has never spoken to; 1 for (HELLO), for someone the player has already spoken to; and 2 or 3 for (BEGINQ) or (ENDQ), typically set by BEGINQUEST and ENDQUEST dialogue commands. Higher values will load a dialogue into topics (H4), (H5), etc.

Numbers and variables can be used interchangeably, but attempts to set read-only items (such as SET 5=7) are ignored. The editor compiles each line as you enter it, so if you enter a unknown command or a syntax that the compiler can't figure out, it will delete the line.

Macro Commands:

ADVTIME: "ADVTIME 45" - Advances the game time by a number of minutes. After the macro completes, the hourly macro (if used) will run however many times it needs in order to catch up.

AMBIENT: "AMBIENT 60" - Plays a sound on the "ambient" channel – a looping sound channel that can be used for ambient noises. Sounds on this channel do not mix; playing another sound will stop any ambient sound already in play. Play sound 0 to stop it.

BMSG: "BMSG 7" - Shows a Short Message, but in the bottom section of the screen. Does not wait for a keypress.

CLRSCR: "CLRSCR" - Clears the screen.

ENDGAME: "ENDGAME" - Halts the program.

EXEC: "EXEC 4" - Executes specified external program, from the Exec Table. See the later section on external programs.

FADEOUT: "FADEOUT 50" - Fades the screen to black, at the specified speed. Can create a nice dramatic effect. You can then display whatever you want on the screen, but it will be invisible until you make it appear with FADEIN.

FADEIN: "FADEIN 50" - Fades a black screen back in. Use this command after FADEOUT.

GETADJ: "GETADJ X Y 12" - Prompts the player for a direction, and returns the X,Y coordinates of the adjacent space in that direction (for use with commands like MAPCHK). If the player hits ESC, then the macro will GOTO the line number specified by the 3rd argument (or will terminate the macro, if that number is 0).

GETKEY: "GETKEY B" - Gets a keypress, and stores the ASCII value of the key (uppercase) in a variable. Arrow keys (up/down/left/right) return 128 through 131. If the key is any other extended key, such as F5 or PgUp, then it will return 0.

GOTO: "GOTO 8" or "GOTO A" - Jumps to a new line number in the macro.

IF: "IF A=7 THEN 4" or "IF A>4 THEN 3" - If the condition is true, then it will GOTO the number. "IF B<6 THEN 7" means, "If B is less than 6, then goto line 7." Operators are = > <, as well as ! (not-equal), & (bitwise AND), | (bitwise OR) . If you don't specify any line number, and just say "IF A=1 THEN", then the following line is treated as the operation to perform if the condition is true. Or is skipped if the condition is not true, so make sure there is a line afterwards for it to skip to, even if it's just "STOP".

LMSG: "LMSG 5" - Displays a long message. Does not wait for a keypress.

LOADSCR: "LOADSCR" - Loads the screen that was saved (see SAVESCR).

MACRO: "MACRO 5" - Jumps to the beginning of another macro.

MAPADD: "MAPADD X Y I" - Stacks an item onto a map location. The first two arguments are X Y coordinates in the current room or worldmap and the location must be within about 16 spaces of the player. The final argument is an object number. This command can be used for any object types on room maps, but should only be used for items and vehicles on worldmaps (because terrain does not stack).

MAPCHK: "MAPCHK 13 7 X" - Looks at a map location and sets a variable with whatever object type it finds there.

MAPCHKD: "MAPCHKD 13 7 X" - Looks at a map location and sets a variable with whatever object data value it finds there. Normally only used to inspect the value used for custom spaces and custom obstacles.

MAPSET: "MAPSET 123 45 #6-GRASS" Changes the object type at a map location into the object type specified. In this example, it would turn the existing terrain at (123,45) into #6 grass. If a creature is at the spot, it will be destroyed. Generally you should only use this command to turn one object into another of the same general type (terrain into terrain, portals into portals, items into items). Note that you must enter the command just as numbers. For example, the command mentioned earlier would be typed in as "MAPSET 123 45 6". If the third value is a number (you can also use variables) the macro editor will display the object name just for convenience.

MAPSETD: "MAPSETD 123 45 36" Sets the object data at a map location. Arguments are the same as SETOBJ (three arguments: map X, map Y, and number or variable for the data value.) Normally only used to set the value used for custom spaces and custom obstacles.

MAPTAKE: "MAPTAKE 6 4 0" Opposite of MAPADD – removes the top item from a stack. Use "0" for the third argument to take any object, or specify an object number to take only that object (and do nothing if that object is not on top.) This will remove any object types in rooms, but in worldmaps it only takes away item-type objects, and terrain is ignored. Note also that object properties are never checked, so "replaced by" values are irrelevant.

MOSAIC: "MOSAIC 6" - Displays a Mosaic exactly as shown in the Mosaic editor.

MUSIC: "MUSIC 59" - Plays a sound on the "music" channel – a looping sound channel that can be used for music. The region-specific music settings will override this once the character is able to walk on the map, and sounds on this channel do not mix (playing another sound will stop any music sound already in play) so it's good for temporary music scores during cutscenes or dialogues. Play sound 0 to stop it.

OFFARMR: "OFFARMR 2" - Turns OFF armor type (1-7) from being counted. (See USEARMR)

PAUSE: "PAUSE 10" - Creates a delay, measured in 10ths of a second. "PAUSE 10" would pause for 1 second.

PAUSEK: "DELAYK 40" – Like PAUSE, but allows the player to press a key to interrupt it.

PICTURE: "PICTURE 7" - Shows a BMP picture, silently waits for a keypress, and then fixes the palette and clears the screen. See the section on custom content for info.

