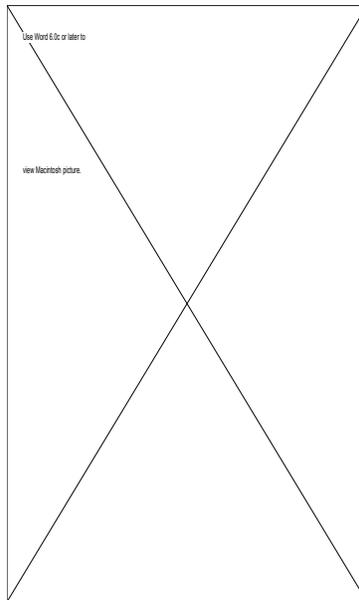


# A Lightning Bolt Product Manual...

## ExtractIt! for Stacks

Covers **ExtractIt! for Stacks**  
Version **4.0**



*Copyright ©1996 Lightning Bolt. All rights reserved. Version 4.0.0r1.*

# Disclaimer

---

Just in Case...

BY USING ANY OF THE SOFTWARE, YOU AGREE TO BECOME BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT USE THE SOFTWARE.

The accompanying software ("Software") is licensed, not sold, to you by [Lightning Bolt](#) for use only under the terms of this License, and [Lightning Bolt](#) reserves any rights not expressly granted to you. This License allows you to use the Software only on a single computer at a time. The Software is protected by United States copyright law. The Software contains trade secrets and to protect them you may NOT decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human perceivable form. YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, RESELL FOR PROFIT OR CREATE DERIVATIVE WORKS BASED UPON THE SOFTWARE OR ANY PART THEREOF.

This License will terminate immediately without notice from [Lightning Bolt](#) if you fail to comply with any provision of this License.

You agree that neither the Software nor any direct product thereof is being or will be shipped, transferred or reexported, directly or indirectly, into any country prohibited by the United States Export Administration Act .

THIS SOFTWARE IS LICENSED "AS IS" AND [Lightning Bolt](#) MAKES NO EXPRESS OR IMPLIED WARRANTIES WITH REGARD TO THE SOFTWARE OR ITS PERFORMANCE. IN NO EVENT WILL [LIGHTNING BOLT](#) BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMIT-

ATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

**Word** is copyright ©1983-1995 Microsoft Corporation. Version 6.0.1.

GetDir, and FullText copyright ©1990-95 Frederic Rinaldi.

SysEnvirons copyright ©1990 Jon Pugh. Version 1.0.

Files copyright ©1989 Guy Picciotto. Version 1.0.

HexToDec copyright ©1989 Itty Bitty Software. Version 1.0.

GetVolName copyright ©1990 Leonard Buck. Version 1.0.

**HyperCard** copyright ©1995 Apple Computers, Inc. Version 2.3.

Think C copyright ©1994 Symantec Corporation.

VolumePath copyright ©1989 Apple Computer, Inc. by Anup Murarka & Eric Carlson. Version 1.1.

SillyCon Valley created by Jim Ratliff, 1991.

All other copyrights respective to their owners.

For trademark and registered trademark information, see the [Trademarks](#) portion of the [Legalese](#) section, in the [Appendices](#) chapter (whew).

**ExtractIt! for Stacks** copyright ©1996 [Lightning Bolt](#). All rights reserved. Version 4.0.

This manual copyright ©1996 [Lightning Bolt](#). All rights reserved. You may distribute this manual freely in its original form, but you may not print or mass produce it.

Rescue! II copyright ©1996 [Lightning Bolt](#). All rights reserved. You may distribute the song in its original and unmodified form given that [Lightning Bolt](#) is credited for creation of the song.

# Table of Contents

**INTRODUCTION..... 5**

**LIGHTNING BOLT.....**

- ABOUT LIGHTNING BOLT.....
- CUSTOMER SUPPORT.....

**EXTRACTIT! FOR STACKS.....**

- INTRODUCTION.....
- VERSION HISTORY.....

**ABOUT THE MANUAL.....**

- READING THE MANUAL.....
- WHY WORD?.....
- WHAT HAPPENED TO DOCMAKER?.....
- IF YOU *STILL* CAN'T READ IT.....

**IMPORTANT INFORMATION..... 13**

**REGISTRATION.....**

- INTRODUCTION.....
- WHY IS IT FREE NOW?.....

**GENERAL OPERATION.....**

- USING "WINDOWS".....
- USING THE TOL.....
- WHEN IT'S TOO BIG.....

**OPERATION..... 17**

**STACK BROWSER.....**

- INTRODUCTION.....
- READING OR WRITING DATA.....

**SYSTEM BROWSER.....**

- INTRODUCTION.....
- OPERATION.....

**FILE BROWSER.....**

- INTRODUCTION.....
- NAVIGATING THE BROWSER.....

**RESOURCE BROWSER.....**

- INTRODUCTION.....
- OPERATION.....
- EDITING A RESOURCE.....
- COPYING A RESOURCE.....

**SCRIPT BROWSER.....**

- INTRODUCTION.....
- MAIN SAMPLE SCRIPTS.....
- USEFUL SCRIPTS.....

HYPERTALK REFERENCE.....

**MASSIVE EXTRACTIONS BROWSER.....**

  INTRODUCTION.....

  OPERATION.....

**LOCKING BROWSER.....**

  INTRODUCTION.....

  ENVIRONMENT.....

  STACK.....

**MESSAGE BROWSER.....**

  INTRODUCTION.....

  SENDING A MESSAGE.....

  GETTING REFERENCE.....

  RECORDING MESSAGES.....

**MATHEMATICS.....**

  INTRODUCTION.....

  BASIC MATHEMATICS.....

  CONVERTING NUMBERS.....

  ALL THE REST.....

**EDITOR.....**

  INTRODUCTION.....

  READING DATA.....

  SAVING DATA.....

**OBJECT COUNTER.....**

  INTRODUCTION.....

  COUNTING OBJECTS.....

**PREFERENCES.....**

  INTRODUCTION.....

  SETTING PREFERENCES.....

