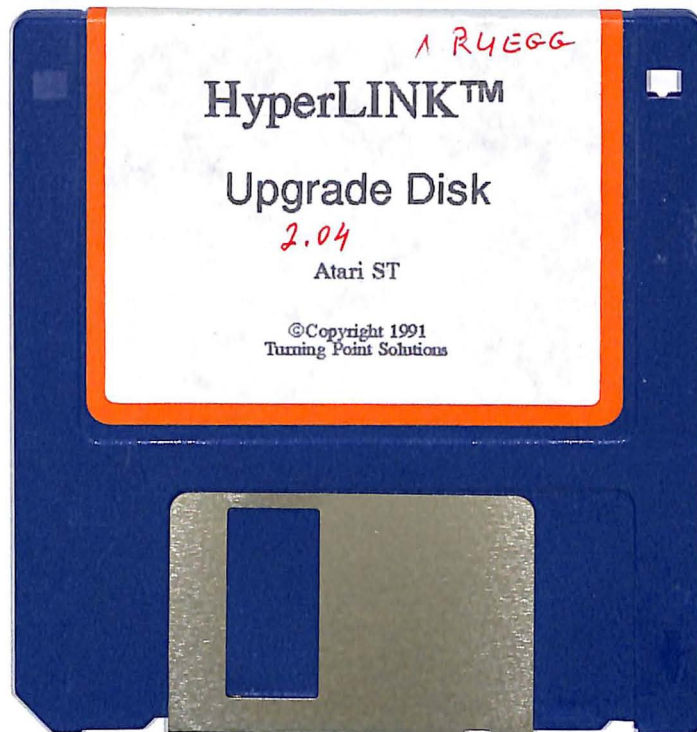


0152 10158

**HYPHER
LINK**

JMGE





HYPERLINK

*Bringing the Power of
Hypertext to your
ST!!!*

Featuring...
Database Control
Relational Objects
Hypertext
Graphics, Sound
Unlimited Expandability

and much more...

Distributed by :
JMG Software International, Inc.
892 Upper James Street
Hamilton, Ont. L9C 3A5



HYPERLINK

TABLE OF CONTENTS

Section 1 : Introduction & Setup

<i>Legal Stuff</i>	2
<i>What is HyperLINK</i>	4
<i>Getting Started with HyperLINK</i>	6
<i>General Info on Sub Directories</i>	8
<i>HyperLINK Modules</i>	9
<i>HyperLINK Applications</i>	9

Section 2 : HyperLINK Anatomy

<i>Title Screen</i>	15
<i>HyperLINK Menu Selections</i>	16
<i>The Mouse Buttons</i>	17
<i>The HyperLINK Screen</i>	21
<i>The Builder</i>	21
<i>Building with Text</i>	23
<i>Building with Data Fields</i>	24
<i>Building with Buttons</i>	25
<i>Building with Icon Buttons</i>	27
<i>Building with Graphic Objects</i>	28
<i>Building with Box Objects</i>	28
<i>Building with Multi Boxes</i>	29
<i>The Visual Screen</i>	29
<i>The Expression Screen</i>	30
<i>The Linking Screens</i>	
<i>Link to a HAP</i>	30
<i>Link to a HAP & Search</i>	31
<i>Link to a HAP & Pass Data</i>	31
<i>Command Links</i>	32
<i>The Application Info Screen</i>	32
<i>The Application Defaults Screen</i>	33
<i>The Text Editor</i>	34

CONTENTS

<i>The Text Linking Builder</i>	39
<i>The Report Generator</i>	40
<i>Building Reports with Text</i>	41
<i>Building Reports with Data</i>	42
<i>Report Setup</i>	
<i>Custom Sorts & Searches</i>	43
<i>IMG Graphics Displayer</i>	44
<i>The Animation Player</i>	45
<i>The NUL Module</i>	45

Section 3 : Step By Step

<i>Creating a HyperLINK Application</i>	49
<i>The Baseball Card HAP</i>	49

Section 4 : HyperLINK Reference

<i>The HyperLINK Desktop</i>	
<i>The Title Screen</i>	68
<i>The HyperLINK Desktop</i>	70
<i>HyperLINK Menus</i>	71
<i>The Application Builder</i>	
<i>Parents vs Children</i>	80
<i>The Builder</i>	82
<i>Text Parts</i>	83
<i>Field Parts</i>	85
<i>Button Parts</i>	88
<i>Icon Button Parts</i>	91
<i>Graphic Parts</i>	94
<i>Box Parts</i>	96
<i>Multi Box Parts</i>	97
<i>The Visual Screen</i>	98
<i>The Expression Screen</i>	101
<i>The Linking Screens</i>	102
<i>The Builder Menu</i>	109

HYPERLINK

<i>The Application Info Screen</i>	110
<i>The Application Defaults</i>	116
<i>The Text Editor</i>	120
<i>Text Editor Menus</i>	120
<i>Creating Text Links</i>	127
<i>The Report Builder</i>	129
<i>Report Text Parts</i>	130
<i>Report Data Parts</i>	131
<i>Sorts & Searches</i>	135

APPENDIX A

<i>Linking - A Different Approach</i>	139
<i>A HyperLINK Analogy</i>	140

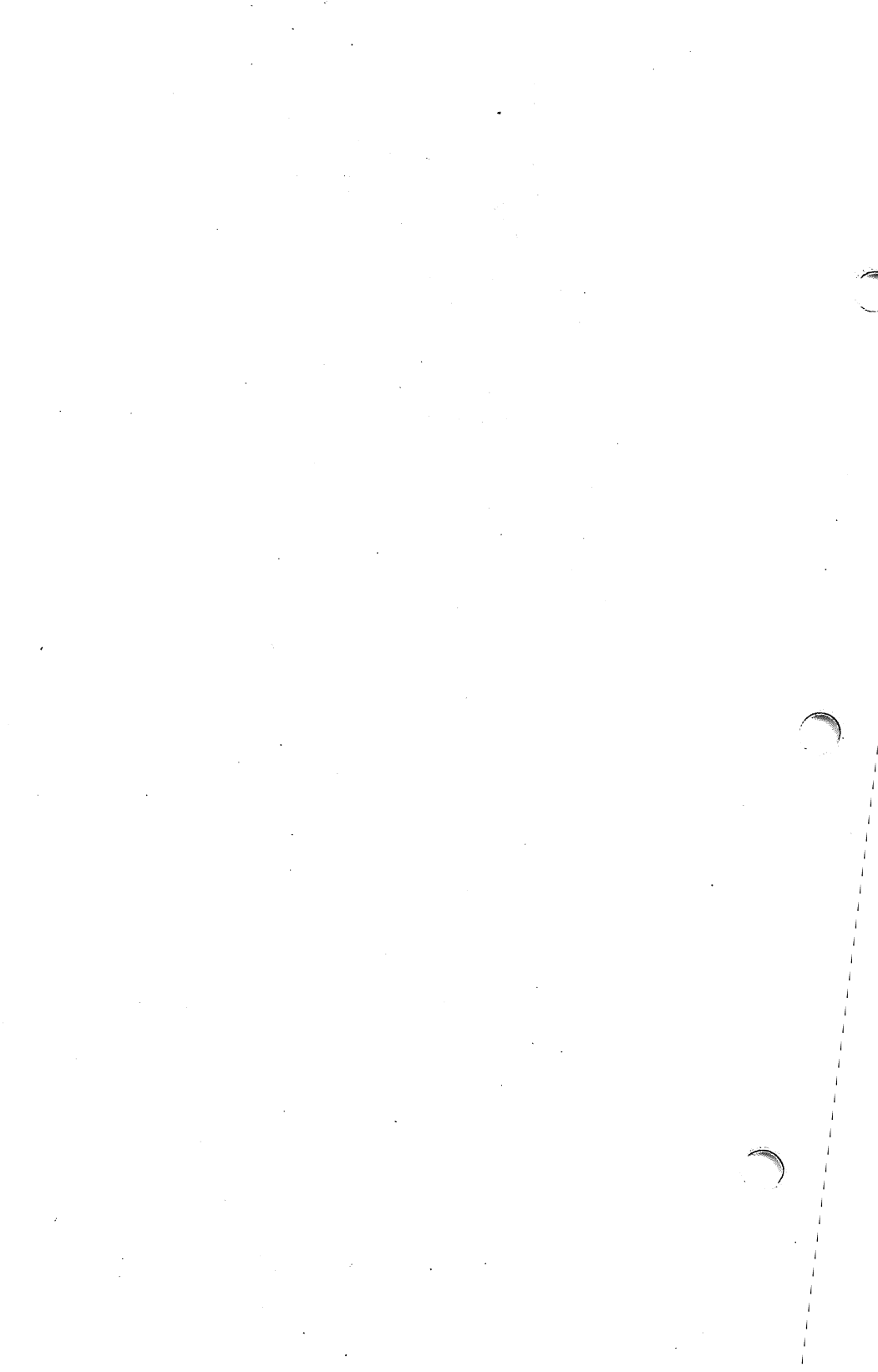
APPENDIX B

<i>Expressions</i>	151
<i>Numeric Operators</i>	153
<i>Character Operators</i>	153
<i>Relational Operators</i>	154
<i>Logical Operators</i>	155
<i>Value Result Expressions</i>	158
<i>Logical Result Expressions</i>	158

APPENDIX C

<i>Command Links</i>	161
<i>List of Commands</i>	162

INDEX	165
--------------	------------





HYPERLINK

Introduction

An introduction and explanation of the concepts and components that make up HyperLINK.

This section also contains suggestions and instructions on setting up HyperLINK on your system.

ONE

Some Legal STuff

HyperLINK is not copy protected, so the first thing you should do is make a backup of the distribution disk. Most software publishers are trusting souls who put their life's work up to public scrutiny for the public's own benefit (and maybe some financial reward). Theoretically programmers are appropriately reimbursed for their efforts by people buying their product if it is found to be worthwhile, and they are penalized by people shunning their software if it is not. Also the Atari ST market is not the most fiscally rewarding software market in the world, and if Atari owners want to see quality software continue to be written for their machines they will PURCHASE AND NOT PIRATE software that serves them well. All copies of HyperLINK contain a Serial Number, and as such we will be able to trace back pirated copies to their registered source. As HyperLINK will be updated continually to registered owners who will be linked to their serial number, it is for your own best interests to get registered with us as soon as possible. We will do our best to track down the source of any pirated copies which may turn up and pursue any and all legal options available to us.

Just a couple of more nasty sentences and then we can get on with the business of using HyperLINK...

HyperLINK is Copyright (C) 1990,91,92 by Turning Point Solutions. All rights are reserved. This document or the associated software may not

HYPERLINK

be copied or reproduced without the express written permission of Turning Point. This manual and the related software are sold "AS IS", without any warranty as to their performance. Turning Point assumes no liability for any loss or damage (personal or mental) to any individual as a result of using this product.

JMG Software warranties for a period of ninety days the media on which this product is carried. If after ninety days a problem should develop with the media, then return it to JMG Software with a check for \$5 to cover handling and shipping charges, and we will replace it.

HyperLINK, HyperEditor, and Hyper-Relative Technology are trademarks of Turning Point Solutions.

We hope that you find HyperLINK to be the productive tool that it was meant to be. And we would like to thank you for purchasing this product. We will do our best to provide many future upgrades and add-ons that make HyperLINK an even more sophisticated and useful product.

What is HyperLINK?

HyperLINK can be many things to many people. It can be looked at as a Relational Database, a Text Editor, and more. We prefer to consider HyperLINK an "Application Generator" capable of allowing a mixture of tools to be used to create your own applications.

HyperLINK is based on the concept of Hypertext which was developed mostly in the 1970's. Hypertext is a system that aspires to replace normal, linear text with smaller text topics "linked" to each other.

HyperLINK applies this system past text though, to encompass data, graphics, sound, and external control of other devices as well, hence the popular term Multi-Media. As an example of this, consider a computer-based encyclopedia open at the entry for the mammal Whales. Imagine on this text based screen, an icon of a speaker. When the user clicks on this icon, a link to a sound module is made and the result is that the computer then plays back a sample of whale song. Further, on that same original text screen, imagine also a camera icon representing a link to graphics. When the user selects this icon the result is the calling up of a picture of a whale.

Linking from one data form to another though is not the only use of HyperLINK. You can also create links from one application to another. Imagine two databases, one a Telephone Message database, and the other an Address Book database. Apart these are both useful but simple

HYPERLINK

database applications. HyperLINK can take them one step beyond : Suppose a record was entered into the Message Database listing a message from a customer, but not giving their telephone

The screenshot displays two overlapping windows. The background window is titled 'MESSAGE' and contains the following text: 'To Earle, Geoff', 'Date 04/30/91 Time', 'From Geczy, George of JHG', 'Phone', and a section with checkboxes for 'Telephoned', 'Returned Call', 'Wants Visit', 'Wants to Come', and 'Please Call / Will Call Again'. Below this is a field for 'Received by GG' and 'Message What's new?'. The foreground window is titled 'ADDRESS BOOK' and contains the following text: 'LAST NAME : Geczy', 'FIRST NAME : George', 'COMPANY : JHG Software Int'l', 'ADDRESS : 892 Upper James St', 'CITY : Hamilton', 'STATE : ONT', 'COUNTRY : Canada', 'POSTCODE : L9C 3A5', 'PHONE : 416-575-1208', 'FAX : 416-575-8283', and 'NOTES : Chief Programmer/Bottlewasher.'.

(Figure 1.1)

number. Through a built-in link, by double-clicking on the name of the customer in the Message database, the Address Book Database would be called up and would automatically go to the record on the customer involved. Not only did HyperLINK call up a different application, it sorted

that database and searched for a matching record. Taken one step further, it would be possible to double-click on the telephone number in the Address Book and have a link created to a module designed for handling the serial port such that the computer would then dial the telephone number.

The power of HyperLINK resides in its ability to do this "Linking" from one application to another, from one type of data to another, and from one HyperLINK Module to another.

Getting Started with HyperLINK

Some knowledge of the Atari ST, and of the GEM environment on the ST will be assumed throughout this manual. If you have problems understanding any specific aspect of the Atari Operating System or HyperLINK itself, please consult your ST owner's manual, your local dealer or user group, ask us on GENie, or just give us a call. We will be more than happy to try and solve any problems you have encountered.

While HyperLINK will run okay from just Floppy Drive Systems, like any database program, the way it operates definitely lends itself better to Hard Drive Systems. If operating HyperLINK from Floppy, remember **NOT TO CHANGE DISKS** while any Application Window is open, as this means a database file is also open. The last thing you want to do is confuse HyperLINK on the path for finding files and maybe end up corrupting your data files as well.

Also, the more memory your computer system has, the more HyperLINK Applications and Modules you can have loaded at any one time. For any reasonable use of HyperLINK you should have at least 1 Megabyte of RAM. If your system has only 512K of RAM you may experience problems creating applications in the Builder and using the Report Generator.

Another note before running HyperLINK is the graphics settings of your computer. HyperLINK works in Medium and High Resolution settings on both the ST and the TT lines of computers.

HYPERLINK

While there is currently no support for displaying colour within HyperLINK, it is an area which we will be further developing, especially using the extended graphics capabilities of the TT.