QA: "QA PUT ONE OF THIS HERE: #67" Queues an Action. This is a work-around for the fact that macros cannot perform many things that actions can: it allows the macro to queue up some actions that will be performed after the macro ends. Three important notes here. First, the line must be entered with the action *number*, not the name. (For example, you must enter "QA 24 1" which will then become "QA summon creature #1". See the list of object actions, in the section on the object editor, for the action numbers.) Note also that this will ONLY work if the macro itself is called by an action. Macros called other places (creature death macros, intro macros, step macros, hourly macros) will not be able to queue actions. Third, these will be called in order -- there can be up to 16 queued up by the macro -- but they don't start until the macro ends. You can, however, have the last QA action call another macro (although the limit of 16 applies to the entire sequence, so this cannot be used to QA beyond that limit)

Q TILE: "Q TILE 960 AT 1" – Displays a quarter of a graphic tile at certain locations above/below the menu. The first number is the tile number. Take the original the graphic tile number and multiply it by 4. This would be the first (top-left) quadrant; add 1, 2, or 3 if you want to use the top-right, lower-left, or lower-right quadrant. The second number (AT 1) is the location along the bar, and is a number 1 through 6 (to go above the menu – in the time bar area, specifically) or 7 through 12 (to go in the blank space between the menu and the save/quit keys). Useful if you want an icon to symbolize something and have it always present. Normally used with skins – refer to the section on custom time and screen skins.

SAVESCR: "SAVESCR" - Saves the current screen image to a buffer, to be redisplayed with LOADSCR. The macro processor generally assumes you will leave the screen like you found it, so you may need to start your macro with SAVESCR and end it with LOADSCR. It depends on what you are doing and where your macro is being called from; you should test it in your game to see.

SAYVAL: "SAYVAL HP AT 3 23" - Displays a value on the screen. Lines 1 to 22 are LMSG lines, 23-24 are SMSG lines, 25-26 are BMSG lines.

SET: "SET A=7" or "SET MONEY=A+7" or "SET A=B*2" or "SET D=R/2" ... Sets the value of a variable.

SETPORT: "SETPORT 18 TO 3 0 37 112" or "SETPORT 6 TO 4 7 3 4"... Sets a portal location. Notation is [Region,Room,X,Y] – use room 0 for worldmaps. Portal numbers can be found in the map editor by moving the cursor over a portal. An example would be a setup where you can get into a taxi, tell the driver where you want to go, and his dialogue can trigger a macro that will set the exit door to lead to any one of many places. See also the PORTAL variable.

SHOWBMP: "SHOWBMP 5" - Would display a bitmap picture file. Similar to the PICTURE command, but it doesn't do the housekeeping tasks afterwards, so you must use FIXPAL and CLRSCR afterwards (before the end of the macro) or you may have funky colors. FADEIN and FADEOUT can also be used.

SHOWLOC: "SHOWLOC" - Displays, "You are at (X,Y)" at screen bottom.

SHOWMAP: "SHOWMAP" - Shows the local area (48x48) in a tiny window.

SMSG: "SMSG 5" - Displays a short message in the usual pop-up box, and waits for a keypress.

SOUND: "SOUND 7" - Plays a sound effect on the regular sound channel. "SOUND 0" will stop any playing sounds on this channel. All normal sound effects (from actions, combat, bump sounds, creature sounds) are played on this channel. Sounds on this channel do not loop, but they do mix; you can have several playing at once over top of each other.

SPEECH: "SPEECH 63" - Plays a sound on the "speech" channel – a non-looping sound channel that can be used for voice-overs, such as in dialogues. Sounds on this channel do not mix; playing another sound will stop any speech sound already in play. Play sound 0 to stop it.

STOP: "STOP" - Ends the macro.

TBMP: "TBMP 306" - Like SHOWBMP, but the color of the lower-left pixel is considered the "transparent" color for the rest of the image. Useful for overlaying one BMP onto another, or overlaying special effects if you can match the BMP palette with the game palette. Warning: it is likely to be slow, as it has to loop through each pixel.

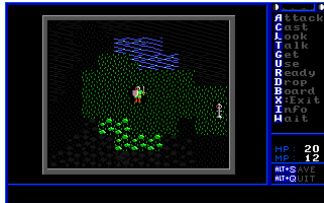
TMOSAIC: "TMOSAIC 5" - Like MOSAIC, but blank space is transparent.

USEARMR: "USEARMR 2" - Turns ON armor type (1-7). (See OFFARMR)

WAITKEY: "WAITKEY" - Waits for a keypress. It will say "PRESS A KEY..." If you want to wait for a keypress without any screen text, use "GETKEY 0"

Misc. Tips and Tricks

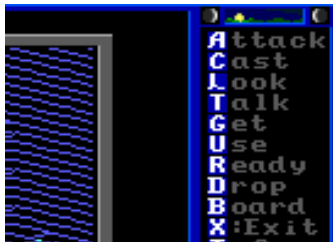
Darkness and the Game Clock



You can enhance the realism and mood of your game by giving a sense of time, and/or a sense of darkness. There are a few ways to do this, and you can use any or all of them.

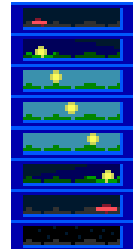
Most of the settings are in the Configuration Editor. The TIME setting is the time of day, in 24-hour time. The DAYS ELAPSED setting shows how many days have passed in the current game (in a new game, it would be 0). At the start of each day, there is the option of running a macro, the DAILY MACRO. If you wanted a game with a time limit, this macro could check how many days have elapsed, and act accordingly. The DAY/NIGHT VIEW

SIZING option can give the player feeling for the passage of time; during the day, the display will be normal, but as night falls, the game will announce "Night is falling" and the distance that the player can see will be a shrinking circle. At night, the player can only see a short distance. At dawn, it will say "Dawn is breaking" and the display will slowly return to normal. Use of objects that provide temporary light (the TORCHLIGHT and DAYLIGHT actions) can increase the view radius.



Either way, you can enable (in the Configuration Editor) display of an on-screen clock or an on-screen image showing approximate time of day (and this image can be customized, see the section on custom skins).

If you decide to use these options, you will need to make a few settings in the Map Editor as well. Each map has its own setting for the number of seconds or minutes that will pass during one step. For an outdoor map with a large scale, try a large setting, maybe 5 minutes; for an indoor map, however, the scale is much smaller, so 5 or 10 seconds per step might be appropriate.



There are also map editor settings that can customize the visibility options (if you use them) for individual regions.

Advanced Vehicles

Vehicle objects have many options that allow you to fine-tune exactly where they can go. In this example, we will have four vehicles: Ship (capable of blue-water and harbor sailing), skiff (capable of harbor and rough water travel), hovercraft (capable of harbor and some land travel), jeep (capable only of land travel), tank (capable of land and shallow water travel), and helicopter (capable of air travel).

Our vehicle objects would be set up as follows:

*FLATWATER V

(This never appears to the player, but you'll see below how it's used)

Can go: only where needed.

Counts as: itself.