**HINTS AND TIPS.....**

  INTRODUCTION.....

**ABOUT THIS STACK.....**

  INTRODUCTION.....

  ABOUT THE THEME SONG.....

**APPENDICES..... 35**

**GLOSSARY.....**

**LEGALESE.....**

  COLOPHON.....

  TRADEMARKS.....

**CREDITS.....**

  STACK CREDITS.....

  MANUAL CREDITS.....

**INDEX.....**

gh gh gh gh gh

# Introduction

---

What Did I Just Download Again?

# Lightning Bolt

---

The Boys in Blue

## About Lightning Bolt

---

**Lightning Bolt** is a mostly **HyperCard**-based company geared to developing utilities, development tools (to help you *and* ourselves [I remember the days when **ExtractIt! for Stacks** was so buggy we had to use the finished portion of the program just to fix the buggy portions]), and games.

This is Lightning Bolt's 8<sup>th</sup> software release. Other releases include, in chronological order, **AddColor Resource Template**, **ExtractIt! for Stacks 2.0**, **ExtractIt! for Stacks 3.0**, **ExtractIt! for System 1.0**, **ExtractIt! for System 1.0.1**, **NumberSayer 1.0**, **NumberSayer 1.0.1**, **Fatal Encounter Episode I**, and (by the time you read this), **Fatal Encounter Episode II**.

Also, if you ever want information, want to check out previews, download some more of our software, or try out our beta programs, just visit our web page at <http://members.aol.com/MorganBolt/lightningbolt.html> and click around!

## Customer Support

---

We try as hard as we can to make life easy for you (you'd do it if you got paid, too). If you *ever* have a problem, don't hesitate to contact us! Our support staff will be happy to try to solve your problem (or at least *act* happy). Here are all the places you can reach us:

### Lightning Bolt Technical Support

---

733 Sutton Dr.  
Walnut Creek, CA 94598  
Phone: (510) 256-7895  
Fax: (510) 256-0277  
E-Mail: [MorganBolt@aol.com](mailto:MorganBolt@aol.com)

### Lightning Bolt Customer Relations

---

1026 Pinehurst Ct.  
Concord, CA 94521  
Phone: (510) 827-4958  
E-Mail: [LBoltSftwr@aol.com](mailto:LBoltSftwr@aol.com)

# ExtractIt! for Stacks

That is, the Program You Downloaded (in Case You Didn't Know)

## Introduction

**E**xtractIt! For Stacks has a nice, long history behind it, and a great

future ahead of it. It started it out in 1993 when I, Timothy Morgan, was a computer jockey and [Lightning Bolt](#) was only a large complex in my fantasy day-dreaming. I was pointing and clicking my way through the large archives of a CD-ROM called **CD7**, when I came across the **HyperCard** folder.

I had been a novice at **HyperCard 2.1** at the time. My **HyperCard** craze started out when I was fooling with a player on my Mac, and I got my way into the scripts of a stack called **BOMBER**. I looked at the scripts. Compared to the only other programming language I knew, ObjectLogo, it looked like English.

So I desperately tried to modify stacks with the player, all to no avail. I went as far as to try to combine two different players with **Resorcerer**. Eventually, I gave up.

Then, as I was looking through another CD-ROM (at that time, CD-ROMs were like the Internet to me), **CD Funhouse 8.0**, I noticed a folder called **HyperCard 2.1**. I opened it up, and there it was—a full **HyperCard**. There were only two files, *HyperCard 2.1* and *Home*, but that was enough for me.

I churned out graphical stacks with no scripts by the dozen. Then I decided it was time to learn HyperTalk. I had no reference, no manual, no guide. All I had was one stack from **CD7**, called **HyperTalk Reference**. It listed all **HyperCard 2.0** commands and functions and gave brief descriptions on them.

I worked with it for years. Crazy enough, HyperTalk is now my best language. Of course, now I have Danny Goodman's [HyperCard 2.2 Handbook](#) and the manuals to **HyperCard 2.3**. But I went from zero to a **HyperCard** Intermediate-Expert in only two years—with almost no reference.

Anyway, going back to **ExtractIt! for Stacks**, I was cruising the **HyperCard** folder. I came across a stack called **Script Changer**. It was made by a group of Apple workers in Switzerland. It was simple and straightforward. You select a stack with a **Choose Stack** button, select an object type with a radio button list, and enter the object number, the background number, and the card number. Then you click **Read** and it reads the script (no locking messages or anything). Click **Write** and it writes the script. That's it.

But to me it was a dream come true. I decided to work off of that, making my own stacks. I had gotten into the little fad of naming my stacks with an "It!" suffix (similar to some ancient stacks I created, like **OpenIt!**, an **At Ease** unlocker) so I decided to call it **ExtractIt! for Stacks 1.0**. Of course, I couldn't distribute it, though.

My next step was to beef it up a bit. I was experimenting with a stack I called **Mass Extractor**, and it was working pretty well. It had one button and one field. Click the button, select a stack, and it would return an organized list of all the properties of all the objects of that stack. I put it in **ExtractIt! for Stacks** and version 1.1 was born.

I then added a card called “Features” (something like that, I forget) making version 1.2. It had a button to DeProtect, a button to do resource copying, and two others which I can’t remember.

Version 1.3 saw another button added to the “Features” card, I think it was a text file editor.

Version 1.3.1 had minor bug fixes and the message box displayed progress for Mass Extracting.

Version 1.4 saw color.

Version 1.4.1 saw a better colorization scheme.

I was thinking of doing a version 1.5, where everything is a bit more organized, but I was attacked by the release of **HyperCard 2.3**, and an inner power told me to buy it. So I did. When I played around with it, I decided **ExtractIt! for Stacks** needed to have just as big of an upgrade. So version 2.0 was born. Major overhaul. The first cards were [Lightning Bolt](#) logos—the extended animated ones (that took about 30 long seconds to get through). Everything was organized into a Table of Linkages. I had about 12 features: [Main Extractions](#), [Massive Extractions](#), [Stack Unlocking](#), [Environment Unlocking](#), [Resource Copying](#), [Stack Macros](#), [Hints and Tips](#), [Help](#), and [Preferences](#). I put in a nice **DOC-Maker** Manual and set it off.