As we stressed in the "Legal STuff" at the beginning of this manual, HyperLINK is NOT copy protected in any way. This is done for the convenience and safety of the user, so MAKE A BACKUP!!! After you have made a backup of your original HyperLINK diskettes, store them in a safe place and continue with the installation. For floppy users, you really have now finished your installation. For Hard Drive users, create a folder "HYPRLINK" (or anything else you want to call it as the sub directory name is not specific). Once done, copy all the files from your two backup disks into this sub directory. The HyperLINK program, and all the HyperLINK Modules will then be inside this main directory and all the sample HAPS we have included will each be inside their own sub directories. Also there will be two sub directories, ICONS.32 and ICONS.48 which contain a library of icons which you can use in building your own applications.

General Info On Sub Directories

With every HyperLINK Application, all the files associated with that HAP have to be located in the same directory. For instance with the Address Book Database, the HAP itself, the database files, and all indexes you may be using have to be in the SAME DIRECTORY. As well, if you are linking out to any text file, that also has to be in

ONE

Note: All versions of HyperLINK before 1.55 had to have the desk accessory file "HYPREDIT.ACC" loading in to use the text editor within HyperLINK, but this is now no longer the case. If you still have the ACC file loading in, make sure to delete it now as it will conflict with the most recent version of the Editor which now acts as a regular HyperLINK Application.

the same directory. If you are linking out from an Application to another HAP it may or may not have to be in the same sub directory. If the other HyperLINK Application you are going to be linking to is already loaded in, then it does not have to be in the same sub directory but if it is only going to be loaded in to the computer through this "Link", then it must be in the same sub directory as the initial HAP. More of a description about creating a list of "Default HAPs" to load in will be covered later in the manual.

You are essentially ready to dive right in to HyperLINK. Simply by double clicking on the H_LINK.PRG icon you will start up the program.

Good luck and good linking...

Some Important Concepts...

HYPERLINK MODULES

A HyperLINK "Module" is a tool for use by HyperLINK Applications in manipulating or collecting data, or for controlling external devices or actions.

Modules Included :

- Text Editor
- Database
- Application Builder
- Report Generator
- IMG Graphics Displayer
- NUL (for "Interactive Screens")
- Animation Player
- (Under Development)
- Phone Dialer
- DMA Sound Player
- Text to Speech Synthesis Player

While the listed modules are all that are included with HyperLINK at this time, many additional modules are planned for development and with the availability of developer information, third party development of modules is expected to take root.

As you can see, while these Modules are essential to HyperLINK, they are merely the building blocks from which a user can build any desired Applications. The fact that HyperLINK is capable of mixing these modules in any application is what sets it apart from just database software, or just text editing software etc.

HYPERLINK APPLICATIONS

HyperLINK Applications (HAPS) are the end result of what you are creating using the builder

ONE

in HyperLINK. In other words, the actual database and other implementations that the user makes through the various modules in HyperLINK. A HAP can be something as simple as a flat database, or as complex as a mini word processor, relational database, or even a full Inventory / Point of Sale System (mixing a Customer Database / Purchase Order Book / Inventory Control / Customer Order Book / Invoicing / and more to create a complete inter-related system for operating a retail store). The various modules included with HyperLINK are the building blocks used to create whatever HAPs the user desires, from the simple, to the incredibly powerful!

Some HyperLINK Modules also do double duty as HyperLINK Applications. That is, while they provide some of the building blocks to be used in the construction of other HAPs, they also are capable of stand-alone functions. For the sake of this manual, anything that can be used by itself within the HyperLINK shell should be considered an application.

The following listing of Modules illustrates which ones can also be considered Applications in their own right :

MODULE	HAP
HL_EDIT	EDITOR
DATABASE	-----
HL_BUILD	APPLICATION BUILDER
ANIMATE	-----
HL_IMG	GRAPHICS

HYPERLINK

MODULE	HAP (Continued)
HL_SOUND	DSOUND
HL_TALK	TALKER
NUL	-----
REP_GEN	REPORT GENERATOR

(For further technical information on how these Modules and Applications interact, please refer to the HyperLINK Developer's Kit, available separately.)

To help you understand HyperLINK a little better, we have included some sample HAPs with the program to help illustrate how various applications can be created and basically how HyperLINK itself works. Throughout the manual these sample HAPs will be used to show many different aspects of HyperLINK.

HAPs Included : Message Database
Baseball Card Database
HyperLINK Info
(Interactive Text)
Address Database
CANADA : A Hyper Tour
and more...

Note : Some versions of HyperLINK may not have all the applications listed above. If not, the ones you are missing are Public Domain and can probably be found either on GENie or on our own BBS Service. The primary ones we will be referring frequently in this manual to are the first five in our list as these represent a good cross-section of different uses for HyperLINK.

ONE



This Page Intentionally Left Blank





HYPERLINK ANATOMY

A Screen-By-Screen introduction to the many integral parts of HyperLINK.

This section is meant to provide users with a brief summary of what each screen can do and how to do it.

For more detailed explanations of any of the screens, see the Reference Section.



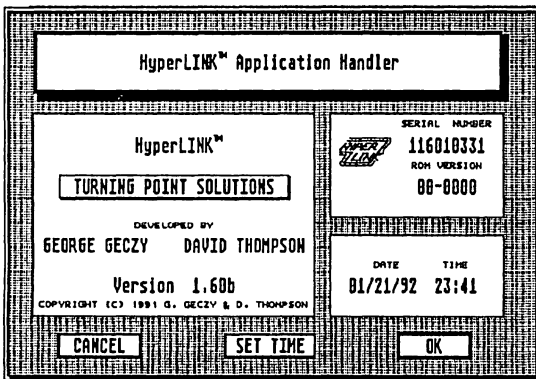
HYPERLINK

THE HYPERLINK ANATOMY

By its very nature, HyperLINK is made of many different parts or sections that on their own don't provide much in the way of power and flexibility. It is the sum of all the parts that make HyperLINK stand on its own as the multi-media solution it is. In this section of the manual, we will give a very brief explanation of all the ingredients that make up HyperLINK and we will try to introduce them in a logical "Order of Appearance" manner.

Further details and more involved explanations of each section of HyperLINK may be found in the REFERENCE Section of this manual and in the Tutorial Section wherein we take you along on the creation of some HyperLINK Applications from first idea, to finished product.

THE TITLE SCREEN : When you first call up HyperLINK, this is the screen you are met with.



(Figure 2.1)

It tells you the version number of the HyperLINK program you are using and also displays your serial number. Further, you can click on the HyperLINK Version Number and up will pop a display of the version numbers of the various functions making up the HyperLINK program. These are merely parts of what make up the whole of HyperLINK.

TWO

Revision Levels			
GEN Utilities	1.30	RSCLIB Loader	1.04
System Init	1.06	Main RSC	1.00
App Vars	1.05	Task Manager	1.00
HyperFORM Init	2.05	Clipboard Init	1.00
lp PRT Driver	1.02	Report Printing	2.00
ACI Interface	1.07	DataBase Ctrl	3.03
DB Commands	3.09	Local DBase	3.05
IPC Handler	1.05		

(Figure 2.2)

Available at any time from us is a listing of what are the most current of these versions.

At the title screen, you are also given the option of updating your current system clock settings. By just hitting <Return>, you accept the current clock settings and immediately go into the program.

THE HYPERLINK MENUS :

After the title screen disappears, you will see messages flashing by on the screen dialogue boxes telling you what HyperLINK Applications are loading in. This automated loading of applications is achieved very easily. When HyperLINK starts up it looks for a text file (in the same directory as the H_LINK.PRG file) called DEFAULT.HAL ("HAL" stands for HyperLINK Application Listing). This file is just a listing of the all the HyperLINK Applications which should be loaded in every time the program is initiated.

NOTE : HALS vs HAPS

HAPs stand for HyperLINK Applications while HALs are HyperLINK Application Listings. These HALs are text files which act as a sort of a macro for loading in more than one application at

HYPERLINK

CANADA.HAL :

```
cdnstart.hap
canada.hap
canada1.hap
cdn_stat.hap
images\symbols.hap
rfld.hap
ns.hap
nb.hap
pei.hap
ont.hap
que.hap
man.hap
sask.hap
alta.hap
bc.hap
yuk.hap
nwt.hap
```

a time. For instance, the CANADA.HAL is shown in the sidebar and reads...

This is a listing of all the needed HAPs to successfully run the Hyper Tour of Canada Application. By loading in the HAL file, it saves the user the trouble of manually having to load each HAP separately.

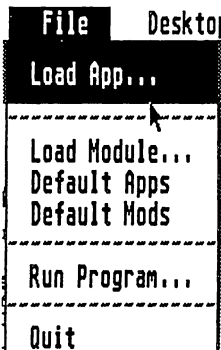
Finally you have gotten to the initial HyperLINK Main Screen. Each of the icons in the box on the left hand side of the screen represents a HyperLINK Application that has been loaded in and is available for your use.

THE MOUSE :

LEFT BUTTONS vs RIGHT BUTTONS

The left mouse button is used to select things within HyperLINK. As with most ST software, this is the primary mouse button used within the software, but as a short cut, in various parts of HyperLINK the right mouse button may call up extra menu selectors. The areas of HyperLINK where the right hand button is used : The Builder, any Database Application, and the Report Generator.

(Figure 2.3)



FILE MENU :

LOAD APP - Used for loading in a HyperLINK Application that is not one of the defaults already loaded.

LOAD MODULE - Used for loading in a HyperLINK Module other than the ones that

TWO

come with HyperLINK. All of the HyperLINK Modules that are standard with the product, load in every time HyperLINK is started up, as they are listed in the DEFAULT.MOL file.

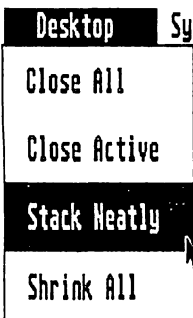
DEFAULT APPS - The text file listing of all the HyperLINK Applications which are loaded every time the program is booted. By clicking on this selection, the text editor is called up and this file is loaded in for editing.

DEFAULT MODS - Similar to the DEFAULT APPS, this is a text file containing all the HyperLINK Modules that you want loaded in every time the program is run.

RUN PROGRAM - This selection allows a user to run another regular ST program from within HyperLINK. When you quit this external program, you will find yourself back in HyperLINK at the same point you left it.

QUIT - Obviously is the menu selection for exiting HyperLINK.

DESKTOP MENU :



(Figure 2.4)

CLOSE ALL - This selection closes all the open HyperLINK Application windows.

CLOSE ACTIVE - This closes only the "Active" HyperLINK Application window. <F10> does the same thing as this menu selection.

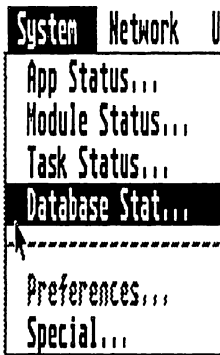
STACK NEATLY - Takes all the open HyperLINK Applications and arranges these windows in an orderly manner on the screen. How well it works depends on the window sizes

HYPERLINK

of the open HAPs.

SHRINK ALL - Takes all the open HyperLINK Application windows and minimizes these HAPs at their current records. The same as hitting <F9> but for all open HAPs, not just the active one.

SYSTEM MENU :



(Figure 2.5)

APP STATUS - Lists all the Applications currently loaded into HyperLINK and their associated database files, their ID's and Version numbers.

MODULES STATUS - Lists all the Modules currently loaded into HyperLINK and their ID's and Version numbers.

TASK STATUS - Not really an incredibly useful stat, but shows the user how much of the system processing time is being taken up by some background system functions, such as the Database Controller.

DATABASE STAT - Lists the currently open databases which the HyperLINK Applications are using, and also shows how many records are contained in these databases.

PREFERENCES - Not currently used by HyperLINK

SPECIAL - Not currently used by HyperLINK, this is a "spare" menu entry that can be used by custom modules or load-in functions.

TWO

NETWORK MENU :



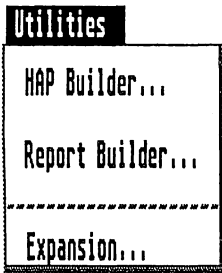
(Figure 2.6)

NETWORK SETUP - Lets the user configure HyperLINK for use with some of the commercial networks available for the ST.

SEND MESSAGE - Allows one station to send messages to other stations on the network. Only selectable if the network is up and running.

CHECK NETWORK - Checks the integrity of the network. Is only selectable if you have the network up and running.

UTILITIES MENU :



(Figure 2.7)

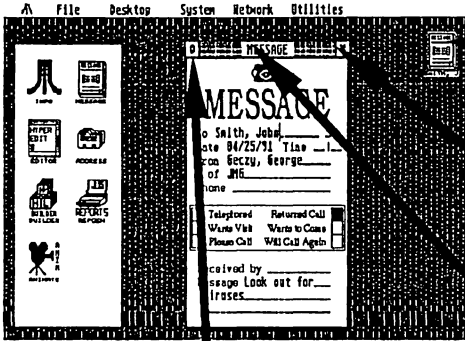
HAP BUILDER - Operates the same as selecting the Builder from the main menu. Is a good short cut, particularly if the Builder has not been loaded into the main menu. In this case, this menu selection will not only load the Builder in, but also call it up.

REPORT BUILDER - Similar in operation to the above menu item, this provides a short cut for first loading in the Report Builder and then running that HAP.

EXPANSION - Not currently used by HyperLINK, but when it is, it will operate the same way as the APP BUILDER selection.

HYPERLINK

APPLICATIONS ON SCREEN :



(Figure 2.8)

← This illustrates a "Minimized" HAP and it keeps you at the last record.

The Upper Right Corner Button in a HAP is used to do the minimizing.

This is the Title Bar and is also used for moving a HAP around on the screen.

This is the button used for closing an Application.

THE BUILDER :

The very heart of HyperLINK is the BUILDER. For detailed examples of how the BUILDER can be used to create whatever you want, check out the Tutorial Section of the manual where there are three Step-By-Step examples.) The BUILDER is the source for all the user created forms, such as database layouts, and interactive screens. The Application Builder itself though is made up of many parts...

(Figure 2.9)



THE BUILDER MENU : (Called up by clicking on the Right Hand Mouse Button)

LOAD - Used to load in an existing application in to the builder for modification.

NEW - The selection made to start creating an application in the builder. This loads in a blank

TWO

form from which you can start creating your own HAP.

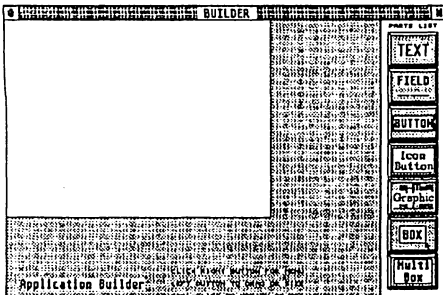
APP INFO - See the later description of the screen called up by this selection. Basically used for naming and setting up some of the defaults your HAP will use. IT MUST BE FILLED OUT PROPERLY FOR EVERY HAP! If you do not fill in the information for your HAP before you save it for the first time, HyperLINK will guess at what this info should be. While HyperLINK is intelligent enough to understand what you are trying to do, it might be misunderstanding your intentions. Therefore it is always a good idea to manually fill out this information for every HAP you create.