SHIP

Can go: only where needed.

Counts as: *FLATWATER V

SKIFF

Can go: only where needed.

Counts as: *FLATWATER V

HOVERCRAFT

Can go: anywhere player can.

Counts as: *BOAT

JEEP

Can go: anywhere player can.

Counts as: itself.

TANK

Can go: anywhere player can.

Counts as: SKIFF

HELICOPTER
Can go: anywhere player can.
Counts as: itself
Boarding macro: SET FLY=1
Exit macro: SET FLY=0

And our terrain objects would be set as follows:

DEEP WATER
Passable to: SHIP

WATER
(This object will represent calm deep water, like harbors and large rivers)
Passable to: *FLATWATER V

WATER
(This object will represent shallow rough water, like small rivers or coastal areas)
Passable to: SKIFF

GRASS
Passable to: anything

FOREST
Passable to: if not holding HOVERCRAFT

HILL
Passable to: if not holding HOVERCRAFT

MOUNTAIN
Obstacle

Magic Spells

In this example we will set up some spells that require (in addition to costing magic points) a reagent, and have advanced effects (one makes the player invisible, one allows the player to pass through walls, and one dispells an energy field).

We will need to set up some objects:

SPICE (Misc item) -- this will be the reagent. In the examples below the macro refers to it by object number; for these examples it will be #65, but your number of course may vary. You may even have multiple reagents.

For the "dispell energy field" spell:

*DISPELL FIELD (Weapon) -- this will never be owned by the player, but will be used by the spell.
Damage 0
(Note the damage of 0, this will make it be ineffectual if used against anything else, like a creature)

ENERGY FIELD (Obstacle)
Action: play sound: no sound (We need an "action" to happen to satisfy the "disappear" condition, but we're using an action that does nothing)
Trigger: attacked by weapon *DISPELL FIELD
Disappear: when action triggered
Replaced with: eliminate (if you're using this stacked onto another terrain, in room maps) or,
Replaced with: FLOOR (if you know what you're using it on, in worldmaps)

DISPELL FIELD (Spell)
Action: MACRO 1
2nd Action: Allow user to attack with weapon: *DISPELL FIELD
Energy required: 10

Then we need to set up the macro:

Macro 1:

```
1 IF INV[65]>0 THEN 5
2 BMSG 1 // "You have no spice!"
3 SET SUCCESS=0
4 STOP
5 SET INV[65] = INV[65] - 1
6 INVERT // the rest of the macro is optional, this is just special effects
7 SOUND 50 // this assumes you have replaced sound #50 with something suitable
8 PAUSE 5
9 INVERT
10 STOP
```

For the "invisibility" spell:

INVISIBILITY (Spell)
Action: MACRO 2
2nd Action: none
Energy required: 15

You will also want to make a special graphic tile to represent the player while they are invisible. For our example we will use tile # 238.

The macro for the spell:

Macro 2:

```
1 IF INV[65]>0 THEN 5
2 BMSG 1 // "You have no spice!"
3 SET SUCCESS=0
4 STOP
5 SET INV[65] = INV[65] - 1
6 INVERT
7 SOUND 50
8 PAUSE 5
9 INVERT
10 SET INVISICO = 238
11 SET INVISDUR = 20
12 STOP
```

Finally, the "walk through walls" spell – note this only works in scrolling (worldmap-type) maps:

BLINK (Spell)
Action: MACRO 3
2nd Action: none
Energy required: 20

The macro for the spell:

Macro 3

```
1 IF VEH > 0 THEN 13 // this spell shouldn't be allowed to work if you're on a vehicle, if you have vehicles that fly.
2 IF INV[65]>0 THEN 6
3 BMSG 1 // "You have no spice!"
4 SET SUCCESS=0
5 STOP
6 SET INV[65] = INV[65] - 1
7 INVERT
8 SOUND 50
9 PAUSE 5
10 INVERT
11 SET FLY = 1
12 SET F = 4 // tune this value if you want the player to have more or fewer steps where they can walk through walls
13 STOP
```


This one will also need a step macro to include the following lines of code to take the player out of fly mode after several steps. Change the line numbers as necessary:

```
1 IF F = 0 THEN 8
2 IF F = 1 THEN 5
3 SET F = F - 1
4 GOTO 8
5 SET FLY = 0
6 SET F = 0
7 GOTO end
8 STOP (or the rest of your macro can continue here...)
```

Levers and traps

You may want to have levers that open and close doors; for example, a dungeon room that, when you pull a lever, closes off one exit, opens another, and opens a closet full of monsters. (By changing triggers and appearances, you could also make this a trap instead, triggered by stepping on it) The object numbers below are examples; if yours are different, replace the numbers used in the macro.

Objects required:

#101 LEVER 1 (space or obstacle)
Graphic tile: (shows a lever, pointing up)
Passable to: nothing
Action: Run macro
Triggered by: Bumping
Disappear when: action triggered
Replace with: LEVER 2

#102 LEVER 2 (space or obstacle)
Graphic tile: (shows a lever, pointing down)
Passable to: nothing
Action: do nothing

#103 SEALED DOOR (space)
Passable to: nothing (opaque)
Replace with: OPEN DOOR

#104 DOOR (space)
Passable to: anything (opaque)
Replace with: SEALED DOOR

#105 OPEN DOOR (space)
Passable to: anything (transparent)

And here's the macro.

```
1: SOUND 18 // replace sound #18 with something appropriate
2: QA 18 103 // replaces all the sealed doors with open doors
3: QA 18 104 // replaces all the regular doors with sealed doors
```

In room-type maps, because you can stack terrain, you can create even more interesting effects, like having creatures appear in the middle of the room (perhaps cutting off the player's retreat.) Just add this object:

#106 FLOOR 2 (space)
(Make it look and act just like your usual room floor)
Replace with: (eliminate)

In your room, place some creatures, then place this floor over top of them. This will suppress them (they won't act, move, be

visible, etc.) until the floor is removed from them. Then, add this line to your macro:

```
4: QA 18 106 // removes the fake floor, revealing the creatures
```

You could of course use this trick for more than just creatures; anything can be stacked in room type maps, so you can also use this trick to reveal items, portals, other switches, etc. Assuming you are hiding the other items under the same #106 FLOOR 2 object, the macro would not change.