Version 3.0 was on its way already. I had immediately started cramming in more and more features after I let 2.0 go. It was getting huge. Its version progressed from 2.0.1 to 2.1 to 2.5 to 3.0. And it stayed there. I put it up on AOL, with a Word 6 and Word 5 manual (it was getting too hard to copy the Word manuals, page by page, into **DOCMaker** documents and reformat each chapter).

I then added some more features, fixed some bugs, and churned it out again as version 3.0.1. I was thinking of making a version 3.1, but an accident hit me which created version 4.0.

We had a web page! I was so happy. I put up **ExtractIt! for Stacks 3.0** and put a flash news on the front page so everyone could get their hands on it.

I was working on the [Previews](#) section of my web page, when I, not thinking about it, wrote in **ExtractIt! for Stacks 4.0**. Why, I don’t know. I had just vowed to take a break with **ExtractIt! for Stacks** and work on **Fatal Encounter**. I was going to delete it when I thought, *Oh well. I suppose we could do another overhaul sometime or other.* I thought of what I could write for it. I had no idea on what we could possibly do. So I wrote a “better, more integrated interface to make your work more streamlined.” I had **Copland** in mind when I wrote that.

I saved and uploaded the new [Previews](#) section, and then I stopped. *How the heck was I supposed to create a new **ExtractIt! for Stacks** that was all integrated and stuff?* I didn’t have a clue on how I could make it more streamlined and quicker.

I spent one week thinking, sketching, erasing, and screaming in frustration. *Everything* I tried to do ended up being uncoordinated, sloppy, or unnatural; and in most cases, all three. I had nearly given up. I sat at my computer with my head resting in my hand, clicking around on the desktop. I opened a window. I closed another. I navigated around. Then it hit me. Right there. Use the *Finder* as an interface!

From there ideas poured down like rain. Open a stack! Open a resource! List view! Speakable! Post-modern look! Everything seemed like a doable dream. Integration, it can be done! Share code! Share an editor! Share resources! Share power! I saw lines of code forming in my mind, and not long afterward on my screen. Version 4.0 was growing lightning speed.

It was finished and declared internal beta on the day that I wrote this text, and I am sending it to fellow co-workers for testing. I have yet to find bugs, though. It was created on Monday, May 27, 1996 and finished on Monday, June 24, 1996. Twenty-eight days, 99% devoted to **ExtractIt! for Stacks**. Four weeks, that's all it took. It was the fastest a development team has ever gone under my eyes. I am exhausted, and taking a break by writing this (if writing this is taking a break, you can imagine what work must have been like).

Then there's the future. We could add what Aladdin did—true Finder integration. That would be a good idea. Also, of course...more integration.

It makes me feel happy to look back to my early days, my programs' early days, and the longest-living one of all with the highest version number: **ExtractIt! for Stacks**.

## Version History

---

- 1.0:** Bare-bones copy of stack **Script Changer**.
- 1.1:** Added Mass Extract feature, other minor additions.
- 1.2:** Added more features that were organized into a separate card.
- 1.3:** Added even more features.
- 1.3.1:** Bug fixes.
- 1.4:** Color.
- 1.4.1:** Better color scheme.
- 1.5:** Unreleased version overshadowed by version 2.0. Had everything organized into a TOL.
- 2.0:** Major overhaul. More features, highly organized. Better looking altogether. Faster, more efficient, starting to get a post-modern look. Started from the ground up.
- 3.0:** Another overhaul. Doubled features. Fixed bugs.
- 3.0.1:** More features, some bug fixes.
- 4.0:** *Another* overhaul. Started from the ground up *again*. A “better, more integrated interface to make your work more streamlined.”

# About the Manual

---

Its Rather Boring Life, in Contrast

## Reading the Manual

---

Obviously you can read this if you are reading this text (unless you opened it with a data fork editor and are trying to decode the text...if you are, just give it up and call our tech support line). Anyway, the manual is in Microsoft **Word** 6.0 and up format. Also, there should be a translated version compatible with Microsoft **Word** 5.0 and up.

We included a version 5.0 document because not all programs support **Word** 6 translation, but most support 5.0 or 5.1.

## Why Word?

---

No offense to Microsoft (I'm sure the some 30,000 development teams worked hard on it), but there probably are people out there to *hate* **Word**. Of course, there are probably people who *hate* **ExtractIt! for Stacks**, too. If you do, feel free not to pay (unless you *like* us, and want more cool programs), and punch the screen right here.

I personally think **Word** 6.0.1's a bit on the slow side for my '040 (those of you with Power Mac 8500s and 9500s, *stop chuckling*), but it does have a lot of features that most other word processors still need. I am still waiting for Quick-Draw GX support.

If you don't like **Word**, feel free to translate the 6/5 version to **ClarisWorks**, **WordPerfect**, **WorldWrite**, **MacWrite**, **FullWrite**, or any other word processor who's title is probably LikeThis.

## What Happened to DOCMaker?

---

I'm sorry, **DOCMaker** fans, but unless **Green Mountain** Software can find some way to import some form of styled text (or I can make my own C++ document reader), we will just be stuck with **Word**. However, I am considering buying **Adobe Acrobat Distiller** so I can create PDF manuals.

It's just getting too hard to copy each of the very many chapters into **DOCMaker**, and reformat them.

## If You *Still* Can't Read it...

---

If you *still* can't read the manual, we can translate it into various formats. However, because the translators don't always copy *all* the supported formatting, your document may be a little "stripped" of some formatting.

Here are all the formats we can give it to you in:

- Σ Claris **ClarisWorks** 2.0
- Σ Claris **ClarisWorks** 3.0
- Σ Claris **ClarisWorks** 4.0
- Σ Microsoft **Word** 4.0

- Σ Microsoft **Word** 5.1\*
- Σ Rich Text Format (RTF)
- Σ Plain Text\*
- Σ **SimpleText**
- Σ **PowerPlant** Text Traits Format (Txtr)
- Σ RFT-DCA
- Σ Corel **WordPerfect** 5.0
- Σ Corel **WordPerfect** 5.1\*
- Σ \*Indicates both Macintosh and PC compatible formats possible. But why would you be using a PC?