SAVE - Saves the application you are currently constructing.

SAVE AS - Saves the application you are currently constructing but gives you the chance first to give the HAP an alternate name.

MAIN BUILDER SCREEN :

The HyperLINK Builder Application is the place to design your HyperLINK Applications. When first called up it is merely a blank form. Simply by dragging **TEXT**, data **FIELDS**, **BUTTONS**, **ICONS** (and Images), and **BOXES** to the form anywhere you want, you can generate an application. Not only though is the Builder used to create the look of a



(Figure 2.10)

HYPERLINK

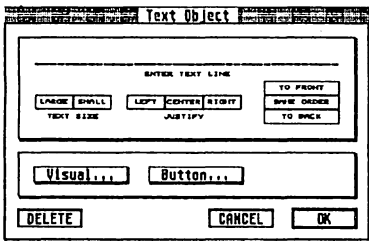
HAP but more importantly the builder is where all the LINKS are created that an application uses and also where much of the information setup is created for your HAP to reference (for instance : Database File Names, HAP Name, Icon to use in HyperLINK menu, etc).

NOTE : Two definitions are helpful when it comes to understanding certain aspects of the Builder...

PARENT - Is an object on the form that has other objects falling within it. The entire form itself is actually the ultimate parent but this does not prevent there being other Parent Objects within.

CHILDREN - These are objects that fall within another object in the Builder Form. Every object that you drag to your HAP from the Building Tools is actually a child of the overall Form. There is no limitation to children having children. For instance, a box in your HAP is a child of the overall form and anything in that box is a child of the box (and I guess a grandchild of the overall form etc.).

Building with TEXT :



(Figure 2.11)

When you want to add TEXT to the Builder Form, you simply drag the TEXT box from the Building Tools onto the form in the location you wish the text to begin. Immediately will pop up the Text Object dialogue box. What you do here is enter the content of your text field, set up the criteria for the appearance of the text ie. Large or Small text size, and Left, Right or Center justified, and

TWO

position the text as it relates to other objects in your form (ie. Front/Back). Also from this screen you can call up two other important configuration screens, the VISUAL dialogue, and the BUTTON* converter. The VISUAL screen is common to whatever tool you are adding to your form and will be discussed in more detail after the rest of the Building Tools have been explained.

* See *Building with BUTTON* for a greater explanation on the types of Buttons available!

The image shows a dialog box titled "Field Object". It has a title bar with the text "Field Object". Inside the dialog, there are several controls: "FIELD LENGTH:" followed by a small input box and "DECIMALS:" followed by another small input box. Below these are four radio buttons labeled "ALL", "CAP", "NUM", and "DAT" under the heading "TYPE:". To the right of these are three buttons: "TO FRONT", "SAVE ORDER", and "TO SPACE". Below the "TYPE:" section is a label "FIELD NAME:" followed by a text input field and a button labeled "Expression". At the bottom of the dialog are two buttons: "Visual..." and "Linking...". At the very bottom of the dialog are three buttons: "DELETE", "CANCEL", and "OK".

(Figure 2.12)

Building with FIELD :

When you wish to add a Data Field to your Builder Form, you simply drag the FIELD box from the Building Tools onto the form in the location you wish the Field to begin. Like any of the objects you are adding to a new form, if you

don't like the initial placement of an item, it may always be moved around later. Immediately will pop up the Field Object dialogue box. While this dialogue box is very similar to the Text Object dialogue, there are some important differences as well. Within this info screen you can set the length of your data field, what it is comprised of (ALL - Any Character / CAP - Alphabetical characters in capital format / NUM - Numerical characters / DAT - A date format ie. 01/17/92), how many decimal points are available to numerical fields, as well as the BUTTON settings, and relevant placement to other objects. From the Field Object screen you can also call up the LINKING and VISUAL screens, and also a third screen, specific only to Field Objects, the EXPRESSION screen. This is used for adding

HYPERLINK

mathematical and other function capabilities to your data fields.

Building with BUTTON :

There are four types of buttons used by HyperLINK and it is important that you understand the differences between the various types. There really is no way to just give an Anatomy of the different buttons, so the content here is virtually the same as it is in the reference section of the manual...

BUTTON DEFINITIONS :

LINKING BUTTON - This is the regular type of Action Button you would select in your HAPs. It indicates that when this application is being run, you can select this Object on the form (ie. Click on it to cause an action). What happens when you select this object is still very much customizable through the LINKING screen.

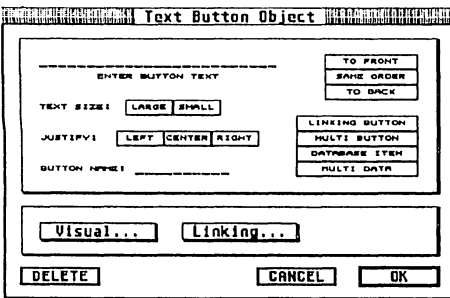
MULTI BUTTON - This type of button is part of a collection of buttons. When a button is MULTI it means that in the same collection of buttons (all with the same parent object), only one Multi button may be selected at one time. For instance, imagine a Parent Box in a database that deals with a person's sex. Within that Parent Object would be two MULTI BUTTONS, one labeled "Male" and the other "Female". When you are actually running the HAP, you could activate the Male button. If you were then to click on the Female button, the Male button would deactivate at the same time as the Female button was highlighted. In other words, within a Parent Object, among all the Children Object Buttons that are marked as

TWO

MULTI, only one may be selected at any given time. By selecting another, you are also de-selecting the first. (It is possible though to have both MULTI and NORMAL buttons within the same parent.)

DATABASE ITEM - This really is using a Button to represent a data field. As an example, suppose you have a questionnaire HAP to be filled out. You could on this questionnaire have a button to select if you are married. HyperLINK is handling this button as a Yes/No type data field and is interpreting a selected button as a Yes answer, and a non-selected button as a No answer and internally replacing your button with either a "Y" or an "N" in the actual record. In this manner it allows for easy interpretation in any report you may later set up.

MULTI DATA - This Button type is a cross between a Multi Button and a Database Item. It is used under the same condition as a Multi Button (ie. A series of buttons wherein only one is selectable at a time). Rather than storing a field in your database for each Button though, Multi Data stores one field for the Parent (must be a Multi Parent) which is a numeric item indicating which button is selected.



(Figure 2.13)

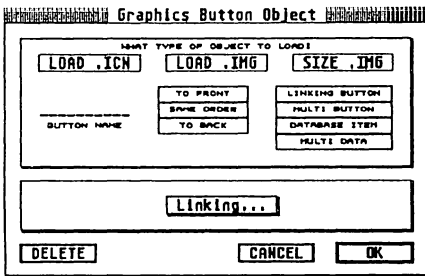
Back to Building with BUTTON :

Within the Builder, dragging a Button Object to your form calls up a similar Dialogue Box to that you got when you dragged a Text Object to your form. The difference being, the default settings within the

HYPERLINK

VISUAL screen (it already has a border around whatever text you input), and the additional setting for Button type. Also a button inherently performs some form of action and therefore needs to be given a name. Again, if you don't give your button a name, HyperLINK will do it for you.

Building with ICON BUTTONS :



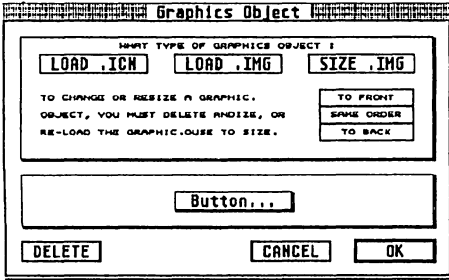
(Figure 2.14)

This Building Tool is somewhat wrongly labelled. If there were appropriate room it should be ICON / IMAGE. This is the object you drag to your Builder Form when you want to load an Icon or an IMG graphic to your application. When you want to add an IMG file or an Icon to the Builder Form to be used as a

button, you simply drag the ICON BUTTON box from the Building Tools unto the form in the location you wish for the upper left hand corner of the graphic. Immediately will pop up the Graphics Button dialogue box. What you do here is select whether you will be loading an Icon or an IMG file, give it a file name, indicate what type of button it is, and position the graphic as it relates to other objects in your builder (ie. Front/Back). Also from this screen you can call up one other important configuration screen, the LINKING dialogue. The VISUAL screen is currently not available from within the Graphics Object as there is no ability to alter the appearance of the graphic you are loading in.

TWO

Building with GRAPHIC :



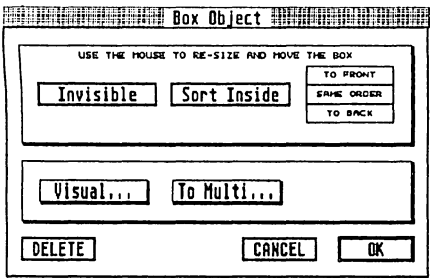
(Figure 2.15)

Very similar to the Icon Button with the exception that when you drag this onto your HAP you are not given any Button Type options. In other words, this GRAPHIC Object is being used only to improve or enhance the appearance of your HAP (such as a backdrop) and does not perform any function.

However if you do want to convert a plain Graphic into a button this is very simple. Just select "Button..." and you are in effect transforming your graphic into a button with all the regular options.

Building with BOX :

When you want to add a BOX to the Builder Form, you simply drag the BOX item from the Building Tools onto the form in the location you wish the upper left hand corner of the box to begin. Immediately will pop up the Box Object dialogue. A Box is just a graphic rectangle within the HAP and performs no actual function other than being a parent to any objects inside it.. As



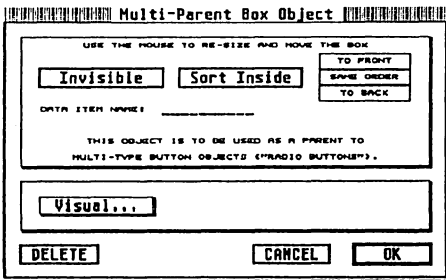
(Figure 2.16)

such, all you can set within this screen is the appearance of your box and the sorting of any children inside it, and position the box as it relates to other objects in your builder (ie. Front/Back/Same Order). As well, you set whether the box is Invisible, in which case it merely acts as a transparent Parent for

HYPERLINK

grouping Children Objects. Also from this screen you can call up two other important configuration screens, the standard VISUAL dialogue, and the "To Multi..." converter which is a special case Box.

Building with MULTI BOX :

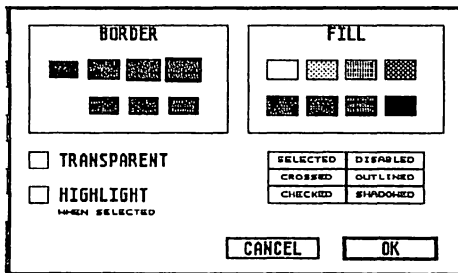


(Figure 2.17)

This is a specialized box that designates all the buttons within it (ie. its Children) as Multi Buttons. The use of a Multi Box can save you some time and effort in the creation of a HAP. Other than the special characteristics of a Multi Box, this screen is

essentially identical to that of any regular Box Object. NOTE : While a normal Box Object in any HAP does not need a field name, a Multi Box does.

The VISUAL Screen :



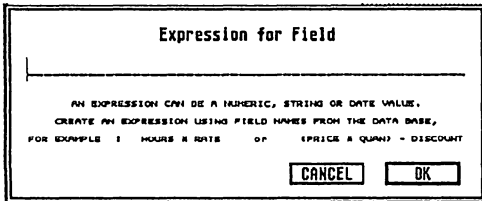
(Figure 2.18)

This is probably one of the easiest of all screens to understand within HyperLINK. Here is where you control the physical appearance of how an object will be displayed on a HAP. You control : Borders, Fill Patterns, Shadowing, and other effects. The best way to figure out the capabilities of

this screen are by playing around with it and seeing which effects suit your HAP best.

TWO

The EXPRESSION Screen :



(Figure 2.19)

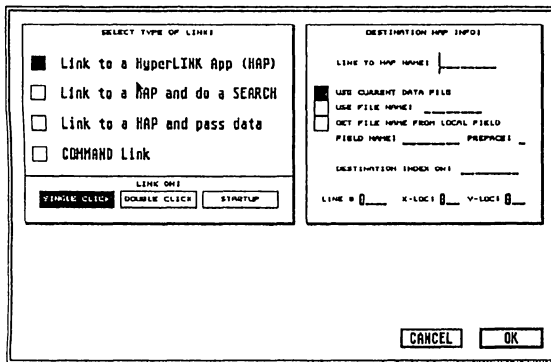
Only accessible from within the Field Object screen, this is only applicable to database fields that are to do more than just contain data, but also perform some mathematical calculations or other expressions.

The LINKING Screens :

"Linking" being possibly the hardest concept within HyperLINK to fully understand, we have included sections on this not only in the reference section of the manual, but also in a technical section on how the linking works (Section V). Links are created within HyperLINK to join different types of data from one application to another. There are changes to the basic Linking screen depending on the type of link you are creating...

Link to a HAP :

(Figure 2.20)



Finally after many different ideas in earlier versions of HyperLINK, we think that the Linking screens are set up in as easy to understand a way as is possible, considering the complexity and the power of what these

HYPERLINK

screens actually do. On this screen for instance, used if you are performing a straight link to another HAP, simply select how that link is first called (ie. "Link On:" a single mouse click, or a double click), which HAP you are linking to, and tell HyperLINK which datafile name to use in the destination (linked to) HAP.

Link to a HAP & Search :

The screenshot shows a dialog box titled "SELECT TYPE OF LINK:" with four radio button options: "Link to a HyperLINK App (HAP)", "Link to a HAP and do a SEARCH" (which is selected), "Link to a HAP and pass data", and "COMMAND Link". Below these are three buttons: "SINGLE CLICK", "DOUBLE CLICK", and "STARTUP". To the right, under "DESTINATION HAP INFO:", there is a "LINK TO HAP NAME:" field, a "USE CURRENT DATA FILE" checkbox (checked), "USE FILE NAME:" and "GET FILE NAME FROM LOCAL FIELD" checkboxes (unchecked), "FIELD NAME:" and "PREFIX:" fields, a "DESTINATION INDEX DNI:" field, and "LINE #", "X-LOC", and "Y-LOC" fields. At the bottom, under "LOOKUP SEARCH OR PRESSED DATA SOURCE:", there are three checkboxes: "GET FROM LOCAL FIELD:", "DIRECT PARAMETERS:", and "EXPRESSION:", each followed by a text input field. "CANCEL" and "OK" buttons are at the bottom right.

(Figure 2.21)

Very similar to the above link with the exception that you have to fill in the details on any search criteria you want HyperLINK to perform after it completes the initial link.