Mountain Climbing

The Ultima kit includes an object called “climbing gear” that can be used to climb mountains, with the custom key command “K (Klimb)”. (Similar to Ultima V). Note that it uses the 'B' variable in a way that would prevent you from using it elsewhere. Here's how it's done:

In the Configuration Editor:

Extra Command #1 key: 'K'
Extra Command #1 macro: 6

The objects required:

MOUNTAIN (space, opaque)
Action: Set variable B to 0
Triggered by: passing
Passable to: Variable B > 0

CLIMBING GEAR (misc. item)
Action: do nothing

Finally, here's the macro – for this example, the object number for CLIMBING GEAR will be #80.

Macro 6

```
1: IF INV[80] = 0 THEN 5  
2: BMSG 1 // “Select a direction to climb”  
3: SET B = 1  
4: STOP  
5: BMSG 5 // “You have no climbing gear!”
```

Poisonous Swamp

Here's a simple example of Ultima-style swamp, where if you're not wearing “swamp boots”, you will be poisoned:

The objects:

SWAMP (space)
Action: Run macro #7
Triggered by: passing

SWAMP BOOTS (armor)
(For the example below, we will say the object number for the boots will be 116)

The macro:

```
1: IF WORN[116] > 0 THEN 4 // if wearing swamp boots, do nothing  
2: IF POISON > 1 THEN 4 // if you're already poisoned worse, do nothing  
3: SET POISON = 1 // poison player at rate of 1 HP per 10 turns  
4: STOP
```

Ultima Moongates

In this example, we will be implementing Ultima-IV style moongates, where there are 8, they all appear on the main world map, and they are linked to the phases of two moons. The first moon will be represented by variable D, and determine the currently active moongate; variable E will be the second moon and determine the destination of the moongate. (D2 is also used to determine how long until the next change of D, and so must not be used elsewhere; Z2 is being used as a temp variable, and can be used as such elsewhere if desired)

Note that all of this is already set up in the ULTIMA kit: the Britannia map included in the kit has fully functional moongates.

First, eight portal objects need to be created. They can all be named MOONGATE later, but for troubleshooting purposes, name them separately:

MOONGATE 1 (portal)

Passable: when D = 1

Action: run macro (use the number of your moongate entry macro, see below)

Triggered by: passing

Disappear: while impassable

Replaced with: GRASS

MOONGATE 2 (portal)

Passable: when D = 2

Action: run macro (use the number of your moongate entry macro, see below)

Triggered by: passing

Disappear: while impassable

Replaced with: GRASS

Repeat this for 3 through 8.

If you want the moons indicated on the screen, in Ultima IV style, then two special graphic tiles will need to be created, with four moons each, one on each quarter of the tile.

Now, some macros are required. First, the following lines should appear in the INTRO macro, to set the moons to a starting point:

```
SET E = 0
```

```
SET D = 1
```

```
SET D2 = 1
```

Next, an "hourly" macro to drive the changes of moon phase. In this example, the first moon will change every three hours (game-time hours, not real-time hours), and the second will change every hour.

```
1: SET D2 = D2 + 1
```

```
2: SET E = E + 1
```

```
3: IF E < 9 THEN 5
```

```
4: SET Z = 1
```

```
5: IF D2 < 3 THEN 10
```

```
6: SET D2 = 1
```

```
7: SET D = D + 1
```

```
8: IF D < 9 THEN 10
```

```
9: SET D = 1
```

```
10: SET Z2 = 955 + D
```

```
11: QTILE Z2 AT 1
```

```
12: SET Z2 = 955 + E
```

```
13: QTILE Z2 AT 6
```

Next, a "moongate entry" macro that will set the destination of the moongate upon entry, based on the phase of the second moon. Note the way it conditionally branches by GOTO (moon phase times 2, plus 4). The first two lines and last two lines provide a sanity check; if E is outside the 1-8 range, the portal will go nowhere.

```
1: IF E < 1 THEN 22
```

```
2: IF E > 8 THEN 22
```

```
3: SET Z2 = E * 2
```

```
4: SET Z2 = Z2 + 4
```

```
5: GOTO Z2
```

```

6: SETPORT PORTAL TO 1 0 19 27
7: STOP
8: SETPORT PORTAL TO 1 0 31 109
9: STOP
10: SETPORT PORTAL TO 1 0 54 117
11: STOP
12: SETPORT PORTAL TO 1 0 34 12
13: STOP
14: SETPORT PORTAL TO 1 0 119 230
15: STOP
16: SETPORT PORTAL TO 1 0 201 33
17: STOP
18: SETPORT PORTAL TO 1 0 199 88
19: STOP
20: SETPORT PORTAL TO 1 0 173 42
21: STOP
22: SET SUCCESS=0
23: STOP

```

You must edit the macro to use your own SETPORT locations. They appear here in order (2nd moon phases 1 through 8) and will cause the moongate to send the player to the coordinates specified, instead of the coordinates of the portal. For example, when the second moon is at phase 3, any moongate the player steps into (if it's open; determined by the 1st moon) will take the player to the coordinates in the macro.

You can find the coordinates in the map editor. The first number is the region number. The second number is the room number (0 if it's a worldmap-type map). The third and fourth numbers are the X,Y location, which you can see in the map editor: for example, the highlighted location in the picture below is 22,33. If this were in region 1, the SETPORT syntax would be "SETPORT PORTAL TO 1 0 22 33" (The keyword "PORTAL" means "the current portal". Refer to the section on macro editing for more info on the SETPORT command.)



Finally, place the portals on the map. You will be prompted to set destinations, but these destinations will be ignored, because the moongate entry macro will set the new destination as the player enters.

Terrain Hit Points

In addition to using action triggers on objects to trigger an action "when attacked", you can also make terrain objects have something like "hit points" where weapons can destroy them. build in the ability to attack inanimate objects. There are a couple rules: First, weapons always do standard damage to terrain; skill is not factored in. Second, an attack is all or none: it will either destroy the terrain or it will have no effect. You set an object with a threshold, called "Structural strength." For example, set a chair to a strength of 7. Any weapon that can do a damage of 7 or more will destroy that chair with a single blow. Any weapon that does less than 7 damage will never have any effect on that chair.