If you *still* can't read it, you need to slowly back off and walk away from your treasured Altair and step into the Apple Company Store.

We will promptly ship you (by disk or e-mail file attachment) a translated document.



# Important Information

---

The Stuff You Will Actually Need to Know

# Registration

---

Painfully, I Must Say it is No More

## Introduction

---

OK, by now you've probably launched **ExtractIt! for Stacks**. You've probably even hacked a few stacks. Then it hit you. This *isn't a demo!* You jump up and down in pure joy, thinking we may have made a mistake on uploading. But you're not going to tell us. No, you are not the kind of person to go up with those "Bambie" eyes and say, "Excuse me, but you—by accident—uploaded the full copy of this program and everyone's getting it for free." No way.

Well, to tell you the truth, **ExtractIt! for Stacks** is free now. Yes, you have unrestricted total stack-hacking at no cost whatsoever. No gimmicks, no catches, no tricks.

However, **ExtractIt! for Stacks** is not necessarily freeware. Nor is it shareware. It falls between the two. I like to call it *payware*. If you like it, pay us a small donation. You are not obligated to send us money, and you will receive no benefits from it. However, the money will go into future copies of this program.

## Why is it Free Now?

---

OK, now I will take you behind the scenes on the previous prices. The only reason version 3.0 cost money was because ResCopy cost money. ResCopy cost \$25 annually. So we decided to charge \$13 for **ExtractIt! for Stacks**. Once the first 2 people register, we send in the \$25, then we give those people their full copies, with a \$1 profit.

Then, until the next year, we make a \$13 profit for each stack sold because we only have to pay annually. When nobody registers for a year, we discontinue the stack.

However, this new **ExtractIt! for Stacks** uses our own integrated method of copying resources, rolled into an editor, viewer, and browser. It doesn't use ResCopy, we have no expenses, and we don't need to charge money for it.

Of course, we could do what Netscape Communications did. We could make it free on the Internet, and charge money for it on a catalog to pay for the catalog's fee (see for yourself—the navigator is free from <http://www.netscape.com/>, but it costs \$49 off of catalogs). But we're nice.

# General Operation

---

How to Get the Darn Thing to Work

## Using “Windows”

---

All of the browsers are window-based (the object in the Finder, not the operating system). You cannot move a window via its title bar. You cannot Command-click the title bar to show a pop-up of the hierarchy. You cannot click on the column names to re-sort a window.

However, there are things you can do. You can select objects, speak them, and get info on them. You can edit them, too. You can go to the top level, close, open, save, and quit.

Most all windows have a close box, . It looks like a normal close box. Clicking it will bring you back one level in the case of browsers, or finish editing whatever you are editing (asking to save if saving is possible) in the editor.

Most windows also have an export box, . Clicking this in a browser will give you a dialog allowing you to chose which field to export, and then a Save File dialog box asking you to export it as a file. It then exports the field as a SimpleText or TeachText file. Clicking it in the editor prompts you to export the main window with a Save File dialog. In then exports it to a TeachText or SimpleText file.

The locking browser has an unlock box, . Clicking it will ask you if you want to use the UnProtect XCMD or the DeProtect XCMD to unlock the stack. Choose, and it will use that XCMD on the selected stack. Both externals remove password protection from a stack, but just in case you had a personal need or preference, we included 'em both.

The editor has a speak box, . It will speak whatever is in the editor window using PlainTalk 1.3 or higher, using the default voice.

The editor also has a format box, . It will format and indent the editor window as if it was a HyperTalk script. It will not report errors, though.

Finally, the editor has a run box, . It treats the window as a script and runs it. You don't need to enclose the script in on's and end's (a little offspring of odds and ends, if you get the joke).

Clicking on the title, and all of the header fields will speak those fields, too. Sorry if it speaks text like “27 line(s)” and your computer says “twenty-seven line...ess.” I will probably incorporate code to differentiate between plural and singular forms later, but not now.

## Using the TOL

---

It's not that hard. Click on an item to select it, or use the arrow keys. Double-click it, hit return, or click **Go** to go to the selected item. Click **Quit** to quit. Click on **Time and Date** to show or hide the current time and date. Click on **Speak** to speak the selected item.

Clicking on the description field will speak the description for the currently selected item. Clicking on the time and date field will speak the time and date.

## When it's Too Big

This applies to all extractions that return data to a field, but the most occurrences will probably occur in the MassExtract browser. If the data returned exceeds 30,000 characters (the maximum a field can store), you will be presented with a dialog asking you what to do with the data.

Clicking **I Guess** will let you save the data to a text file, and then ask you if you want to display the first 30,000 characters, *too*. Clicking **Forget It** cancels the operation. Clicking **Show Part** shows the first 30,000 characters, with no way to retrieve the rest (except redoing the MassExtract).

# Operation

---

How to Work This Thing

# Stack Browser

---

The Quickest Way to Modify or Ruin Your Stacks

## Introduction

---

The stack browser was the very first thing we created (obviously). Even before the animation screens, even before the TOL. We were just testing it out to see if browsers would work. And it did, quite well.

If you look at the scripts of the stack, card, and background, you'll notice they creep *very* close to the limit of 30,000 characters. That's because we need to put in a description, name, and way to extract every property, object, and resource available!

## Reading or Writing Data

---

It's extremely easy to edit anything in the stack browser. Before you can do anything, you must first select a stack. Do this by pressing the **Choose Stack...** button. Select your stack via the Open File dialog that will appear. Its pathway should appear in the field below.

A list of all stack attributes and the words "Cards" and "Backgrounds" should appear in the main "window." If you want to edit a stack attribute, just double-click it, or select it and hit return (from this point on, we will refer to it as simply "opening it"). You will be transported to the editor where you may view and/or edit the attribute (to learn how to use the editor, go to that chapter).

