



ACE!

The Advanced Character Editor

ACE! 1.3 by Richard Drysdall © Flying Horse Software 1997-98.

Introduction

ACE! allows you to edit the 'save game' documents created by many commercial and shareware games (these documents are referred to as 'characters' by some games). Additionally, ACE! allows you to print details of these save games.

Initially, you can edit the save games from 'Bard's Tale', 'Doom' (1, 2 and Ultimate), 'Escape Velocity', 'Foobar vs the DEA', 'Harry the Handsome Executive', 'Heroes of Might & Magic', 'King's Bounty', 'Mantra' (1 and 2), 'Marathon' (1, 2, Infinity, Excalibur and Evil), 'Might and Magic' (1, 2 and 3), 'Omega' (version 0.75 and 0.80), 'Pathways into Darkness', 'Realmz' (4 and 5), 'Ultima III', 'Unlimited Adventures' (design passwords and characters), 'Wizardry Gold' and 'Yipe! III'. However, ACE! can be extended to edit other save games through the use of plug-in 'modules'.

ACE! is scriptable, and a number of example scripts are provided that extend ACE. You can also make your own scripts to automate your use of ACE.

ACE! is freeware.

Note to users of previous versions of ACE: The Save As command has been renamed Save A Copy As. The latter command makes a backup of the current file and does not set the active window to the new file.

Requirements

ACE! will run on any Macintosh or MacOS compatible computer. It requires System 7.0 or higher, 350k of disc space, and 384k of free RAM (preferred 1500k).

ACE! is distributed as a fat binary, meaning that it runs native on both 68k and PPC processors.

Installation

You can place the ACE! application anywhere on your hard disc. It can also be run from removable media such as a floppy disc or zip drive.

The modules folder must be named "Modules", and must be present in the same folder as the ACE! application. The "Scripts" folder should not be renamed.

If you do not intend to make modules or scripts for ACE, you can discard the "Development"

folder. If you do not want to use any scripts with ACE, you can discard the "Scripts" folder.

Instructions

Before editing any save game file, always make a backup copy. Do not dispose of the backup until you have tested the edited file.

Open the ACE! application and choose the name of the game whose save files you want to edit from the Module menu. Select Open from the File menu, and choose the save game file you want to edit. Alternatively, you can simply drag the save game files onto the ACE! application icon in the Finder - the correct module will be automatically selected.

A dialog will appear with various controls to allow you to edit the save game. When you have finished editing, select Save from the File menu to save your changes.

Choose Quit from the File menu to exit ACE. The save game is ready for use.

Hints

- Balloon help is provided for all menu's and most dialogs.
- In a dialog, you can type the first letter of a button to 'click' it (provided there is no editable text in the dialog).
- Hold down the option key while choosing Open from the File menu to open any file. This can be useful if the save game files have lost or have changed their type/creator information, or if the file has been copied from a PC format disc. Note that this still uses the current module - it does not select a new module.
- Hold down the option key while choosing a module from the module menu to temporarily toggle the current tabbed windows preference.
- Hold down the option key while choosing a script from the script menu to obtain a description of the script.
- If you are using MacOS 8 and have a contextual menu extension that lets you open files with a particular application, you can use this feature to open save game files with ACE!
- Keep an alias of the ACE! application on the desktop, or in your games or documents folder, so you can use drag-and-drop to open the save game files in ACE!

History

The idea behind ACE, an extensible save game editor, is not new. Nevertheless, I was motivated to write this application for many reasons:

1. To learn how to program the MacOS.

2. To provide a means of editing 'classic' (older) games. Although information or applications to edit save files from games such as Bard's Tale or Might & Magic existed at the time these games were popular, it is difficult, if not impossible, to obtain now.
3. To make an editor with a good Macintosh interface. ACE! uses no fancy fonts or custom controls, but provides standard interface items such as 'Save A Copy As', 'Revert', 'Undo', forward delete and movable modal dialogs. Other Apple technologies such as appearance manager, balloon help and drag-and-drop are also supported.
4. To make an application that could do more than just edit save game files. This is achieved via scriptability, which allows ACE! to be extended and used in new ways - the scripts provided are just a sample of what is possible.
5. To write a program that could be easily extended. ACE's plug-in modules are easy to write, requiring no programming knowledge and only a little knowledge of 'hacking'.
6. To write a program that other people could contribute to. I hope that many people will be encouraged to write ACE! modules or scripts for their favourite games, or extend the scripts or modules provided and thereby make ACE! a group project.

Feedback

If you like ACE, or have suggestions for improvements, or criticisms, please write to me! I would very much like ACE! to be a product of many people's ideas. If you would like to contribute, consider helping with one of the following:

1. Making new modules, or modifying existing modules.
2. Making new scripts and testing other scripting languages (Frontier, MacPerl).
3. Hosting a web page devoted to modules, updates, news etc.
4. Localisations to other languages (or even just advice on how to do this).
5. Bug testing - if you find a bug, and can reproduce it, please let me know.
6. Compatibility information - does ACE! run under MAE, Executor, Shape Shifter, A/UX etc?

Credits

As ACE! has grown, so too has the number of people who have contributed to it. In particular, these people deserve special thanks:

- Todd Clements for contributing the System 7.x moveable modal dialog code,
- Ron Nichalson Jr. for providing the interpreter,
- Martin Lai, Stephen Thornton, Phil Cotching and Andrew Ware for beta testing,
- Kevin Killion for a great deal of help with scripting,
- Kevin Barry and Bruce Sponagle for testing new modules,
- Michael Yopp for providing new Wizardry Gold information,

and everyone on the newsgroup 'comp.sys.mac.programmer.help' who has answered my questions.

Contact Information

I can be contacted at

E-mail: drysdall@waikato.ac.nz

Postal: Richard Drysdall
Physics Department (care of R.S.C)
University of Waikato
Hamilton
New Zealand