When the terrain is destroyed, it can simply disappear. On room-maps it could be part of a stack, so this might be no problem. On worldmaps it could leave an inconvenient blank spot. Another option, therefore, is to set a "Replaced by" option. When the terrain is destroyed, it is removed from the map and replaced with the terrain or item that you specify. You could have the chair replaced by terrain called "pile of junk," for example. Also, the terrain can run a macro when it is destroyed.

Wandering monsters

This term, common in most role playing games, refers to random encounters. ACK allows you to provide for random encounters by creating wandering life that will sometimes be added to the map when the map section or room is loaded.

While editing a map, the "Wandering" option allows you to place a wandering monster. It will appear in the spot that you are placing it, but only sometimes. You can decide the creature that might appear, and the odds that it might appear when the room or section is loaded. (Rooms load much more frequently than worldmap chunk sections – you will therefore want to tune your odds much lower in room maps – but it depends on how your player travels.)

This option allows you to make an unlimited monster supply. To make a dangerous countryside, place a few of these wandering monsters. (You can have up to 224 per region.) Then, each time the player enters that area, there might be a couple monsters to deal with. The player can kill them this time, but next time, there might be some again. Or there might not; they are, after all, random encounters.

Creatures will not accumulate without limit; ACK will not stack creatures, so another creature would only be placed if the previous creature moved or died. Also, worldmaps can only have 254 creatures active per region.

In the map editor, you will see a "W" on any square with a wandering monster. Moving the cursor over the spot will tell you the type and odds. If you press "W" again when you are already on top of a wandering monster's spot, you will delete that wandering monster.

Running External Programs

ACK can run external programs from a macro. ACK can even swap itself out of memory and free up a large amount of memory for your external program, if you wish. If you have a program that will display a picture or an animation, or play special sounds, you can have ACK call that program as well.

To run an external program, you use the EXEC table from the main menu of the Macro Editor. For each EXEC entry, you give it the name of the program, the command line parameters, and a few other options. Some have to do with the graphics settings going to and from the external program. The SWAP setting determines how ACK will handle memory. If you do not swap memory, then ACK can pick up the macro right where it left off, but memory will be very limited. If you do choose to swap, you will have most of the base memory free to use, but the macro/conversation will terminate.

Debugging/Playtesting Adventures

The following key commands can be used while playing an adventure. They must first be enabled in the configuration editor.

The following commands require "Debug Mode":

TAB: Reload any changed graphic tiles, object definitions, and creature definitions from disk.

Control-P: View or set a variable

Control-X: Run a macro

Control-F: Toggle "Fly Mode" (walk through obstacles, disable step and bump actions, see through opaque)

Control-W: Warp to a location.

Debug Mode also enables logging; you will find a .LOG file in your adventure folder. You should disable Debug Mode before you distribute a game; the log file can get quite large, and the debug-mode cheats can break the playability of an adventure or even crash the game.

The following commands require "Cheat Mode" (and also work in Debug Mode)

Control-Z: Toggle "Invisible Mode" (hostile creatures don't react to player)

Control-T: Show the current time.

Control-L: Show the current location.

Control-H: Restore HP and MP.

Custom Content: Pictures, Sounds, and Skins

ACK has support for external content, like pictures and sound effects. All custom content is called by number, and is named as *number-description* -- For example, "307-Intro_Picture.BMP" or "7-Cheering.RAW". The description part is optional. If you use a description longer than 6 characters, ACK will still see the file, but the description will be cut off. Note that you may have to restart your ACK session before it can see the new files you copy in.

You can also choose to leave the description off; for example, "307.BMP" will also work.

ACK Sound Effects:

Previous versions of ACK used an assortment of sound formats, but as of v3.1 the sound engine has been changed to play just one format: .RAW files. You can convert other formats, such as MP3 and WAV (and even VOC files from earlier versions of ACK) using the bundled sound conversion tools, SoX and LAME. ACK includes a simple script for performing these conversions, if you are using Windows.

The script is in the ACK\SOUNDS\CUSTOM folder. Simply rename your source files (remove any spaces, and give them a name ACK can use, like 412-applause.WAV) and then place the files you want to convert into the "in" folder. Run "convert.bat", and it will create .RAW versions of your sounds in the "out" folder.

If you are not running Windows, and wish to download versions of these third party applications appropriate for your system (or just want to learn more about these programs), you can go to the following sites:

SoX (for converting just about all formats except MP3 and MIDI to RAW) can be found at <http://sox.sourceforge.net>

LAME (for converting MP3 to WAV, which SoX can then convert to RAW) is at <http://lame.sourceforge.net>

Additional tools worth mentioning:

SynthFont (for converting MIDI files to WAV) can be found at <http://www.synthfont.com> and can help you convert MIDI files (many music samples for other old games are in this format) into a format that SoX can convert to RAW.

Audacity (for editing and recording WAV and MP3 files) is at <http://audacity.sourceforge.net> and is an excellent tool for editing sound samples before converting them to RAW. It supports MP3 files if you provide the lame_enc.dll file from LAME, above.

If you do use third party tools for the RAW conversion, make sure they are set to encode .RAW files as headerless, 8-bit signed, single channel, 11025 Hz. These files can be just about any length (from a fraction of a second, for special effect sounds, up to 10+ minutes long, for music or ambient sounds).

Sounds are then played in ACK on one of four "channels"; the SOUND, MUSIC, AMBIENT, and SPEECH channels. Any or all of these can play sounds simultaneously, and each of these have corresponding macro commands. The SOUND channel is for non-looping sound effects, and all actions, creature sounds, and combat sounds use this channel. Sounds on this channel can mix (more than one can play simultaneously). The MUSIC channel is a looping channel used for in-game music, and can be set by region in the map editor. The AMBIENT channel can be used by macros to play other looping sounds, like ambient noises; and the SPEECH channel can be used by macros or dialogues to play non-looping, non-mixing (playing one will end any other SPEECH sounds currently in play) sounds for voice-overs. Any ACK sound can be used on any channel.

If you copy custom sounds into your adventure and they are within the 1-99 range allowed by the "sound picker" (selecting sounds for creatures and actions is done through a scrolling list) then they will show up in the list as their filename instead of as their default name.

Custom Pictures or Photos

BMP: Both Actions and Macro scripts can display BMP files. They must be saved in Windows format, 320x200, 256 color. The image is displayed by either the "SHOW PICTURE" action, a PICTURE setting in the config for a title or intro picture), or the PICTURE and SHOWBMP macro commands. Actions can use 255 pictures, macros can use up to 64,000.