Clicking on the "Cards" item will list all the cards. Clicking on the "Backgrounds" item will list all the backgrounds. From there you can select a card/background, and open it to list its properties. There you can edit the card/background, or go even further as to open "Buttons" or "Fields." It will then list all the buttons or fields of that card/background.

Opening a button/field will show its properties. That is the deepest you can dig.

To write data, just close the editor window! Read about the editor for more information.

# System Browser

---

All the Stuff About Your Computer the Dealer Tried to Hide

## Introduction

---

As it was originally, the system and file browser were going to be totally integrated. You would start out with two options, Disk and System. However, this got to be too confusing and the scripts got so tangled up that we had to separate them.

Now, the system browser is its own section, and there is many options for finding out what you can or cannot do on your computer.

## Operation

---

It's really simple. You are presented with a list of properties about your system, and their HyperTalk equivalents. Just open a property to view/edit it in the editor. Some are read-only.

# File Browser

---

Why Your Free Disk Space Can be Measured in Bytes

## Introduction

---

After I made the stack browser, I made the title screen and the TOL, and then this. It was originally going to be integrated with the system browser, but it go to be too hard. I then made each browser sequentially afterward. The file browser was number two. It wasn't quite as big as the stack browser, but it was definitely the hardest thing to do.

Every single thing I did had a bug in it. Glitch here, bug there, it looks easy enough to fix. But it wasn't *Where is the stupid bug?! I can see it onscreen but it's not in the code!* I finally got it right, after a *long* time.

## Navigating the Browser

---

You will first be displayed, in the window, a list of all mounted volumes. Opening one will show its contents. Note that you may see some items you didn't think were in the disk you opened. This is because some files/folders are invisible—the Finder doesn't show them. But the file browser does. They are no different from any ordinary file or folder (but don't play with them—they are invisible for a reason).

Just point-and-click your way to the item you want to modify. If it is a folder or a disk, just select it and press **Get Info**. If it is a file, open it. Choose the property you want to look at or modify, and open it. You will again be transported to the editor.

# Resource Browser

The Only Way to Ruin Your Programs

## Introduction

Originally, the resource browser was going to be a lot of separate parts. But, in the rule of integration, we totally wound them up so you have an all-in-one browser, viewer, editor, and copier for resources. It works pretty quickly, too.

## Operation

First, select a file whose resource fork you wish to edit by clicking the **Choose File...** button. Its pathway will appear on the field below. Then, it will go through the resource list and delete any unneeded lines. *This does not restrict you from certain resources.*

Selecting a file generates a list of resource types. But the same resource type is repeated for each resource *within the type*. If you have a file with five PICTS and three STR#S, it will return a list with five occurrences of the word "PICT" and three occurrences of the word "STR#." Therefore, **ExtractIt! for Stacks** must go through the script and delete any repeated lines so the result is a nice list of each type of resource, with no recurrences.

Once you have found your resource type, open it. A list of all the resources of that type will appear. Alongside each resource name will be its ID, size in bytes, and attributes.

Its attributes will be a list of letters. Each letter corresponds to an attribute set to be *true*. All the attributes set to be true will be shown. The letters are as follows:

Letter	Description
C	Changed
P	Preload
S	System Heap
U	Purgeable
L	Locked

If your resource has the attributes "PSL" then it is preload, in the system heap, and locked. Nothing means none of the attributes are true.

Once you have your resource selected, you can do many things with it, as follows below.

## Editing a Resource

Opening a resource will take you to the editor where you may edit the resource. You can view it, but not save it. If you come across a number in brackets, it could be one of two things. It could be part of the resource, or it could be a representation of a character of that ASCII code.

If the number in brackets is not part of the resource, then it is in place of a character, with an ASCII ID of the number. If you come across a [0], then a null character is there (the null character has an ASCII ID of 0).

## Copying a Resource

Selecting a resource and clicking **Copy...** will copy it. You will be presented with an open file dialog box where you may select the file to copy the selected resource to. It will then copy the resource.

If a resource with the same ID already exists, you can choose to abort copying, renumber the resource to be copied, or replace the existing resource.

Replacing will put the resource to be copied in place of the identical resource, deleting the old one forever. Clicking **Renumber** will let you choose between two modes of renumbering, sequential or random. Random will assign the resource to be copied a random ID. Sequential will assign the resource the next number up from the highest ID already in that file of that resource type. Either way, this does not affect the ID of the resource in the file it was copied from, just the ID of the duplicate resource.

# Script Browser

For the Not-So-Advanced HyperTalkers

## Introduction

The script browser was the hardest and longest thing to. It is a simple, straightforward way to find out how HyperTalk works, get reference on HyperTalk commands, functions, etc., check out code, and make your program smaller, better, and cooler.

Note that some of the sample scripts are stored in the Preferences file, so please don't modify that file. It is deathly important or the scripts will be messed up.

## Main Sample Scripts

The main sample scripts are the first 5 or so items on the browser. It's really simple to view them. Double-click them, and the editor will show you the script. Close the editor to go back to the browser.

## Useful Scripts

OK, OK. This is a total and complete import from the **Useful Scripts** section of version 3.0. Oh well, that's that. Here's how to use the features.

### Basic Buttons

These are—you guessed it—basic buttons. You normally wouldn't want to click on them—the scripts need a bit of customization. Just look at the scripts to see how it is done.

### Informational Inquiries

These are buttons and fields that do out-of-the-ordinary things. The **Help:** pop-up button will, when clicked, display a step-by-step procedure on how to extract a script or attribute of a stack. Selecting one of the steps will display a detailed description of the step.

The **About** button, when clicked, will show a field about this program. Click on the field to hide it.

The **Copyright...** button will, when clicked normally, display a copyright notice. Option-clicking the button displays my personal grunt message towards this stack.

### Forgotten Features

This is a bunch of features I bet ten to one you aren't going to need. But the scripts are sort of interesting.

The **Really Random?** button evaluates a theory stating that, if a random number of 2 possible is generated an infinite amount of times, there will be a 1:1

ratio of the 2 numbers. Because of the imperceptible size of  $\infty$ , we decided to go with 100 times. Click on it to see.