Link to a HAP & Pass Data :

The screenshot shows a dialog box titled "SELECT TYPE OF LINK:" with four radio button options: "Link to a HyperLINK App (HAP)", "Link to a HAP and do a SEARCH", "Link to a HAP and pass data" (which is selected), and "COMMAND Link". Below these are three buttons: "SINGLE CLICK", "DOUBLE CLICA", and "STARTUP". To the right, under "DESTINATION HAP INFO:", there is a "LINK TO HAP NAME:" field, a "USE CURRENT DATA FILE" checkbox (checked), "USE FILE NAME:" and "GET FILE NAME FROM LOCAL FIELD" checkboxes (unchecked), "FIELD NAME:" and "PREFIX:" fields, a "DESTINATION INDEX DNI:" field, and "LINE #", "X-LOC", and "Y-LOC" fields. At the bottom, under "LOOKUP SEARCH OR PRESSED DATA SOURCE:", there are three checkboxes: "GET FROM LOCAL FIELD:", "DIRECT PARAMETERS:", and "EXPRESSION:", each followed by a text input field. "DESTINATION FIELD FOR PRESSED DATA:" is also present. "CANCEL" and "OK" buttons are at the bottom right.

(Figure 2.22)

This links to a blank record in your destination HAP and fills data in to any specific field from one of the fields in your source HAP.

TWO

Command Link :

SELECT TYPE OF LINKS

Link to a HyperLINK App (HAP)
 Link to a HAP and do a SEARCH
 Link to a HAP and pass data
 COMMAND Link

LINE OPT:

COMMAND LINE INFO

Command: _____
Parameters: _____

(Figure 2.23)

Rather than linking to another HAP, this type of link performs or completes a specific command. It could be a complex command, or even something as mundane as a button to go to the next record in your data file.

The APPLICATION INFO Screen :

Another important screen associated with a prime part of the BUILDER that has to be properly completed for each application is the Application Info Screen. It is called up in one of two ways from within the BUILDER. <ALT> + <I> is the quickest way to get to this screen, but for those of you more mouse oriented, you can call up the App Info by clicking once with the Right Mouse Button (this calls up the BUILDER MENU) and moving the mouse pointer to App Info and selecting it with the Left Mouse Button.

(Figure 2.24)

Application Information

MUST BE FILLED OUT

HAP Name : _____
Uses MODULE : _____

DATABASE INFORMATION

STANDARD HYPERLINK DATABASE
 CUSTOM OR NO DATABASE

DATABASE NAME DEF INDEX

USER DEFINED INFO

AUTHOR ID: _____ NUMERICAL ID: 0____
HAP REVISION: 0,____

VISUAL INFORMATION / OPTIONS

LOG IN ICON WHEN HAP LOADED ?
 AUTOMATICALLY LOAD FIRST RECORD ?

FIRST EDIT FIELD

HAP ICON

HIDDEN FEATURES

HIDE
HIDE SHOW CLOSE BOX
 HIDEUP
MOUSE BAR FULL/MIN

SPECIAL / CUSTOM INFORMATION

SECONDARY FILENAME FILE SELECT

On the APP INFO screen, the user defines many things for use with their application. Everything from the name and version of the HAP, to the Icon it uses in the main HyperLINK Menu (if it uses one at all), to the database file this HAP

HYPERLINK

will use, to the actual screen appearance of the application. These are all defined here on this one screen.

The APPLICATION DEFAULTS Screen :

GENERAL INFORMATION	MODIFIABLE PARAMETERS
HAP Name : BUILDER	Database File : _____
Author ID : JMG Ver 200	Default Index : _____ LIST
Module Name : BUILDER	Secondary Name: _____
DB Type : APPDEF	First Edit : _____
INDEX FILE CONTROLS	
MAKE NEW INDEX : Create INDEX	LOCATION : X <input type="radio"/> B <input type="radio"/> V <input type="radio"/> (1M CHRS)
FIX/BRO INDEX : RE-INDEX	<input type="checkbox"/> AUTO LOAD 1ST REC <input type="checkbox"/> FILE SELECT
REMOVE DELETED : PACK	CHECK FIELDS ETC : CHECK Date
REMOVE INDEX : Delete INDEX	
CANCEL OK	

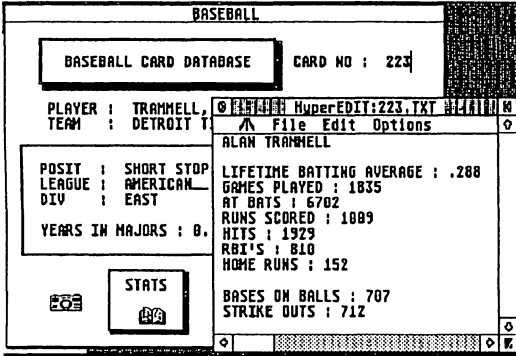
(Figure 2.25)

There is also a further screen that is relevant to the BUILDER, the APPLICATION DEFAULTS screen. This screen is not called up from within the BUILDER though. Rather it is called up from the Main Menu. Instead of just clicking

on an application in the Main Menu, by holding down the <ALT> key when you click on any application, you bring up the Application Defaults for that HAP. Some things on this screen are duplicated from the APP INFO screen within the BUILDER, such as which data file to use, which default index to use, whether or not to load the first record when your database is called up. Also on this screen you can create new indexes for your database, or pack your database to eliminate deleted records, etc. NOTE : Any changes you make on this screen are only valid in this session of HyperLINK. To make your settings permanent, use the APP INFO screen within the Builder.

TWO

THE TEXT EDITOR :



One of the things that separates HyperLINK from any other ST Database software is the inclusion of a text editor. This specific HyperLINK Application can be used independantly of other HAPs or it can be an integral part of another HyperLINK Application.

(Figure 2.26)

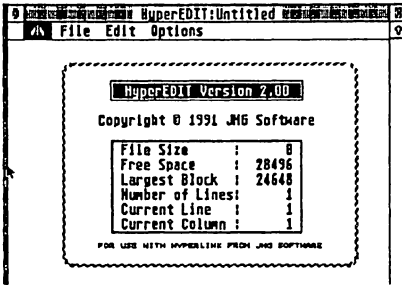
Using the Baseball Card Database as an example, you can see a standard database that contains a button "STATS". By selecting this button you are calling a LINK to a text file associated with that database record. When this type of link is used, the Text Editor is called up and the LINKED text file is loaded in to the editor. NOTE : For those of you in the ST world that are familiar with the Text Editor "STENO", HyperLINK's Text Editor should look familiar. It is a modified version of that product. We wish at this time to thank Eric Rosenquist for his help and his support with this aspect of HyperLINK.

The HyperLINK Text Editor is pretty basic in operation as compared with other editors available separately. The HyperLINK Editor though does save and read ascii format, so that if you wanted to either edit or create a file in another word processor this is not a problem.

HYPERLINK

TEXT EDITOR MENUS :

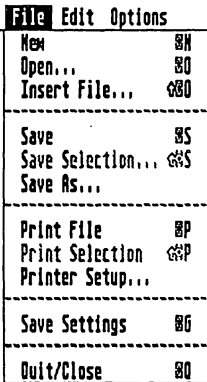
Unlike typical ST Gem Menus, if you wish to select one of the window menus of the editor, you have to click the left mouse button on the desired menu to activate it.



(Figure 2.27)

If you want to find out the status on the document currently loaded in, click on the Atari Fuji Symbol and then select "HyperEdit Info" from the menu.

EDITOR FILE MENU :



(Figure 2.28)

New File - Clears whatever text file you are editing out of memory for you to work on a new file. If you haven't saved the file you were working on, you are given the chance to save it before the current file is abandoned.

Open File - Similar to above option, but calls up your file selector asking which text file you wish to load in. Also gives you the option at this time of saving your current work before it is lost.

Insert File - Calls up the file selector asking for which text file you wish to Append to the current document being edited. Append occurs at current location of the cursor.

Save - Does not abandon current document, just gives you the option of saving the text file currently being edited. (Remember, it is always a good idea to save your files often.)

Save Selection - If you have a marked block within your text file, this menu item allows you

TWO

to save out only the selected text to a separate file.

Save As - Saves the file, first calling up the file selector to give you the option of saving a file under a different name or in a different path. (If you continue using the same name every time you save a file, you just write over the last version of that file. Therefore it is a good idea to change the name every so often when you are saving to help insure the safety of your data.)

Print File - As it says, sends the current text file to the printer.

Print Selection - If you have a marked block within your text file, this menu item allows you to print out only the selected text.

Printer Setup - Used for setting up the page format for your printouts. Lines per page, Margin, this is where you set up things like this.

Save Settings - This is used for saving the default configuration of your editor window. Window sizing, window positioning, can all be saved here. This feature is not available in the current version of the editor, but is planned for inclusion in a future update.

Quit/Close - This selection closes the active editor window giving you the option to save first.

EDITOR EDIT MENU :

Much the same as any editing functions you are used to in other software, to select a block of text for editing simply hold down the left mouse

HYPERLINK

button and drag the cursor over the text you wish to highlight.

NOTE : In this section, a "CLIPBOARD" is referred to many times. This is used by the text editor as an external file for temporary storage. Only one item can be in this buffer at one time. If you send a second item here, then the first one is written over and lost. The Clipboard being saved as a temporary file and not just allocated to memory allows it to be kept even if you exit the HyperLINK program.

Edit	Options
Cut	⌘X
Copy	⌘C
Paste	⌘V
Delete	DEL

Select All	⌘A

Reformat Paragraph	
Reformat to End	

Find...	⌘F
Find Next	⇧⌘A
Change...	⌘H

(Figure 2.29)

Cut - Moves any selected text from where it is in your file to the Clipboard. This is used for moving text blocks about in a document.

Copy - Similar to "Cut" but the copy function does not delete the text it is moving to the memory buffer. This is used when you might be using the same block of text somewhere else, either in this document or another.

Paste - Inserts whatever text is currently in the Clipboard, starting at the present location of your cursor.

Delete - Erases the currently selected block of text from the document.

Select All - Marks the entire text of the current document.

Reformat Paragraph - If you've inserted text to a paragraph, this option reformats the text to match the current configuration of the editor. Only reformats the current paragraph where your cursor is located.

TWO

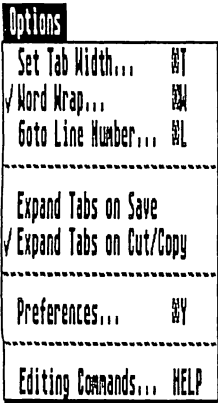
Reformat to End - Similar to reformatting a paragraph, this option is not limited to just one paragraph, but reformats the document from the point of the cursor to the end of the file.

Find - Searches for an occurrence of a specific text pattern. Used primarily for finding specific words or phrases within a document.

Find Next - Does not change the criteria of the last search, merely goes on to look for the next occurrence.

Change - This is a "Search and Replace" option. Whereby the user can tell the editor to change all occurrences of one specific text content to something else.

EDITOR OPTIONS MENU :



(Figure 2.30)

Set Tab Width - Sets the interval at which tabs occur on a line in your text. For instance, a tab width of 4 allows for four spaces between tab marks, therefore there would be tab marks at column 5, 10, 15 etc. Also under this selection you can configure the HyperLINK Editor to display the tab marks in any open window.

Word Wrap - This is used primarily to set the line length of your document. Being that the HyperLINK Editor window can be sized variably, you would usually set the "Wrap Column" width to what would fit in your open window.

Goto Line Number - This moves your cursor to whichever line number in your text file you specify.

HYPERLINK

Expand Tabs on Save - If this option is enabled, then when you save a file, every tab mark is expanded and replaced by spaces.

Expand Tabs on Cut/Copy - The same as for "on Save" as it relates to things being sent to the Clipboard.

Preferences - Allows you to modify the point size of the screen font to be used with HyperEdit. Also allows the user to configure the amount of computer memory allocated for the default size of HyperEdit files. For instance, if the Memory Buffer Size is set to 32K then the Editor puts aside 32K to store text. If a larger file is loaded, HyperLINK will try to allocate additional RAM if it is available.. A third setting under this option is the location of the editor's Clipboard.

Editing Commands - Calls up a HELP screen for commands within the HyperLINK Editor. The commands covered are Page Up, Page Down, Go to the End of a Line, Delete Current Line, and more.

THE TEXT LINKING BUILDER

(Figure 2.31)

TEXT LINK BUILDER

LINK TO HWP NAME : LOOKUP AFTER LINK

FILE NAME TO USE : (BLANK FOR CURRENT FILE)

INDEX TO USE : (BLANK FOR DEFAULT)

TO LINE # : (WHERE APPLICABLE)

LOOKUP STRAIN :

The Text Linking screen is called up when you are creating a link from an existing text file. You can form a link to a Database HAP, another Text File, a Graphic File, or any other module that may be available in the future for HyperLINK. To originally

TWO

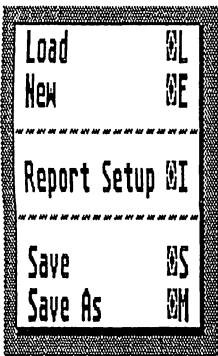
call up this screen, <ALT> click on any word in the file currently loaded in the text editor. For more details on "Linking" in general, see both the Reference Section and the Appendices.

THE REPORT GENERATOR :

Like the Application Builder, the Report Generator is not just comprised of one component. It is very straightforward to create reports that are just spewing pre-canned data to a printer, but it gets a little more complicated when you also wish for sorting, searching, and calculations done. As such, the more involved reports call up an additional screen to handle these parts.

The REPORT GENERATOR MENU :

(Similar to the Builder, the menu for the Report Generator is called up by clicking with the Right Mouse Button.)



(Figure 2.32)

LOAD - Used to load in an Report in to the Report Builder for modification.

NEW - The selection made to start creating a new Report in the builder. This loads in a blank form from which you can start creating your own Report Form.

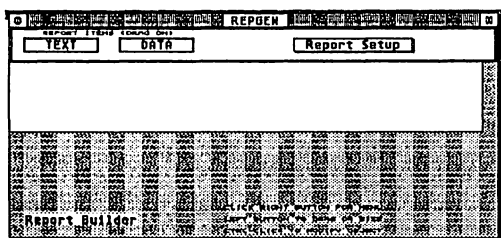
REPORT SETUP - Calls up another screen for doing the general setup as well as the intricate details of your Report. This is gone in to in much further detail later in this section.

HYPERLINK

SAVE - Saves the Report you are currently creating.

SAVE AS - Saves the Report you are currently constructing but gives you the chance first to give it an alternate name or an alternate path.

MAIN REPORT BUILDING SCREEN :



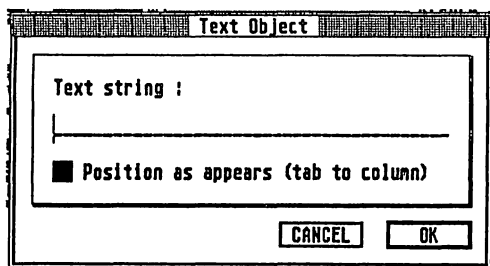
(Figure 2.33)

The HyperLINK Report Generator is the place to design all of the Reports that you want your HyperLINK Applications able to access and print.

It works in much the same manner as the Application Builder itself and when first called up it is merely a blank form. Simply by dragging TEXT, and DATA to the form anywhere you want, you can generate an application. Not only is the Report Builder used to create the look of a Report but more importantly this builder is where all the Searching and Calculations for your Report are configured.