When used in a macro, the PICTURE command will do everything for you: show a BMP image, silently wait for a keypress, then clear the screen and reset the color palette. If you need more control than this, the SHOWBMP command lets you use other commands to do so manually: WAITKEY will ask the user to press a key (or GETKEY 0 will wait for one without asking); or you may use the DELAY command to delay and then show another picture without a keypress, like a slideshow.) You can also use the PAUSEK command to get the best of both worlds; delaying for a time period, but allowing the user to press a key to hurry it up. (And if you set the ESCGOTO variable, pressing ESC can be handled as well.)

After you finish displaying pictures, you can use CLRSCR and then FIXPAL to clean up the display. You may want to precede them with a "SAVESCR" and follow them with a "LOADSCR", depending on where you are using the macro; this allows the macro to leave the screen just as it was when the macro started.

Custom Skins

There are a few places in ACK where custom BMP images can also be used to alter the appearance of basic game elements (often called a "skin"). **These skin BMP images must match either the ACK palette (if you use the default colors) or the palette you set in your PALETTE.BMP file.** If the BMP file has a different palette, it will be ignored, and you will see funky colors.

Note also that many of these features require an adventure to be set as v3.0 or higher. (Not a problem if you're creating a new game, as it will be v3.2)

The original copies of the SKIN files can be found in the ACK\SKINS folder. Do not modify the original copies of the files; copy them to your adventure folder, and modify the copies.

Palette:

If you have a PALETTE.BMP file in your adventure folder, ACK will use the 256-color palette in this file instead of using the default ACK palette. Note that it reads only the color table from the metadata of the file; it ignores the content of the image entirely. So any old image that has the palette you want will do. Much of ACK assumes, however, that colors 0-15 will be the "standard" 16 DOS colors (and especially that 0 will be black), so some tweaking may be necessary.

Time Bar:

This is a small (58 pixel by 10 pixel) bar in the upper right corner of the screen, above the key menu, showing a different small image depending on time of day. This uses the skin file TSKIN.BMP. You can also make region-specific skins by naming the files TSKINx.BMP (where x is the region number.)

You can customize the look of this even further by using the QTILE macro command to place symbols onto it – positions 1 to 6 appear on top of this bar.

Screen Skins:

If you set (in the configuration editor) the menu to use a BMP skin, it will use RMSKIN.BMP (for room maps) and WMSKIN.BMP (for world maps). You can also make them region specific by naming them RMSKINx.BMP or WMSKINx.BMP (where x is the region number). Try to leave the commands in their normal spots; these parts of the screen are hot-zones for the mouse pointer.

Once again, you can customize the look of this even further by using the QTILE macro command to place symbols onto it. Positions 1 to 6 appear above the menu, and 7-12 appear between the gameplay keys and the save/quit keys.

Dialogue Skins:

Dialogue skins can be used to display a portrait of the person or creature the player is talking to. Dialogues can also change the portrait during the conversation, if you want to create the illusion of other people "interrupting". If the game finds a TALKx.BMP file (where x is the dialogue number) then it will use it as a background during the dialogue. The "skin" command in the dialogue editor will change the skin (with 0 being the original one that was loaded.)

Startup, Character Creation, and Saving/Loading games:

The character creation screen will display on top of START.BMP. If a file called ABOUT.BMP is present, it will show on the menu as an "[?] ABOUT THIS GAME" option. And if SAVEGAME.BMP and LOADGAME.BMP are present, they will be used as skins for those features.

Appendix A: Running ACK in DosBox

ACK is a DOS game (DOS being the operating system that ran on PCs before Windows) and does not run properly on modern PCs without special software: an application called “DosBox” that allows you to emulate a DOS PC. The ACK launcher will provide and run DosBox for you if you use Windows or MacOS, but if you can’t (or prefer not to) use the ACK launcher, you can use your own DosBox installation to run ACK.

The latest version of DosBox is available from <http://www.dosbox.com> or <http://dosbox.sourceforge.net>. Many Linux distros also provide DosBox as a downloadable package.

You will need to modify dosbox.conf for ACK to run properly. Here are some suggested settings:

```
[cpu]
cycles=9000

[render]
scaler=hq3x
aspect=true

[mixer]
nosound=false
rate=22050
blocksize=2048
prebuffer=100

[sblaster]
sbtype=sbpro1
sbbase=220
irq=5
dma=1
```

In Windows, you can also control how DosBox launches by modifying the DosBox shortcut. If you right-click your DosBox shortcut, and look at the “Target”, you will see something like this:

```
"C:\Program Files\DOSBox\dosbox.exe" -conf "C:\Program Files\DOSBox\dosbox.conf"
```

You can make a copy of your DosBox shortcut, modify it to use a special dosbox.conf file just for ACK, use the -noconsole switch to disable the separate status window, and add a -c switch to run ACK (assuming you use an [autoexec] section in your dosbox.conf to mount the folder, etc.). For example:

```
"C:\Program Files\DOSBox\dosbox.exe" -conf "C:\DOSgames\ACK\ackdosbox.conf" -noconsole -c "ack"
```

In addition to launching the main menu with the -c switch, you can also launch different editor modules (or the play module) directly. This is done through the ACK utility called ACKMOD. For example, this one will load the graphic tile editor:

```
"C:\Program Files\DOSBox\dosbox.exe" -conf "C:\DOS\ACK\dosbox.conf" -noconsole -c "ackmod -g"
```

The syntax for the “ackmod” utility is as follows:

ACKMOD -r [advname] [-mouse]	Run (play) adventure
ACKMOD -n	Create a new adventure
ACKMOD -c [advname]	Edit config
ACKMOD -i [advname]	Import kit
ACKMOD -f [advname]	Font editor
ACKMOD -g [advname]	Graphic tile editor
ACKMOD -o [advname]	Object editor
ACKMOD -t [advname]	Text (message) editor
ACKMOD -m [advname]	Map editor
ACKMOD -p [advname]	People/creature editor
ACKMOD -x [advname]	macro editor

Appendix B: What are all these files in my adventure folder?