The **Estimating  $e$ ...** button will run an estimate of the constant  $e$ , approximately equal to 2.71828.  $e$  is a non-terminating constant, and the further you carry out the formula, the more digits you get. We carried it out tenfold.

The two blank buttons, when clicked, will first contract and expand the one clicked, and then do the same to the other blank button. Just try it.

The **Speech Tunes** button will make PlainTalk sing a crummy impression of the tune on that catfood commercial. Check out the scripts to see how you can make PlainTalk sing, dance, pause, and scream.

The **Other Oddities** section we'll let you figure out how to work. Just click around.

## HyperTalk Reference

If you downloaded **ExtractIt! for Stacks 3.0**, you've probably realized that its HyperTalk reference is just the AppleScript event resource dictionary for **HyperCard**. Now, however, you are in for a big surprise. This is why **ExtractIt! for Stacks** has 419 cards: very close to 90% of them are devoted to HyperTalk reference. That's right. We made a card for every object, property, command, function, operator, system message, expression, and everything else listed in the manual.

It is *really* easy to use. Just click on any of the items in the list field on the left of the card to get information on that item. Typing in a string and clicking **Search** will go through each and every card and search for that string. When it comes across a card with that string as its name, it will stop there. Pressing option will continue searching, and clicking the mouse button will stop the search there.

# Massive Extractions Browser

---

With Massive Result Strings

## Introduction

---

I've always been nagged by my fellow co-workers to create a "custom" massive extractor. I even made it a will-be-here-soon feature. But I never had the time to put a few radio buttons in. Until now. Now there is peace at [Lightning Bolt](#) about custom MassExtractions.

## Operation

---

To use, just follow the outlined steps below. Doing otherwise may result in 5 years imprisonment and a \$50,000 fine (just kidding).

First, select your stack with the—you guessed it—**Choose Stack...** button. All the usual will happen. Also, a list of MassExtract options will appear in the browser.

Next, select what exactly you wish to MassExtract. By default, all will be false. To include something, select it and press **Add**. To remove something from the list, select it and press **Remove**. Once you are done, click **MassExtract...** to perform the mass extraction. The result will be shown in the editor.

Clicking **Top Level** will close the stack, as does the close box.

# Locking Browser

Make Sure Those “Less Powerful” Stack-Hackers Aren’t a Problem

## Introduction

**ExtractIt! for Stacks 3.0** had a locking analysis, stack unlocker, environment unlocker, stack protection, and the whole enchilada. Now, we’ve rolled it into one big card that makes it really easy to get or set environment or stack locking.

## Environment

At the top level, opening “Environment” will show a list of environment options and will allow you to set them by clicking the **Lock/Unlock** button. The **Lock/Unlock** button will do various things for various items. Here is what it will do for the environment items:

Name	Action
User Level	Sets the user level to 5.
Menu Bar	Resets the menu bar to its default state.
User Modify	Sets the userModify function to the opposite of its current setting.
Menu Bar Shown	Sets the visibility of the menu bar to the complement.
Title Bar Shown	Does the same as above for the title bar.
Screen	Sets the lockScreen property to its complement.
Messages	Sets the lockMessages property to its complement.
Recent	Sets the lockRecent property to its complement.
Error Dialogs	Sets the lockErrorDialogs property to its complement.
Tool	Chooses the browse tool (  )

Note that it may take a *long* time to reset the menu bar; this is not our fault, it seems **HyperCard** is really slow at that.

## Stack

To get or set locking properties of stacks, you must first select your stack with the **Choose Stack...** button. All the usual will happen. Then, at the top level, you may open “Stack.” That will display three locking-related properties of the stack, plus “Cards” and “Backgrounds.” Opening “Cards” or “Backgrounds” will list all of that object plus their status (locked or unlocked). For cards and backgrounds, the status is determined by the `cantDelete` property.

# Message Browser

---

How Much Your Little Brother Has Been Snooping Around Your Drive

## Introduction

---

The message browser is a straightforward message recorder and sampler. No catches, no gimmicks, no cheesy demo-like tricks. It is really simple and taken directly from the idea behind **ExtractIt! for Stacks 3.0**.

## Sending a Message

---

To send a message, just select the appropriate message in the main browser and click **Send...** You will be prompted to enter any necessary parameters for that message. Chances are, you will need none. To be sure, you can get reference on the message, covered later. Once you are done, it will ask you to enter a HyperTalk descriptor pointing to an object to send it to. If you want to send it to a card named "Last One," the object descriptor would be card "Last One". Click **OK** and it is sent! The HyperTalk debugger will notify you if something's wrong.

## Getting Reference

---

You can directly link to the reference description of the selected message in the [HyperCard Reference](#) section by clicking **Get Reference**. There isn't much more you'll need to know about that.

## Recording Messages

---

Alright! We admit it! We just copied the card from **ExtractIt! for Stacks 3.0** and deleted one button! Guilty! Hey, I mean, it's our work, and we don't have to worry about copyright infringement, so why not? It does what you want it to, doesn't it?

In case you weren't here for version 3.0, here's how it works. Start recording messages by clicking **Start**. The button's name will change to **Stop**, and every nuance on your computer will be recorded along with the exact time and date it happened. Everything from mouse moves to stack suspending. Click **Stop** when you can't take it anymore. **Export...**, **Clear...**, and **Print...** do exactly as you would expect.

# Mathematics

---

Logarithms, Algorithms, Factors, Protractors, Hectares...Who Cares?

## Introduction

---

Yeah, we know you loved the **Bolt Scientific Calculator** BL-117A. By the way, “**Bolt**” is the prefix we will use when (or more likely, if) we ever make hardware. We know you loved to press its little buttons and come out with an actual *answer*. Well, we found that to be “impractical, bloated, insensitive to integration, and bulky,” as a co-worker once stated.

So we compromised. Not so restricted, more powerful, and, of course, *integrated*. Now the calculator and numerical conversions are fully and seamlessly integrated. This way, you and your work can get done quicker. But enough with the sales pitch—here’s how to use it.