(Figure 2.34)

Building a Report with TEXT :



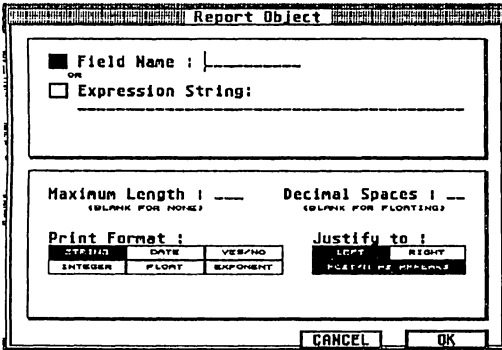
Unlike the Application Builder, dragging TEXT to a Report involves much less configuring. When you move a Text entry onto your Report you have only to enter the text as you want it to appear in the Report and whether you wish the

TWO

text to "Tab to a Column" or whether it will immediately follow the preceding text or field without tabbing. TEXT on a Report is simply a non-changing constant wherein the same text will appear in the same place every time you print your report. It is useful for headings, and any descriptive explanation you wish to have included with your customized report.

Building a Report with DATA :

The fundamental difference between TEXT and DATA in your report is that DATA is an ever changing item. When you are printing a report, it is the DATA portion that makes it unique. DATA can be many different things, for a field in a database, to a mathematical expression, to any number of other logical functions the computer can perform.



When you drag this Building Tool to a report, you have to fill out the appropriate information that tells your report where to get the data to fill this space. For instance, when placing DATA on a report you must then (on the dialogue box that then appears) inform

(Figure 2.35)

HyperLINK whether the data you wish in this location is a "Field" from a database, or some manner of "Expression String" (For complete information on how to fill out Expression Strings and the commands available to you, see the Reference section of the manual). Also on this Data Info screen you must set up the field length

HYPERLINK

for your data, the printing format of your data (ie. String, Date, Yes/No, Integer, Float, or Exponent), the number of Decimal Places in your Data, and the printing Justification of the Data as well.

That just leaves two more screens associated with the Report Generator, and both these screens are directly related. They are the REPORT SETUP and the CUSTOMIZED REPORT SORT & SEARCH screens.

REPORT SETUP :

Report Setup

Print All Using Default Sort
 Use Custom Sort and Search Options

Page Length : 66_ FORM FEEDING
Left Margin : 0_ Right Margin : 88_ BEFORE BETWEEN AFTER

Printer Font : SMALL 0 0 0 0 LARGE

CANCEL OK

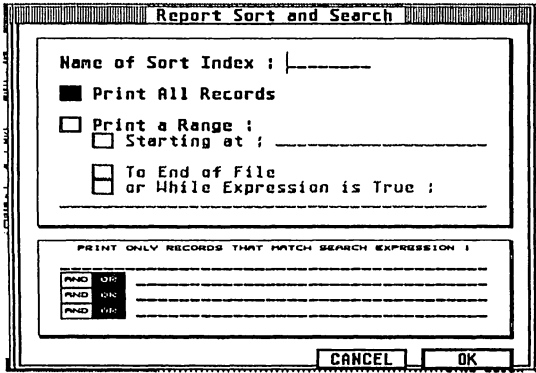
(Figure 2.36)

On the basic Report Setup screen you must enter the Page Length of your report, the Right and Left Margins, the Printer Font to use, and information about Form Feeding on your printer. There is also the availability of entering "Custom Sort and Search Options". This must be completed if you wish your Report to select indexes to use or to perform record searching. For instance, if you wish to use an index other than your HAP's Default Index, or if you only wish to print a limited range from your database you would click on this "Custom Sort" option and up would pop...

The CUSTOMIZED REPORT SORT & SEARCH Screen :

Understanding this screen is key to making the most out of the Report Generator. While it is

TWO



(Figure 2.37)

options you have available from this screen.

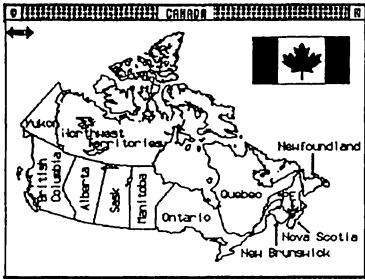
fairly straightforward to obtain a basic grasp, you must fully comprehend the unlimited power that this screen offers with your own Searching and Sorting Criteria! Carefully go over the description of this item in the Reference Section for a complete explanation of the

THE IMG GRAPHICS DISPLAYER HAP & MODULE :

HyperLINK can display .IMG Graphics files either as part of a HAP (ie. graphic supplement included as part of a HAP either for esthetic reasons or for better illustration of a concept), or on their own in a separate window.

The Graphic Module within HyperLINK is the internal programming code that allows these .IMG files to be used by HyperLINK, while the Graphic HAP is basically an empty HyperLINK Window that loads up a graphic (modifying the window's size to fit the graphic as best as possible) for display purposes.

In the Canada HAP for instance we have examples of both the Graphic HAP, and Graphics Modules being used within another HAP. The Canada map from which you can click on any of the Provinces to enter a HAP based on that specific Province is a HAP that is using a graphic



(Figure 2.38)

backdrop and placing invisible buttons on top which maintain the Linking Information to other HAPs. Also within this Canada HAP is an "Images of Canada" section whereby clicking on a button for everything from Queen Elizabeth, to a Mountie, to a Loon, calls up the Graphic HAP and loads in the appropriate image file.



(Figure 2.39)

THE ANIMATION PLAYER :

Similar to the need for a Graphic HAP and a Graphic Module, is the need for an Animation Module and an Animation HAP. Essentially you must have the module, to allow for the existence of the HAP itself.

THE NUL MODULE :

Essentially this is the module used by HyperLINK Applications that aren't databases, text files, graphic pictures, etc. In other words it is the module that handles applications which don't contain any "Interactive Screens". An example of such a HAP is again the Canada HAP. The HAP that is first called up, which displays the Canadian flag, is an application which uses the Nul Module. All that happens on this screen is the Canadian Map is displayed when you click on the flag. This HAP is nothing more than a button to call up another HAP. To do this, the Nul Module is what is used.

*That covers the basic anatomy of HyperLINK.
For more details, stay tuned...*

TWO

This Page Intentionally Left Blank

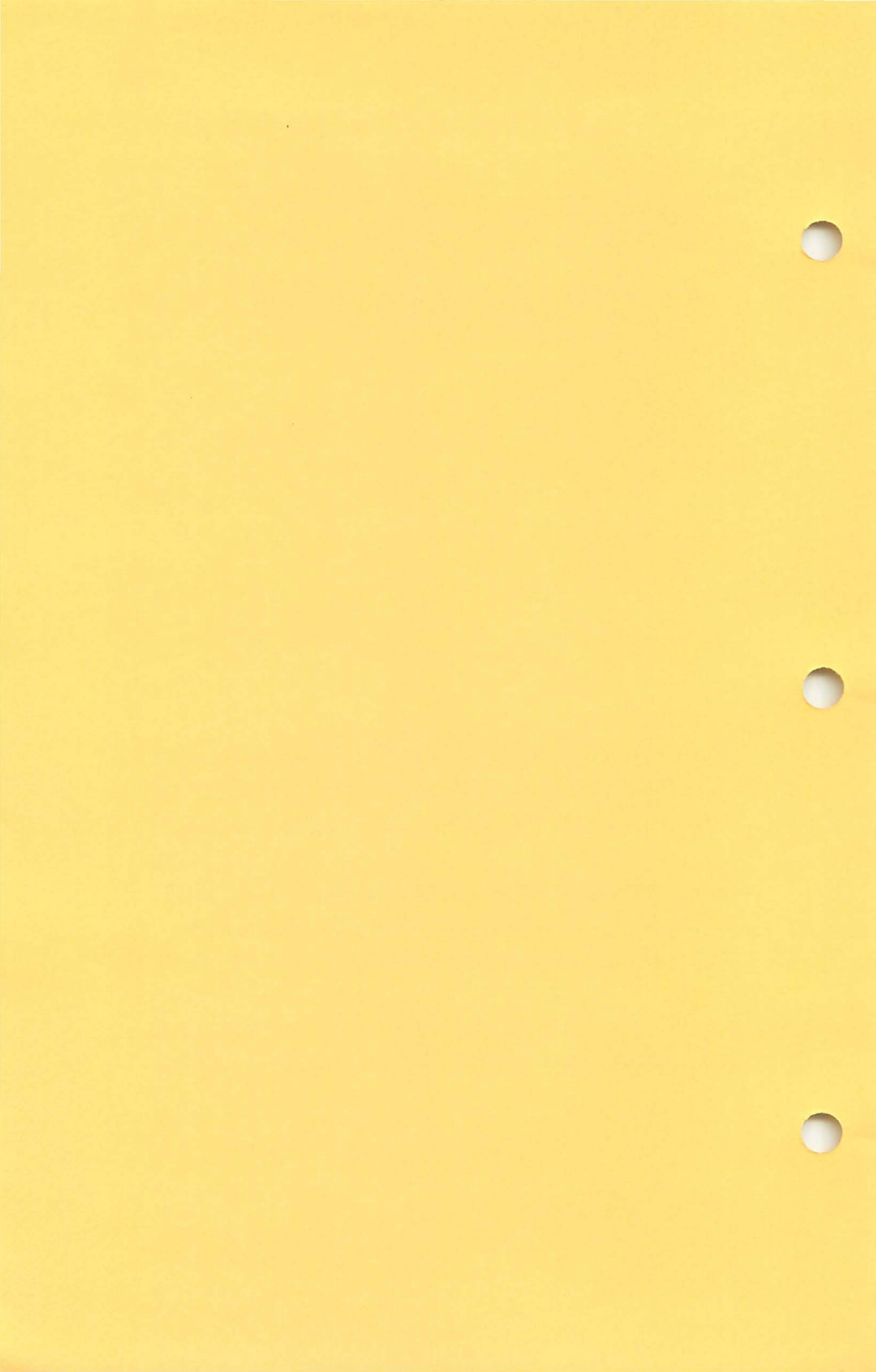


HYPERLINK

Step By Step

This is a Step By Step Tutorial Section which takes you through the building of some sample HyperLINK Applications.

It demonstrates some of the different types of HAPs you can build with HyperLINK and the different types of LINKS you can perform. Use this section to follow through the building process and learn how to create your own HAPs.



CREATING A HYPERLINK APPLICATION :

THE STEP-BY-STEP GUIDE

Now that you have a basic understanding of what HyperLINK is and what it can do, the best way to learn how to put together a HyperLINK Application (HAP) is to follow along on our step by step assembly of some demo HAPs.

We will first look at a typical database application for HyperLINK. For our purposes, (mostly because one of the principals in our company collects them) we will first consider a HAP for a Baseball Card Collector.

THE BASEBALL CARD HAP

The first thing you should do when creating a HAP is figure out what exactly you want to keep track of. For our purposes we will keep track of : Player / Team / Card No. / League / Division / Player Rating / and some elementary stats.

First Ideas Of Application Fields :

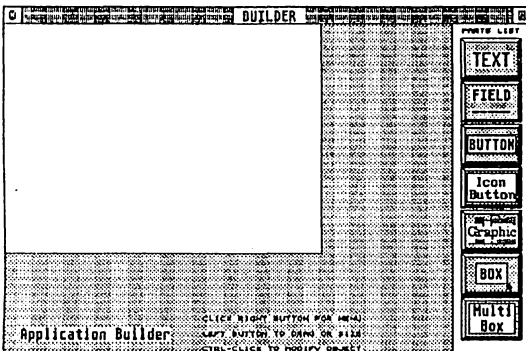
Field	Field Name	Type	Length
Player	Player	Text	40
Team	Team	Text	40
League	League	Text	8
Division	Division	Text	4
Card #	Card	Number	4
Years	Years	Number	2

THREE

NOTE !!! Once the data file has been created for an application, you can not simply add more fields to your HAP and then expect HyperLINK to see them and modify your original database file. You have to use the Check Data option in the App Defaults screen for HyperLINK to recognize additional fields ..

As you proceed in building a HAP, more ideas will occur to you as you are designing it. With HyperLINK there is no need to worry about this. There are no initial requirements for allocating space. Modify an application to your heart's content! Because of this, only the most basic of concepts need be present when you start your building.

To begin creating a HyperLINK Application, start up the HyperLINK program as normal. If the Builder did not load in as a Default Application, load it in through the pull-down menu. Next, open the Builder Application. What should now appear on screen is the empty Builder Construction Form...



To start editing a new HAP, click on the right mouse button and a modified version of the database menu will appear, specifically for use with the builder. Select NEW from this menu and a blank area should then appear in the builder. Size this application screen

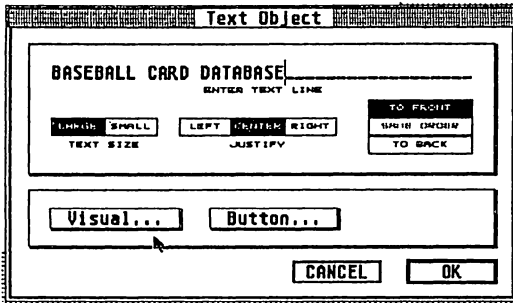
(Figure 3.1)

according to what you think you will need. Don't worry, if you need larger than you first select, you can still drag the box bigger. (At least to the maximum size of the builder screen.)

Now comes the time to first design the basic layout of your HAP, or how it will appear on screen when you call up your application.

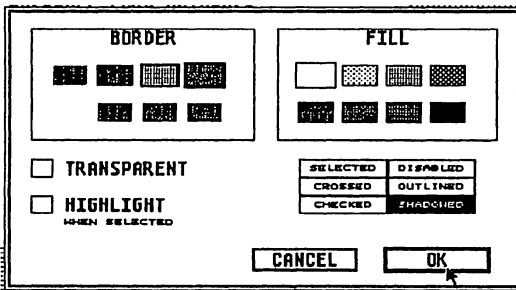
HYPERLINK

Well, let's first create a Text Object that will be the title for our HAP. To do this, drag a TEXT part to the Builder Work Space. Up will appear the Text Object Editor. Enter the text you want, in our case "Baseball Card Database".



(Figure 3.2)

From here we would go into the "Visual" screen and choose how we want our Text Object to appear. Since we are in effect making this Text Object appear as if it is within a Box, all we have to do is select an appropriate border. To give it an even fancier appearance, let's also make our Text Object shaded. This should help give our title a little extra emphasis.



(Figure 3.3)

If you have any trouble Double Clicking anywhere within HyperLINK, all you have to do is hold down the <CTRL> key and then single click with the left mouse button. This accomplishes the same thing.

If at any time in the future, you wish to increase the size of the border on your title or to modify the shadow effect, simply Double Click on the Object in your Builder Screen and the Text Object Editor will re-appear. Again make the "Visual" selection, and you will be able to change the appearance of your title.

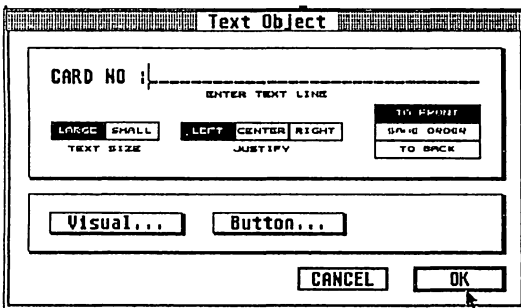
Note : The same is true of any object in your HAP form. If you don't like what you have, Double Click on it with the left mouse button, and you'll invoke the appropriate editing screen for whatever object you wish.