ACK File Extensions

.1x	Temporary Saved Game Map File for Region #x
.2x	Temporary Saved Game Map File for Region #x
.3x	Temporary Saved Game Map File for Region #x
.Ax	Map File for Region #x
.BMP	Custom picture file. Naming format is x-Description.BMP or x.BMP, where x is the picture number.
.Bx	Map File for Region #x
.Cx	Map File for Region #x
.DLG	Dialogue Data
.DOC	Instructions for the game or kit (Text file)
.DQS	Saved Game Dialogue quest tracking file
.DQT	Temporary Saved Game Dialogue quest tracking file
.FNT	Font Data
.GRP	Graphic Tiles
.GR2	Animation Graphic Tiles
.ITM	Object Data
.KIT	Renaming the .MST file to .KIT will turn the adventure into a Kit.
.LOG	Log file generated during DEBUG mode.
.LMS	Long Messages
.MCO	Mapcolor data
.MCR	Macro Data
.MDX	Macro Data
.MOS	Mosaic Data
.MST	Master Adventure Configuration
.PLY	Saved Game
.PPL	Creature Data
.PSF	Extension to Creature Data: Sound Numbers
.RAW	Custom sound file. Naming format is x-Description.RAW or x.RAW, where x is the sound number.
.RGN	Map/Region Master File
.RGT	Map/Region Temp File
.RGX	Map/Region Saved Game File
.SMS	Short Messages
.TPF	Temporary Saved Game
.Wx	Map File for Region #x
.Xx	Saved Game Map File for Region #x
.Yx	Saved Game Map File for Region #x
.Zx	Saved Game Map File for Region #x

Other files that may appear:

PALETTE.BMP	256 color BMP file containing a custom color palette.
TSKIN.BMP	BMP file containing custom “time-of-day” skin images.
TALKx.BMP	BMP file containing dialogue skins (generally NPC portraits)
RMSKIN.BMP	BMP file used as screen background in room maps.
WMSKIN.BMP	BMP file used as screen background in world maps.
START.BMP	BMP file used as screen background during character creation.
SCREEN.TMP	ACK temp file.
*.TXT (MESSAGES.TXT, MACROS.TXT, etc.)	Exported reports.

If you find any .PMS, .CMF, .VOC, .SND, .Sx (where x is a number), or .Mx (where x is a number) files, they are files from older versions of ACK, and are no longer used. They can be deleted.

Preparing an Adventure for Distribution

First, in the Configuration Editor, make sure that “Cheat mode” is not set to “debug mode.” Next, any “Saved Game” files should be deleted before distributing an adventure. This can be done manually, or with the “IMPORT FILES” module. You may also wish to delete any “.TXT” files (the reports may contain spoilers) and set a password in the configuration editor (to disable editing and cheat keys.) You can then zip up your adventure folder and distribute it. You can also distribute your adventure as a standalone program – refer to Appendix C.

Using an Adventure as a Kit

To use an adventure as a kit, copy the folder to ACK\KITS, delete all Saved Game and Map files, and rename the .MST file to .KIT. Then create a .DOC file (the length and number of characters per line are very limited – use the .DOC file from one of the other kits as a template) describing the kit.

Appendix C: Distributing Standalone Adventures

An ACK adventure can be made into a standalone game (in Windows) by removing or renaming certain files. In this example, "GAMENAME" refers to the name of your adventure; replace it with whatever you like.

1. Perform the prep steps in Appendix B (setting cheat mode, deleting saved game, etc.)
2. Make a copy of the entire ACK Launcher folder, naming the folder to match your name (GAMENAME)
3. Make sure your game is in GAMENAME\ACK\GAMES\GAMENAME, if it's not already.
4. Delete all the folders in the GAMENAME\ACK\GAMES folder except for your GAMENAME folder (delete TUTORIAL, BRIGANDS, etc.)
5. Delete the ACK\UTIL folder.
6. Delete the ACK\KITS folder.
7. In the ACK folder, delete ACK.BAT, ACK00.EXE, and ACK04.EXE through ACK10.EXE. (leave ACK01 through ACK03 alone).
8. In the top GAMENAME folder, delete "ACK Launcher.exe", launcher.ico, dosbox.conf, and all the ".bat" files except for "r.bat".
9. Rename "r.bat" to GAMENAME.BAT (or PLAY.BAT, or whatever you like.)
10. Edit ackmod.conf in Notepad, and change the last line to read "ACKMOD -r GAMENAME -exit"
11. Make other DosBox changes to ackmod.conf as desired (such as scaler=normal3x, aspect=true, etc)
12. Edit the .HLP files in a text editor to change the wording, if desired.
13. Cut and paste from the documentation as needed. Make sure to include a note about requiring DosBox. Something in your documentation giving credit to me (Chris Hopkins) for writing the game engine would be nice too, as nothing will indicate that the game is anything but your handiwork at this point.

If you want to make a standalone Macintosh version, it is more complicated; Josh Lawrence (the author of the ACK launcher) has graciously offered to offer help or advice, time permitting. You can contact him at jjsonick@yahoo.com.

Appendix D: Game Walkthroughs

Eventually I'd like to replace the bundled adventures with better ones that take advantage of the features of ACK, or at least ones that are better balanced for difficulty and playability. However, since they serve as a passable demonstration of some of what can be done with ACK, it's worth having a walk-through to speed people through the play content, if desired.

Brigands

1. Enter the town to your southwest.
2. Find the mayor (follow the road north to the fountain, then west to his door) and tell him you will "HELP". He gives you a key to the armory.
3. Go through the door to your south to enter the armory, and take everything in the room. Hit "R" and ready all the armor and the crossbow. Leave out the south door and leave town.
4. Go southeast through the mountain pass. There's a small fort as soon as you come out of the forest – enter and pillage the place, there's a magic wand in the armory that will come in handy later, and some scrolls that will increase your skills a little bit. Make sure to use your healing potions as needed, and attack with the flaming oil (until you get the wand) when enemies are bunched together, as it attacks a 3x3 area.
5. When you finish there, go south and pass around the lake short counter-clockwise, and enter the castle. Pillage the castle (try out the wand!) to gather more potions and gear. One room has a silver key, so keep searching until you find it.
6. Leave the castle and head to the southeast part of the map, you'll find another fort. Enter it and use the silver key to get inside.
7. In the west wing, talk to the demon and offer to help him with the evil priest that has summoned him. He will give you another wand – this one is single-target but will almost always kill in one hit, and is the only weapon that can kill the evil priest. Ready the new wand.
8. In the east wing, kill the evil priest, move his corpse (pick it up and drop it somewhere else) and take and read his book, it will give you a keyword to use later.
9. Outside, join the battle encampment, and you will begin the final battle. Use the death wand, and don't kill the guys in blue, they're on your side. You're looking to get inside the inner labyrinth of the castle, where you need to find the demon king and read him the word from the book you found earlier. This will vanquish him and win the game.