## Basic Mathematics

---

To do a basic math operation, just type in the operation (in HyperTalk) in the **Formula** field and close it (i.e., hit Enter). The result will be displayed in the **Result** field. Pretty simple.

## Converting Numbers

---

Before you start clicking away at the conversion fields (which you probably have already done), note that it does *not* convert the **Formula** field, but the **Result** field. Whatever’s in the result field is what will be converted. Because you can’t type in the result field, clicking **Copy Literal** will copy the exact text of the formula field into the result field.

The result will be displayed in the result field, of course.

## All the Rest

---

Clicking **Clear...** will clear the result field after you click **OK** in the prompt. Clicking **Copy Result** will put the result field into the clipboard. Pretty straightforward.

And if you don’t know what the return arrow means, you need to get some professional GUI assistance.

# Editor

---

Where We Always Shamelessly Refer You to

## Introduction

---

The editor is, if anything, the best example of integration throughout the entire stack. In fact, you could almost say it's OpenDoc-like. In **ExtractIt! for Stacks 3.0**, each card had its own little field for reading, editing, and writing data. Now, in version 4.0, every feature shares a universal editor totally compatible with all the cards.

The editor, when gone to, will automatically figure out the pending task and complete it (i.e., edit a script, open a resource, etc.). If there is no pending task, it will act like a bare-bones text editor.

## Reading Data

---

It should happen automatically. Once it does, information on the data will appear in the header, and a description below that. Even if the data is read-only, you will still be able to edit it. Saving it is where the truth comes out.

## Saving Data

---

Once you are done totally wreaking havoc through whatever you're editing, just click the close box. If the data is read only, it will just take you back to your starting point. Otherwise, it'll ask you if you want to save changes (whether you made changes or not), and you can tell it to or not to. The appropriate steps will be taken to ensure the right data goes to the right place in the right way.

# Object Counter

---

The True Test of a Stack's Worthiness

## Introduction

---

This is basically your version 3.0's object counter made into a browser. However, note that turning something into a browser means more flexibility. First of all, now you can choose what you want to count. Also, you have more options.

## Counting Objects

---

To count a set of objects, first select your stack with the **Choose Stack...** button. Then just select the appropriate item and click **Count**. Note that no item requires you count another item beforehand. They can each operate fine in solitude. The object class will be counted.

Closing the window or going to the top level clears the acquired information.

# Preferences

---

A Little Voice in the Senate Here...

## Introduction

---

This is where you get to personalize **ExtractIt! for Stacks**. And I mean literally, you can actually personalize your copy of **ExtractIt! for Stacks**. That way, we know exactly who is stamping their names on our software. And you get a bit of glory too—not paying for your name on our stack.

Actually, to be honest with you, you were never meant to be able to personalize it. But when I wrote the first sentence of this chapter, I decided that that was a good idea. So I stopped writing, opened **HyperCard**, and did a little modification before getting back to this manual.

## Setting Preferences

---

To set a preference, select it and click **Set**. If the preference is a Boolean value (true or false), it will switch to the opposite value. Otherwise, it will prompt you on what to set it to.

The Clean Environment preference, when true, will reset the menu bar, show the title bar, show the menu bar, etc., when **ExtractIt! for Stacks** is launched. The Show Animation preference will let you set whether or not you want to see our logo's animation.

The Go to a Card preference will, on startup, automatically go to a card. You can set the card name below. The Startup Sounds preference will let you set whether or not you want the startup sounds (thunder and robotic bang) to play.

To personalize your copy of **ExtractIt! for Stacks**, just click the **Personalize...** button. It will ask you for your name and organization (both optional), and then put them in the TOL.

# Hints and Tips

---

Those Little Things You Always Want to Try Out But Never Remember Them

## Introduction

---

Now, if anything in this stack is straightforward, it's this section. Two steps: read, close. Is that not hard? Anyway, this section tells you those little hidden goodies that just might streamline your stack-hacking. Note that some of the stuff is an Easter egg spoiler.

# About This Stack

---

Everything You Didn't Need to Know About it

## Introduction

---

This section is a musical-score, full-blown multimedia presentation—without the color. If you time your mouse-clicking right, you can get the music to grow along with the story.

## About the Theme Song

---

The theme song is Rescue! II. I created it myself. It was originally just going to be a sequel to an earlier song I created, Rescue!, but it turned out to work pretty well with the **ExtractIt! for Stacks** presentation. It took me two hours to compose on the piano, plus a day to port it to my Mac using **PlayerPRO** 4.5.1.



# Appendices

---

Stuff You Really Don't Care About

# Glossary

*n.* A List of Specialized Words and Their Corresponding Definitions, Placed at the Back of a Book

What We Say	What We Hopefully Mean
<b>background</b>	The object layer that underlies card graphics and objects.
<b>button</b>	A clickable object with a name.
<b>card</b>	The object layer foremost to the stack.
<b>chunk expression</b>	Any expression referring to text, such as “word 1 to 2 of line 5 of card field 3.”
<b>container</b>	Any object which can store information, such as a button or field.
<b>descriptor</b>	An expression which <i>eventually</i> ends up referring to an object, like “card button 3 of card 5.”
<b>extract</b>	To retrieve for further editing.
ExtractIt! for Stacks	Hmm...I'll have to think about that one.
<b>field</b>	An object meant to store TextEdit records.
HyperCard	If you don't know that, why did you download this?
<b>Lightning Bolt</b>	Uhh...I used to know.
<b>object</b>	A stack, card, background, button, or field.
<b>pointer</b>	Anything that directs a message to something else (like telling <b>HyperCard</b> to perform the text of a field, that text telling <b>HyperCard</b> to click on a button).
<b>prompt</b>	Something that asks a user for information.
<b>property</b>	A sometimes editable attribute of an object, such as size.
<b>script</b>	The name of the game here.
<b>stack</b>	Something really cool if you work at it all night long.
<b>system message</b>	Anything passed by a script or the Finder.
<b>TOL</b>	The Table of Linkages, the interactive table of contents.
<b>variable</b>	A container existing only in memory.