THREE

Okay, now you have a title... All you need is the actual application below.

Next step is arranging your data fields on the application. Putting the fields on the HAP is really a two-step operation. First you put a Text entry illustrating what the field is going to be. Secondly you drag a Field onto the HAP and then tell that field what it is going to be. (ie. size, type, name of field etc) When dragging a Field Object to your HAP, you are again met with

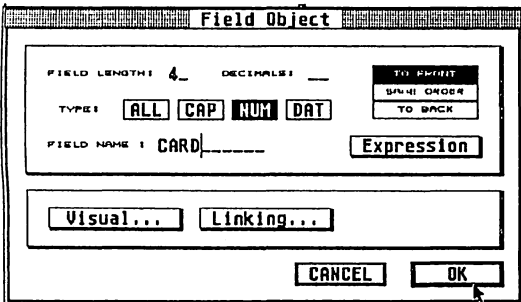
another editing screen. With the Baseball HAP in mind, my first data field is the Card Number. So, drag a Text Object to your form... This is just a simple text entry, so you don't need to change the Visual appearance of it.



(Figure 3.4)

Next, drag a Field Object onto your form and position it next to the "Card No" Text Object. Give your field a length : I don't think I'm going

to need more than four digits for my card number. Tell HyperLINK what type of data is going in that field: Obviously "NUM", representing a numerical data field. Give the field a name (all data fields have to have a name, so if you don't enter your

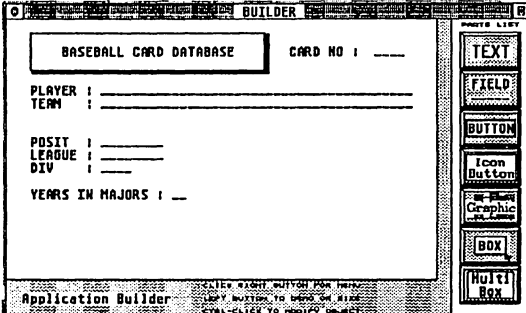


(Figure 3.5)

own name for that field, HyperLINK will generate a name itself) : Pretty straight forward stuff, let's call this field "Card".

HYPERLINK

Now we simply repeat this step for all the data fields we are putting in to our HyperLINK Application. (While we were entering our data fields it occurred to us to also add another field to represent a player's fielding position.)



(Figure 3.6)

Okay, in the space of about five minutes (probably less) we have done most of the ground work for a basic HyperLINK Application. Now lets fancy it up somewhat.

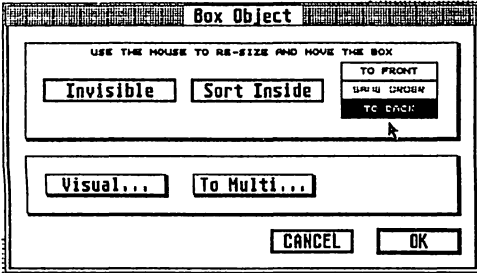
The next thing we decide to do in this Baseball Card Application is rate the players. One way of doing it may be adding Buttons to your HAP representing the different qualities of players you may get. For the purposes of our demo we created buttons for : Stars, Above (Average), Average, Below (Average), Minors (Minor Leaguer). The trick now is to select one, and only one, of these buttons per player. No problem! When you have a group of buttons together and only one of them should be selectable, we have a special designation for these particular buttons, "MULTI". When a button is "MULTI" within HyperLINK, it understands that it is merely one button of a group, and that only one of that group may be selected at any given time. By selecting another button, you are also de-selecting the first.

HINT ! If you are going to have more than one series of buttons on an application though, you **MUST** make each series Children of different Parent Objects. An easy way of accomplishing

THREE

this is to put a Box on the Application and then size it to hold all the buttons in a given series. This keeps Multi Buttons organized! So, let's

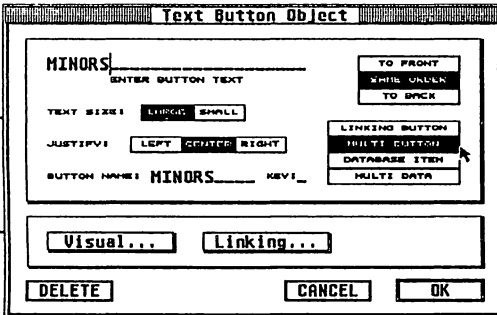
first make a Box around the entire area where our Player Rating Buttons are going to be placed. Since we have decided to place the box over an area that already has Text and Field Objects in it, remember to send this box To Back so that it does not hide the existing Objects.



(Figure 3.7)

Now that we have a Parent Box for our Multi Buttons, let's place them on our form. As you

can see from the example here, it is essential to mark each of these Buttons as "Multi" in order for them to work as intended. Also make sure to give each Button a name. Once again, if you choose to modify the visual appearance you can.

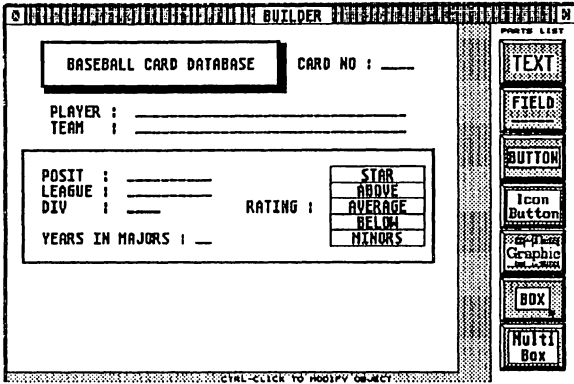


(Figure 3.8)

When originally putting these buttons on your application, if you did not make them "MULTI" it is still possible to do so without even having to edit any of the buttons themselves. To do this, double click on the parent Box Object of the Buttons. You can then configure this parent to be a Multi Parent by selecting the "To Multi" option. This automatically makes all the buttons within the parent "MULTI"s (In programmer's terms, this sets the buttons up as Radio Buttons). Don't forget then to give the First Button in your series an Item Name. Like naming a field, you have to

HYPERLINK

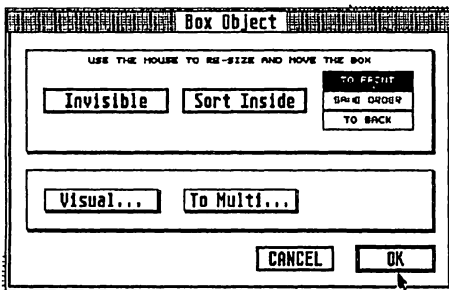
name the first button in the series when it is part of a MULTI. To satisfy this requirement, simply double click on the "Star" button and give it a name. In our case we used "Star" as the button name as well.



So far this is what we've created. It's a good start, but there are many other things you may wish to keep track of.

(Figure 3.9)

Another example of this "Multi" idea comes next. Suppose you get a Rookie Card. Everybody knows these are the ones that become valuable down the road! Why not set up another Multi button to mark a player card as either a Rookie card or maybe some other type of Special Card.

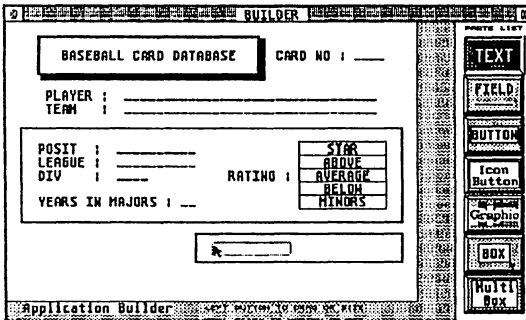


(Figure 3.10)

Let's approach this "Multi" from the angle of making the Parent Box a "Multi" and by leaving the Buttons alone. As you can see by the example, this was done by creating first a parent box...

Our next step is then adding a text entry. Because of the sizing of this Parent Box, I think our text at this point would look better in the Smaller Font Size. Therefore, drag a Text Object to within the Parent Box, enter the text content, and select SMALL

THREE

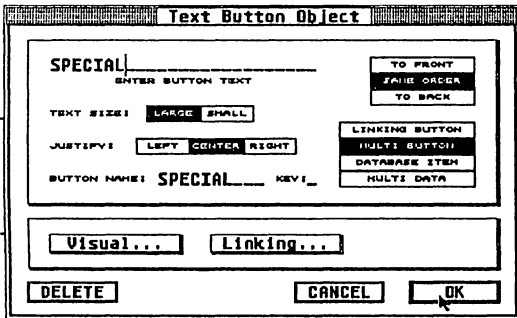


(Figure 3.11)

text size.

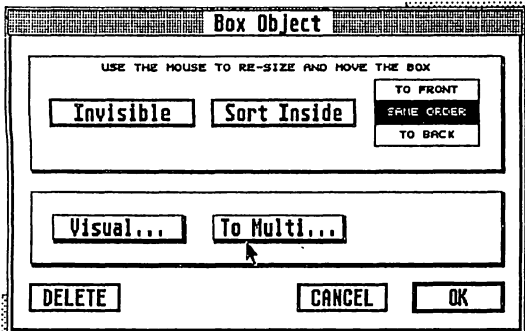
Here you can see the actual drgging of a Text Object from the Builder's "Parts List" onto your HAP in the position you desire. Here it is within the newly created box.

Next we have to create two more buttons : One to mark a card as a "Rookie" and the other to mark a card if it is "Special" in some other manner. So, drag one Button Object onto your form, label it "Rookie", size and position it properly, and then repeat these steps with another Button Object, this time labelling it "Special".



(Figure 3.12)

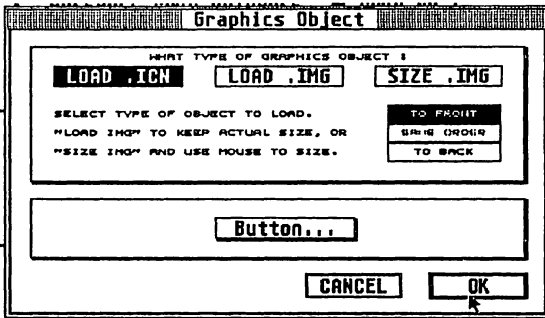
The next step is to go back and edit your Parent Box here, and signify to this Object that it is now a "Multi Box". To do this, either Double Click on the Box in question or <ALT> Click. This calls up the Box Object editing screen. All you need do within this screen is click on "To Multi".



(Figure 3.13)

HYPERLINK

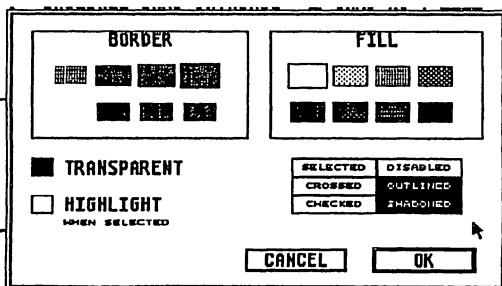
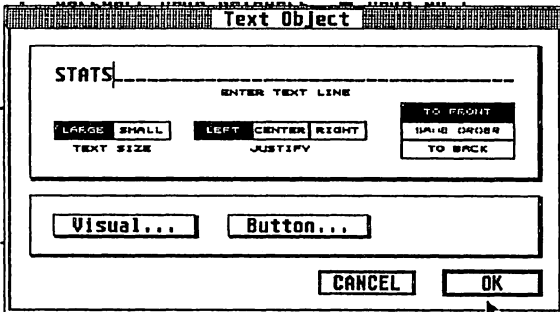
What else can we do with our Baseball Card Application? Well, there sure aren't too many stats showing on the screen. We could link this database to a text file that may have more information on the player. Okay, first let's place a graphic image (or Icon) onto our form which will represent a text file. To do this, simply drag the Icon Object to your HAP. Up will come an option for loading either an ICON or an IMG file. (The icon we select for use with our HAP at this time is "Book2", located in the sub-directory of 32x32 icons.)



(Figure 3.14)

(Figure 3.15)

Next, let's put a text label over the top of this icon to clarify to anyone using this HAP what this picture of a book is meant to represent. Drag a Text Object onto your form above the Book icon and enter "STATS".



(Figure 3.16)

The next step here is to put a Button Object directly on top of the text and icon. To allow for people to see the text and icon below this Button, we must then modify the Visual appearance of it, so that it is Transparent.

THREE

Now what??? Well now comes the fun part... Linking... The whole idea behind HyperLINK! We want this button to link to a text file. For this, after we have modified the Visual we go into the Linking setup screen. Here we find the answers to our needs. To accomplish the link we want, select the "Link to a HyperLINK App" option as the link type. Then on the other side of the linking screen we fill out the details on where we are linking to. Since we are linking to a text file we are really linking to the "EDITOR" HAP, so enter this information. Okay, but how does HyperLINK know which text file to link to. In the case of our example, it is linking to a text file that matches this baseball card. Hmmmm.... Card??? How about creating the link to a text file that matches the card number. Okay, to do this we would also select the "Get file name from

SELECT TYPE OF LINK

Link to a HyperLINK App (HAP)

Link to a HAP and do a SEARCH

Link to a HAP and pass data

COMMAND Link

SINGLE CLICK DOUBLE CLICK STARTUP

DESTINATION HAP INFO

LINK TO HAP NAME: EDITOR

USE CURRENT DATA FILE

USE FILE NAME: _____

GET FILE NAME FROM LOCAL FIELD

FIELD NAME: CARD PREFERENCE: _____

DESTINATION INDEX ONE: _____

LINK = B _____ H-LOCK: B _____ V-LOCK: B _____

CANCEL OK

(Figure 3.17)

local field" option on the Destination HAP Info. Then simply fill out the field name from which HyperLINK is getting the information to accomplish the link. In our example, this is CARD.

Now your link is almost ready to actually work. But wait!!! How is the link actually started? Well you still have to fill out the "LINK ON" information on the linking screen. This is pretty self-explanatory, all we have to do is select "SINGLE CLICK" on the linking screen and your link will be performed when you click just once with the left mouse button.

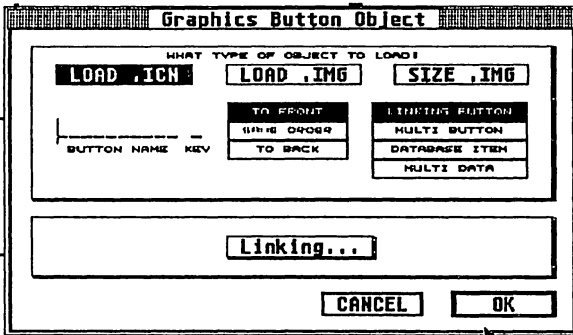
HYPERLINK

To explain in brief : when this button is selected while running the Baseball Card HAP, HyperLINK will take the information from the CARD field (ex. Card # 144), then look up using the EDITOR HAP a corresponding text file. (ex. 144.TXT) HyperLINK will then open the Editor Application and load in any text in this file.

ASIDE ! Thanks to the friendly folks at JMG, quite a few 32 x 32 and 48 x 48 icons have been included with your copy of HyperLINK.

Anything else we wish to include with our Baseball Card HAP? Since HyperLINK also comes with an IMG Graphics Module and Application, why not have scanned images of the cards load in? WOW... Another linking function!