Caravan

The game starts in Rome.

1. Stand next to the box in your house, and “look” at it to open it. Take everything from the box.
2. Leave, then follow the road west, then south, then east to the town center with the fountain. Enter the tavern to the east.
3. Talk to the bartender, he'll try to sell you a drink for two coins (although you can say no.) Ask him “job”, then “price”, and he'll offer to help you if you get him some “conditum” wine, and he tells you to ask around about it.
4. Talk to the bar serving girl. To talk to her, you have to buy a drink for two coins. If you ask her about conditum, she'll tell you to talk to Remus.
5. Go outside, follow the road west, then north, then go west into the shop.
6. Talk to Remus, and ask him about conditum. He'll ask you who told you to ask him (say “Tela”, she's the serving girl) and then he'll give you a bottle.
7. Go back and talk to the bartender, and talk to him about “conditum”, then give him the bottle. He'll tell you to talk to the soldier outside about his “friend”.
8. Go outside, then go west and talk to the guy in blue with the staff. Ask him about “friend” and he'll tell you about his friend Trajan, and how you should talk to the constable and have him release Trajan to be your “guide”.
9. Go east, then north, then east. Talk to the guy on the chair, ask him about “Trajan” and “guide”. He'll sign the release that lets you go talk to Trajan.
10. Go back to the town center fountain, then west, then south to the prison. Go through the west door, then talk to the guy in the north cell. Tell him you want him to be your “guide”. Go through the conversation, and he'll tell you the plan is to go to Egypt, to the city of Memphis, and then east from there.
11. Now you're ready to talk to Lydia, in the bar. She's sitting by herself on the north side of the room. She'll ask you to buy her a drink (you already bought one earlier). She'll ask where you're going, say “Egypt”, then “Memphis”, and when she asks you who is taking you, say “Trajan”. She'll ask to join; say yes.
12. Go back and talk to the constable, and ask for passage out of Rome. He'll also ask where you're going; again, Memphis. He'll give you a scroll that will let you leave town.
13. Go back to the town center, and south out the exit.

After a cut-scene, you will be in Egypt.

14. Leave your inn room, and find the palace. You're looking for a priest in a white robe. His name is Janus, and he will ask you to retrieve a stone tablet out of one of the pyramids.
15. Also talk to the aristocrat (Abdul) at the throne, he will tell you to “ask” Trajan what you need to do next. (If you want to skip the running around in the next few steps, just say “spare”.)
16. Go back to the inn, and find Trajan's room (it's just south of yours) and say “ask”, he tells you to go back and say “militia”.
17. Return to Abdul and say “militia”, he will tell you to find the captain of the guard and ask for “weapons”.
18. Find Akram and ask him about weapons, he will tell you to ask Abdul for the “spare” weapons.
19. Ask Abdul about “spare”, he will give you the militia supplies.
20. In the northwest part of town, on the riverbank, there is a dock and a boat. Get in the boat, and you will be carried across the river.
21. Go southwest to find the pyramids. One of them has an entrance; go inside.
22. Don't try stealing any of the treasures; if you do so, you'll be sealed inside! Take only the tablet at the far west end, then leave again. Go back to the boat, and back to the priest. Tell him “tablet”. He will arrange for your guide.
23. Somewhere in the north part of town, there's a damsel in distress – she explains that bandits are stealing her camels and asks you to take care of them; if you do, she'll give you a camel and the food you need for your trip.
24. Go inside her house and out the back door, and kill the camel thieves (they're easy) and then go back and talk to the woman. Go back and find the storage closet with all the bags and other supplies; take them, and take the scimitar. Also board a camel and take it with you.
25. Go find the shop, and sell a couple of those bags of trading supplies so you can buy food supplies.
26. You now have everything you need to leave Egypt. Save the game, then go out the exit on the east side of town.

After a cutscene, you will be in an encampment, under attack by barbarians.

27. Pick up the shield (there will be other goodies spread around the map) and make sure you have your best gear Readied at all times. Also make sure to use medicine potions whenever your health gets low.
28. You may be encumbered from the supplies you were carrying before; if so, drop them.
29. Walk around the map and kill all the bandits you find. Once you have killed the last one, you will be told you can return to your hut.

After a long cutscene, you will go to the final mini-game (which takes place hundreds of years later), in Japan.

30. Make your way through the mountain-pass maze (roughly northeast), and you will find a Japanese village. You won't be able to talk to anyone, however; you can't understand them.

31. Find another mountain-pass maze, to the south, and you'll find a hut out in the forest, with a man named Akechi.
32. Talk to Akechi. You'll try to have a conversation that won't go anywhere, but when you leave he'll give you a dictionary.
33. Use the dictionary, and then talk to Akechi again; you'll be able to understand him now. He tells you to return to the city.
34. Go back through the mountain-pass maze to the city. Talk to Harima (the guy with the fan) and you will end up offering to go over yet another mountain pass, and talk to his enemy Ikawa.
35. Talk to some of the peasants; you'll see that this Harima fellow isn't exactly loved by his people.
36. Find the mountain pass that leads to Ikawa's realm (it's much bigger than Harima's village) and find Ikawa – he's on the 3rd floor of the castle. You may want to save the game before you talk to him.
37. Talk to Ikawa. He will ask you a series of questions; as long as you don't say you want a war and you're on Harima's side (at which point he'll have you executed) you're okay; ideally, you should say that you're here for peace and that you're willing to be a hostage. He'll then ask you who you represent; say “caravan”. He'll ask you to join him instead of Harima; say yes, and he'll make you a samurai (that's good, and you get a neat new costume) but then order you to go challenge Harima to a duel (that's bad.)
38. Talk to the samurai around town again; they'll tell you to seek out a trainer named Jubei.
39. Find Jubei (he's in a hut in town) and have him train you. Once you're done, ready your new equipment (katana, armor, and helmet) and then go back to talk to Harima.
40. Not surprisingly, Harima doesn't surrender; you are sent out to duel him in front of his men. Kill him to win the game.