# Legalese

---

Keep the Feds Off Our Back

## Colophon

---

As if you really care. Heading font is ITC Palatino and ITC Palatino Bold 24. Header comments are ITC Helvetica 10. Auxiliary headers are ITC Helvetica Bold 18. Text font is ITC Times 12. Some text may appear in Monaco 9. In stereo where available.

Proof pages were printed on a PDD Maker, and a PostScript-based Personal LaserWriter NT. Final pages are up to you to print.

## Trademarks

---

Macintosh<sup>®</sup>, LaserWriter<sup>®</sup>, PlainTalk<sup>®</sup>, HyperCard<sup>®</sup>, and Apple<sup>®</sup> are registered trademarks of Apple Computer, Inc. All rights reserved. Times, Helvetica<sup>®</sup>, Palatino, PostScript, and Adobe are trademarks of Adobe Systems, Inc., and in certain jurisdictions may be considered registered. All rights reserved. Microsoft<sup>®</sup> and Word<sup>®</sup> are registered trademarks of Microsoft Corp. All rights reserved.

# Credits

---

As if Anyone Helped Me

## Stack Credits

---

This stack took a lot of people to turn it into what you are seeing now (which is probably *nothing* like my glamorous visualizations). I'd like to mention a lot of people (either because they helped or they want their name in lights). I'll start with Chern Lee.

Chern and I spent many, long, boring hours clicking and clicking again. We would do everything conceivable that a user would do, find a bug, attempt to fix it, and do that over and over again. More than half of the development time was spent fixing, as Matt Burch puts it, "obstinate bugs that just won't die."

Next up is Susan Morgan. She read over my manual, looked over my stack, and made sure I stayed out of jail, passed "Go," and collected \$200. She also praised me for my tiny nuances of an accomplishment.

Third to go is Michael Morgan. Despite the fact that he didn't know the first thing about HyperTalk, he managed to help me. He provided the C code for the bouncing balls, and left it to me to turn it into HyperTalk. And though he has a hard time showing it, we all know he's at least a bit proud of me.

We mustn't forget good old Frederic Rinaldi and his ever-growing masses of tiny -oids and Xternals. His piles upon piles of tiny XCMDs accumulated to make **ExtractIt! for Stacks** quick and powerful.

Of course, I owe it all to the 1990 Apple resort team in the Swiss Alps. Their stack, **Script Changer**, made **ExtractIt! for Stacks** a reality. Had **Script Changer** never been made, there would be no **ExtractIt! for Stacks**.

Speaking of Apple, that is one company I will never forget. I thank them for the Macintosh computers, still on top of them all. Also for my PlainTalk microphone, which I dropped on various surfaces to get the new, improved ball sounds. And for their program **HyperCard**, which is truly unique. But when will it have *real* color (hopefully in version 3.0)?

Next comes the itty-bitty external HexToDec by Itty Bitty Software. It plays a not-so-itty-bitty part in this stack. And Jon Pugh, we mustn't forget. I spent more time trying to find out who made his SysEnvirons XCMD than I did finding out which ways I can use it.

I also want to thank Anup Murarka and Eric Carlson for the VolumePath XCMD (are they a part of Apple—I don't know...).

Finally, I want to thank Jim Ratliff who created the SillyconValley font, because I like it...and I can't yell that out loud in cyberspace (one of the requirements is to yell out "I love it!" or something like that).

Nobody brought in bug reports, so therefore I can't include anyone's name on the beta testers' list. However, the beta had no bugs! Unfortunately, though, we were unable to get the music to work. More is in the read me file.

## Manual Credits

---

First of all, I'd like to thank Susan Morgan for making sure I spelled everything ccorekly. She and I also spent many hours arguing over the page format ( I still say chapter titles should be printed in a page of their own, but the *sections within the chapters* should share the page with the section text ☺).

Also, Microsoft. I want to thank them for the feature-packed **Word 6.0.1** for the Mac, even if it is the last Mac version.

# Index

---

Lost Again, Are We Not?

- A—
- About This Stack, 32
    - theme song, 32
  - Appendices, 33
- C—
- Credits, 36
    - manual, 36
    - stack, 36
- E—
- Editor, 28
    - reading data, 28
    - saving data, 28
  - ExtractIt! for Stacks**
    - about, 7
    - history, 7
    - version history, 9
- F—
- File Browser, 19
    - navigation, 19
- G—
- General Operation, 14
    - data exceeding max. capacity, 15
    - using the TOL, 14
    - using windows, 14
  - Glossary, 34
- H—
- Hints and Tips, 31
- I—
- Index, 37
- L—
- Legalese, 35
    - colophon, 35
    - trademarks, 35
  - [Lightning Bolt](#), 6
    - about, 6
    - customer relations, 6
    - customer support, 6
    - technical support, 6
  - Locking Browser, 25
    - environment-related, 25
    - name/action table, 25
    - stack-related, 25
- M—
- Manual, 10
    - translation, 10
  - Massive Extractions Browser, 24
    - legal punishment for not following directions, 24
    - operation, 24
  - Mathematics, 27
    - basic math, 27
    - numeric conversions, 27
    - sales pitch, 27
  - Message Browser, 26
    - getting reference on messages, 26
    - recording messages, 26
    - sending messages, 26
- O—
- Object Counter, 29
    - counting objects, 29
- P—
- Preferences, 30
    - personalizing your stack, 30
    - setting, 30
- R—
- Registration, 13
    - why it's free, 13
  - Resource Browser, 20
    - copying resources, 21
    - editing resources, 20
    - operation, 20
    - understanding attribute letters, 20
- S—
- Script Browser, 22
    - basic buttons, 22
    - forgotten features, 22
    - HyperTalk reference, 23
    - information inquiries, 22
    - other oddities, 23
    - sample scripts, 22

Useful Scripts, 22  
Stack Browser, 17  
  editing data, 17  
System Browser, 18  
  operation, 18

—T—

Touchy-Feely Statement, 38

---

*“What is easy to include yet hard to retrieve is a gem towards making a stack perfect.”*

*—Timothy Morgan, 1995. Thanks, everyone!*