Let's drag another icon onto our application... Seeing that we're going to showing pictures, how about using the camera as our icon! Here though, we're not just loading a Graphics Object, we're placing a "Graphics Button". Drag this object onto your HAP, and because it is going to be used

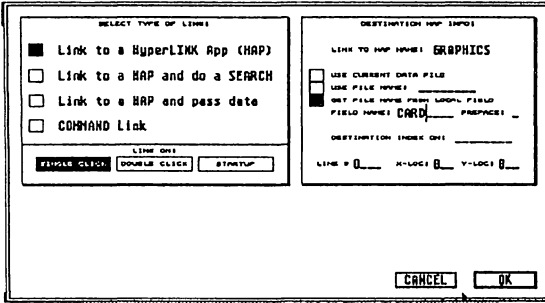


(Figure 3.18)

as a Button, make sure you give it a name. Then select LOAD .ICN, and select the Camera Icon from the directory of 32x32 icons. (That seems to be a good icon to represent a link to a picture!) The next step is to go to the

Linking screen. Similarly to the STATS button, we are creating a "Link to a HyperLINK App". This time though, the HAP name we are linking to is Graphics. After filling out this info, once again you must tell HyperLINK to "Get file name from Local Field" and that this local field is again CARD. Therefore, as the text file had to be

THREE



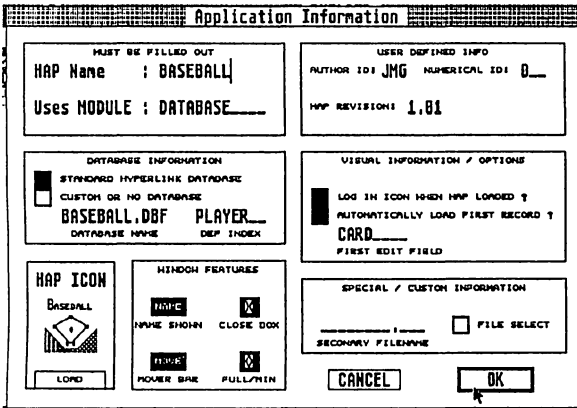
(Figure 3.19)

named 144.TXT, the IMG file has to be named 144.IMG. Lastly, make sure that you complete the "Link On" information by selecting Single Click.

Essentially all that remains to be done

within the Builder is the General Application Information. To get into this, click the right mouse button and select APP INFO.

Here we have to fill out...



(Figure 3.20)

HAP NAME :which is simply whatever you the user want to call your application. BASEBALL.HAP seemed appropriate for our use!

USES MODULE is the HyperLINK Module which this form uses. The

Baseball Card HAP being a database, it uses the DATABASE MODULE from HyperLINK.

DATABASE INFORMATION : Does your HAP use the standard HyperLINK (Dbase) type format; OTHER is not used unless your HAP is not a database. The BASEBALL.HAP therefore uses the standard format.

DATABASE NAME is the name of the data file

HYPERLINK

that an application uses by default. When your application is first started, if it does not find a file with this name, it will ask if you want to create a new one. Users should use a DBF extension on their file names, but this is not required. For the use of our demo, we look for a file called BASEBALL.DBF.

DEF INDEX is the default index (or sorting criteria) by which your database will be sorted when it is first called up. For the Baseball card HAP we use the NAME field for default indexing, and this index was called PLAYER when it was created. Now as we go through the records of players we see they have been arranged (or sorted) in alphabetical order.

HAP ICON is the icon that loads up in your HAP menu (use a 48 x 48 icon for this purpose). You can select LOAD ICON to change this away from the default. We designed a new icon just for our baseball application. Isn't it great!

WINDOW FEATURES By selecting or de-selecting, you change the characteristics of your application's window, ie. whether the window can be closed, minimized, moved, or whether the name appears on the window. In the Baseball HAP, all of these features are selected so that the window can be moved around, closed, etc.

AUTHOR ID is an optional part of the name that can be used as an identifier of who made the form. "JMG" are the initials of the wonderful corporate creator (cough, cough) of the Baseball Card HAP and, of course the HyperLINK program itself.

NUMERICAL ID is a unique number for each

THREE

application that can either be used or ignored. Future developments in HyperLINK may use this to a greater extent. Didn't need it for the Baseball HAP, so we ignored it!

HAP REVISION simply refers to the version number of your HAP. As you add more features and changes, you may want to change your version number so that users may reference the version of your HAP that they have. We modified the Baseball Card HAP a few times, so it ended up with a version number of 2.0. (Ooops, it's only 1.01 in the screen shot!)

LOG IN ICON ? When this is selected, an icon will appear in the HAP Applications Selector on the left side of the screen. If this is de-selected, then no icon will load in. We have it selected in our Baseball App.

AUTOMATICALLY LOAD FIRST RECORD is selected if you want your HyperLINK Application to automatically load the first record when it is called up. The Baseball HAP does not load the first record automatically.

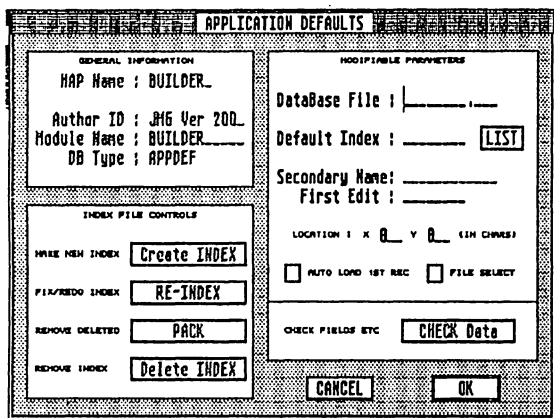
SECONDARY FILENAME is mostly there for future use. It would be used if a HyperLINK application used two files. This would be where the second file name would go. No need for this is our demo!

FIRST EDIT FIELD indicates the field in your database that HyperLINK will default to when you begin an edit. As a general rule, for esthetics anyways, you should use the first field on your form. Therefore the CARD field is the first edit field.

HYPERLINK

FILE SELECT brings up a file selector to select the data file your application is using when that application is called up. We won't be using it with the Baseball HAP.

You have just gone through the making of a HyperLINK Application! Not too tough is it? Now all you have to do is click the right mouse button and save your new application, then load it into HyperLINK using the "Load App" menu selection.



(Figure 3.21)

The last thing to do before using your HyperLINK Application comes up after you have loaded it into your HAP menu. Holding down the ALT key, click on your new application. This will call up the APPLICATION DEFAULTS screen. We have already

covered these earlier in the manual. With reference though to our Baseball Card HAP, maybe we should just mention again that this is the screen from which to create indexes for your database. If for instance you wanted to create an index by TEAM, this is the place!

Hopefully this step-by-step building of a HyperLINK Application has helped you understand the way in which HyperLINK operates. Further demos will be available in the future, either posted on GENIE, JMG's own BBS (416) 389-9064, which may help you grasp the potential of HyperLINK even more.

THREE

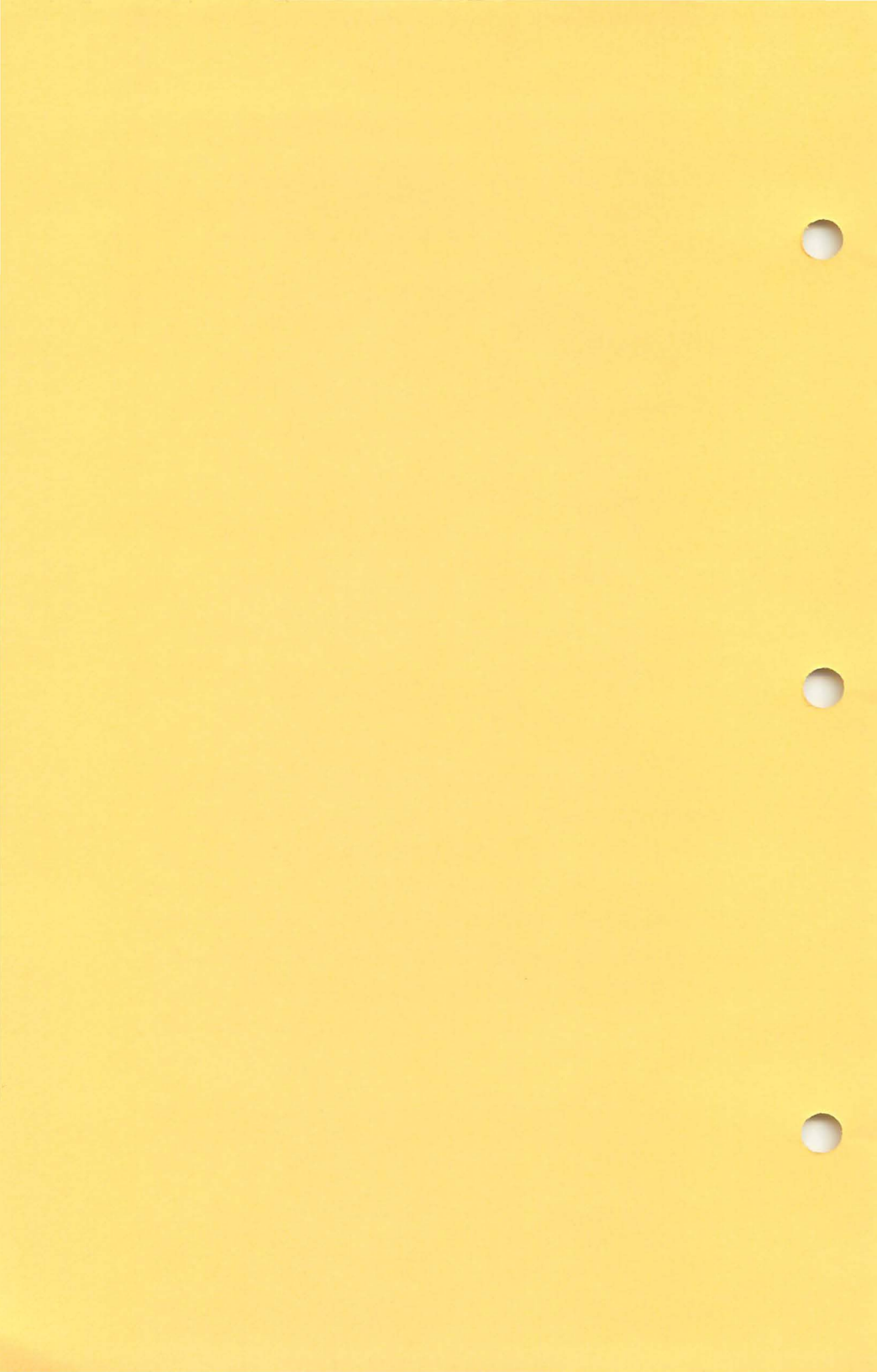
This Page Intentionally Left Blank



HYPERLINK REFERENCE

**This section is meant to provide users
with a more finely detailed look at
HyperLINK.**

**It is not to be the only part of the manual
you refer to. It is conceived to provide a
supplement to the previous sections.**



HYPERLINK REFERENCE SECTION

We've taken a general look at the many parts of HyperLINK in the *Anatomy* section of this manual. Now let's make a more detailed examination of all the ingredients that make up HyperLINK.

CONTENTS OF THIS SECTION :

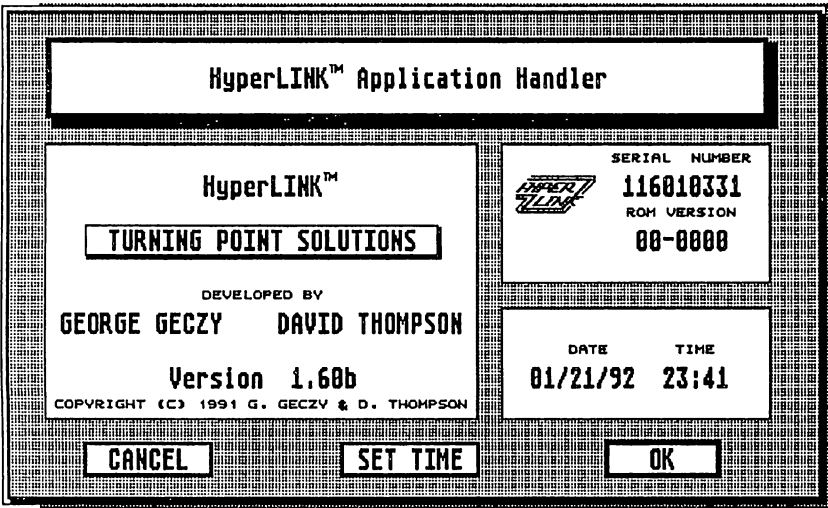
- PART A :** *The HyperLINK Desktop*
- PART B :** *Using the Builder & LINKING!*
- PART C :** *Using the Editor*
- PART D :** *Creating Text Links*
- PART E :** *Using the Report Generator*

FOUR

PART A :

The HyperLINK Desktop

When you Double Click on the HyperLINK program from within your GEM Desktop, you are first met with HyperLINK's Title Screen...



(Figure 4.1)

THE TITLE SCREEN: This screen contains the program credits of HyperLINK as well as some other useful tidbits

Some Notes about the Title Screen :

A) There is a hidden button over the HyperLINK Version number. This calls up another screen (see figure 4.2) giving you the revision numbers of all the HyperLINK parts you currently have. This "Revision Levels" screen provides you a reference on each individual HyperLINK

HYPERLINK

Revision Levels			
GEM Utilities	1.38	RSCLIB Loader	1.84
System Init	1.86	Main RSC	1.88
App Vars	1.85	Task Manager	1.88
HyperFORM Init	2.85	Clipboard Init	1.88
Ip PRT Driver	1.82	Report Printing	2.88
ACT Interface	1.87	DataBase Ctrl	3.83
DB Commands	3.89	Local DBase	3.85
IPC Handler	1.85		

(Figure 4.2)

component. In future, upgrades will be done on a modular basis, with less need to update the overall program.

B) On the mid to upper right of the Title Screen you can see the serial number of your HyperLINK copy. (NOTE :

If this number does not match the serial number written on the label of your original disk, please contact us immediately.) Also referred to here is the current ROM Version... You will notice that this number at the moment is 00-0000. Built in to HyperLINK is the ability to security protect it through the use of a Cartridge Key. This number refers to the Cartridge's ROM Version needed to run this copy of HyperLINK. A ROM Version of 00-0000 simply means that currently you do not need a security key to run HyperLINK.

C) Below the Serial Number area is displayed the current Date and Time of your computer's system clock. Some consider operating their computer under an accurate clock more important than others. Therefore, we felt it useful that HyperLINK display your current clock settings before you proceed any further. Remember, if you create or modify any files, they will be date stamped according to your system clock. To get any benefit out of the date stamping information then, it is important that you keep your system clock accurate.

D) The "CANCEL" button in your Title Screen gives you a chance to exit HyperLINK before proceeding any further. Selecting it at this point

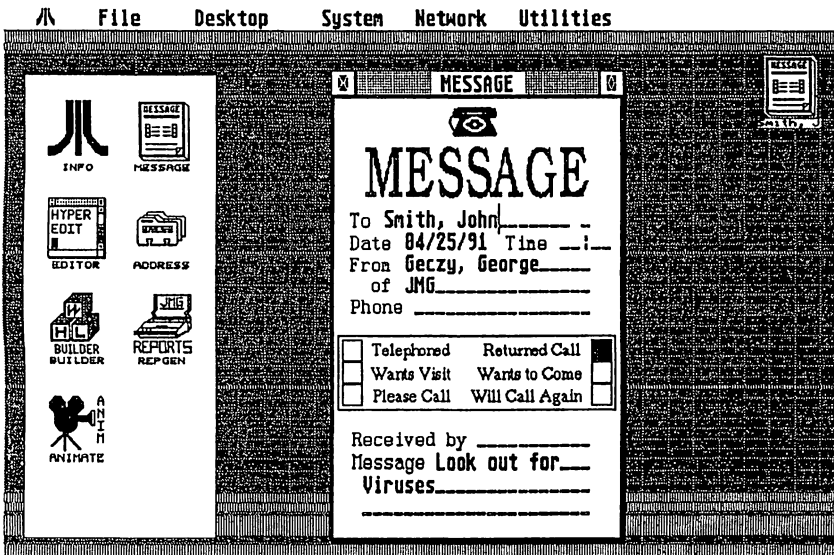
FOUR

returns you to your GEM Desktop. (Since the Title Screen can also be called up at any time during a HyperLINK session, you should note that the only time "CANCEL" quits the program is during the initial call-up of this screen. At other times it merely closes your Title Screen.)

E) "Set Time" gives you the opportunity to change the settings of your system clock if needed.

F) "OK" accepts the current system time, closes the Title Screen window, and proceeds with the loading of the HyperLINK program.

THE HYPERLINK DESKTOP : After the title screen disappears, you see flashing on the screen dialogue boxes telling you what HyperLINK Applications are loading in. This automated



(Figure 4.3)

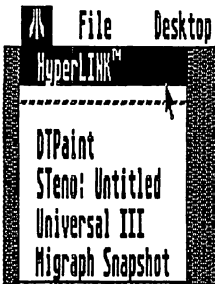
HYPERLINK

loading of applications is achieved very easily. When HyperLINK starts up it looks for two text files (in the same directory as the H_LINK.PRG file), one called DEFAULT.MOL ("MOL" stands for Modules Listing). This is just a listing of all the HyperLINK Modules to be loaded in. The second text file is DEFAULT.HAL ("HAL" stands for HyperLINK Application Listing). This file lists all of the HyperLINK Applications which should be loaded in every time the program is initiated.

Once these Modules and HAPs have loaded in, you find yourself face to face with the HyperLINK Desktop (The rest of Section A refers back to Figure 4.3).

Down the left side of the Desktop you will notice a series of seven icons, these icons represent HyperLINK Applications (HAPs) that have been loaded in. To run any of these HAPs, just click on them with the left mouse button.

Across the top of the Desktop screen you will see various Pull-Down Menus. Looking at them from left to right we find...



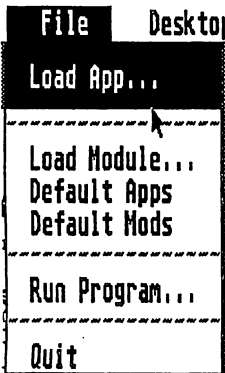
(Figure 4.4)

1) Atari's Fuji symbol represents, as always, the menu item under which any Desk Accessories you have loaded in may be found. Also, the first item, "HyperLINK " under this drop down menu allows you to at any time call up the HyperLINK Title Screen again.

2) The next menu is the "File" Menu. Here you can choose many of the operations that are file specific. There is a basic concept though that you

FOUR

should understand with regard to the File Menu : The difference between a HyperLINK Application (or HAP) and a HyperLINK Module... Essentially HyperLINK Applications are the end result of someone's use of the HyperLINK Modules. For instance, creating an Address Book Database is a specific application that uses the general Database Module within HyperLINK. If you then had a link to a scanned image of any person in your mailing directory, this would use the Graphics Module to display the picture. Confused? Don't be! To put it in simple terms, modules are the internal parts of HyperLINK which allow the creation and use of finished applications.



(Figure 4.5)

LOAD APP - This is used for loading other HyperLINK Applications (HAPs) into the main HyperLINK menu. You can get around having to load the same HAPs over and over though. If every time you use HyperLINK, you use essentially the same applications, you can create a default listing of all the HAPs to load in on program startup. (see Default Apps)

LOAD MODULE - Similar to **LOAD APP**, but this is for loading HyperLINK Modules that are not in the default list.

DEFAULT APPS - This first opens a window on your screen for the Text Editor, then reads a simple text file "DEFAULT.HAL" into it. It is simply a listing of all the HAPs to load in every time HyperLINK is started. It is important to note that if any of the HAPs you wish to load in are located in a different directory from the main HyperLINK program, then in this text file you must also specify that directory. From Figure 4.3, we can see that the default HAPs are :

HYPERLINK

INFO.HAP, EDITOR.HAP, BUILDER.HAP, ANIMATE.HAP, MESSAGE.HAP, ADDRESS.HAP, and REPORTS.HAP.

DEFAULT MODS - Once again, a Text Editor window is opened on your HyperLINK Desktop. This time though, the file loading in is "DEFAULT.MOL". This is a listing of all the Modules to load in whenever HyperLINK is started. To make things easier, this file lists all the Modules that were included in this version of HyperLINK and the only time you should want to modify this list is when more HyperLINK Modules become available.

RUN PROGRAM - This entry provides a door from within HyperLINK to run non HyperLINK programs. Consider as an example a situation wherein you were working on a database within HyperLINK, and then decided you had to get into your spreadsheet program temporarily to do some number crunching. This entry in the File Menu allows you to run any GEM Program, and when you quit it you will find yourself back in HyperLINK at the point you left with the same windows open and everything. (I don't have to tell most of you that this kind of memory handling can be finicky at times... Therefore, if you are going to use this aspect of HyperLINK, make sure that you have saved any data before proceeding.)

QUIT - The simplest entry in the File Menu to explain, this exits HyperLINK and takes you back to the GEM Desktop. (Or to any shell you may have started HyperLINK from. ex. Neodesk, etc)

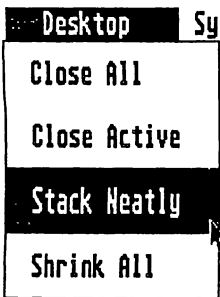
3) The next drop down menu is the "Desktop". This menu item within HyperLINK is used to

FOUR

provide control over the entire status of windows. That is, it is not merely to change any one Open (or Minimized) window, but all currently opened windows on the HyperLINK Desktop.

What is an Open Window? What is a Minimized Window? Well, when you click on any of the HAPs in the main listing down the left side of the HyperLINK Desktop (Represented in Figure 4.3 by the Main Menu Icons), you open this application into a separate window on the screen. You can have as many as seven open windows on the HyperLINK Desktop at any one time. However, only one of these open windows is on "Active" status. The window in the foreground of the desktop is the one HyperLINK is currently using. (Again referring to Figure 4.3, the Message HAP is currently open on the HyperLINK Desktop.)

You can also take any of these open applications, have them at any specific record you choose, and then temporarily store (or "Minimize") them so that they are not taking up a window space. To minimize a HAP, click on the button in the upper right hand corner of the frame of an Active HAP. You can see a Minimized HAP in our screen shot, along the upper right hand side of the illustration. This Minimized Application is also the Message HAP and if we click on this minimized icon, it will re-open the Message HAP at the same record as when it was minimized.



(Figure 4.6)

CLOSE ALL - This menu selection closes all Open Windows, and all Minimized Windows. It returns you to a "Clean" HyperLINK Desktop and lets you start cluttering it all over again.

CLOSE ACTIVE - This closes only the Active

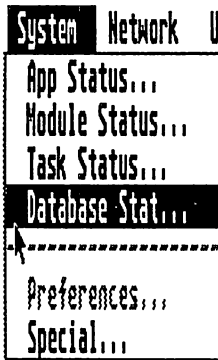
HYPERLINK

Open Window. That is, if you have three open applications, the one on top, or Active is the only one that will be closed.

STACK NEATLY - This arranges all Open Windows into a default order, helping to keep your HyperLINK Desktop less cluttered.

SHRINK ALL - This Minimizes all Open Windows. There is no real limit on the number of Minimized Applications you can have. With this ability to have records minimized, it allows you to switch quickly between various HAPs you may be working on.

4) Next we find the "System" Menu... This menu provides a location to look up the current status of various items in your HyperLINK session.



(Figure 4.7)

APP STATUS - This menu selection lists all the HyperLINK Applications that are currently loaded in and it also lists the data files your HAPs are using, as well as info on the Version and ID of each HAP.

MODULE STATUS - Like the APP Status, this lists all the Modules that HyperLINK currently has loaded in and likewise the Version and any ID info of each Module.

TASK STATUS - This Status item can give you an idea of how much of your computer's processing capabilities are being used by HyperLINK "Background Tasks". ie. Things HyperLINK is doing that are not necessarily obvious.

DATABASE STAT - Lists all the database files that are presently open within HyperLINK.

FOUR

When you first load HyperLINK, there are no database files open. When you open any HAP though, the specs on all database files associated with it are sent to this screen. Even after you may close a HAP, HyperLINK still remembers the details for any database file that has been open in this session.

PREFERENCES - While HyperLINK currently does not allow for Preferences to be set here, when this option is implemented however, it will work much the same as setting your Default HAPs does. That is to say, selecting this item will open an Editor Window and load in a text file listing your choices. Everytime HyperLINK starts up, it will look at this text file and configure itself according to your personal preferences.

SPECIAL - This is really just an empty Menu Item put in with future expansion in mind. With the ability to add more Modules to HyperLINK at any time to provide increased capabilities, we felt it very useful that the menu items be left some room to grow.

5) Next is the "Network" Menu. The items here need only be looked at if you are running one of the supported networks available on the ST. At the moment there is no accepted standard networking software in North America. We at JMG, are currently using HyperLINK networked under "Universal Network" but for more details please contact our company. We will circulating more networking specific documentation and information as it becomes available.

NETWORK SETUP - This is the menu item to select to configure HyperLINK to the commercial networking software you are using. Built in to



(Figure 4.8)

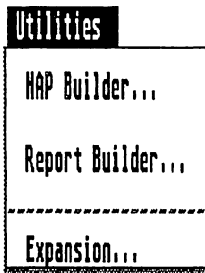
HYPERLINK

HyperLINK is support for Serial, Midi, and Cartridge port networks.

SEND MESSAGE - You will notice that this menu item is disabled. If you have a network setup and HyperLINK configured properly, then this item would be enabled, and selecting it would allow you to send messages immediately to any other station in your network.

CHECK NETWORK - This menu item also only becomes available if you have a network up and running (and of course properly set up within HyperLINK). If this is the case, then "Check Network" checks the integrity of you network, reporting if it experiences any problems communicating with any of the stations.

6) Next comes the "Utilities" Menu. At the moment this only provides some shortcuts for calling up the HAP Builder and the Report Builder. To illustrate this...



(Figure 4.9)

HAP BUILDER - This option opens a HAP Builder Window. While it is much faster just to point and click on the Builder Icon from your main HyperLINK menu, if the Builder was not loaded as a default HAP, then this allows you to LOAD HAP and then Open the HAP in one step.

REPORT BUILDER - Like the above entry for HAP Builder, this dose the same for the Report Builder (Generator). The Report Builder is far less likely to be among your default HAPs though, so you may find yourself using this selection fairly often.

EXPANSION - This is another menu item set aside for use by an add on HyperLINK Module.

FOUR

There is nothing specific planned here, but as more HyperLINK Modules are created, having an empty menu item or two may just come in handy.

Now that we have covered all the menu items from Figure 4.3, let's take a look at an open HAP. The one displayed on our Desktop (once again referring to Figure 4.3) is the Message HAP...

This "MESSAGE.HAP" represent a typical Open and Active HyperLINK Application. Currently we see that it is displaying a record of a call from "Geczy, George" to a "Smith, John". You can create your own HAPs to do anything you want, but one thing that can be noted about this HAP in particular is its use of Graphics as well as Text and Data Fields to give the HAP the look of an actual message pad.

In the upper left corner of an open HAP's frame, there is provided a universal "Close Window" button. Essentially, by clicking on this button you would close this window from the HyperLINK Desktop. For those of you more keyboard oriented than mouse oriented, there is an easy keyboard alternative, the function key <F10>.

Across the top of an open HAP might appear a Title Bar. This serves a dual purpose: It provides a location for the title of a HAP to be displayed, and in accordance with the GEM standard, it also provides a moving bar. That is, by clicking and holding down the left mouse button while the pointer is on this bar, you are able to move the entire window around the screen and place it wherever you wish.

HYPERLINK

There is one other feature to the frame of a HAP, in the upper right corner of an open HAP there is a button which performs a slightly different function from that which most of you would be familiar. Under a GEM window, this button would normally make your window full screen. Under HyperLINK though, this button Minimizes your open HAP. But to keep the GEM purists happy... If you hold down your keyboard's <SHIFT> key while clicking on this screen button, then your window does follow the GEM standard and become full screen. Don't be surprised though if this does not usually happen. To HyperLINK, "Full Screen" means the maximum size of the HAP as it was created in the Builder. Using the Message HAP as an example, if you <SHIFT> click to go to a full screen view, nothing will happen. This is because at it is showing is full size as it was created within the HAP Builder. (Two HAPs that do use this feature though are the Text Editor and the HAP Builder.)

Finally in Figure 4.3 we can see a "Minimized" HAP... As has already been mentioned, this icon represents a HyperLINK HAP that, after being opened, has been minimized to free a window space. HyperLINK is limited to seven open windows (this is a limitation of GEM itself) so minimizing a record is a good way of keeping a HAP at a specific record when you are not using it. By clicking on this minimized icon you re-open a HAP at the same point at which it was originally minimized. There is probably no better example of this than is seen by minimizing an animation. When you re-open an animation HAP it is still at the same frame.

FOUR

PART B :

The Application Builder

The Builder is the ingredient within HyperLINK that provides the program's extensive power. Here is where you create your Applications, and just as importantly, the Links between your HAPs.

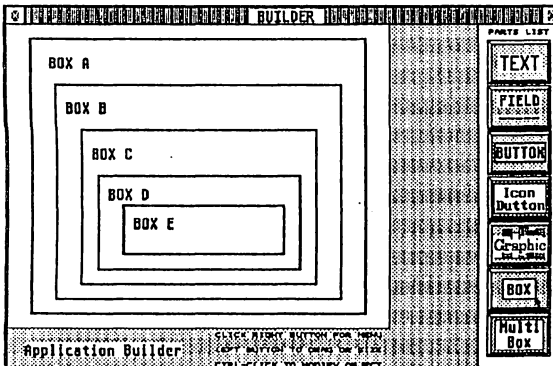
(Before proceeding further in this section, you may wish to refresh yourself with the principle of LINKING. We recommend that you read Appendix A, which provides an analogy of the linking theory using a simplistic "Shipping/Receiving" concept.)

PARENTS VS CHILDREN

As was touched upon briefly in the Anatomy and Step By Step sections of this manual, the concept of Parents and Children is very important when it comes to the Builder. To illustrate the difference, refer to Figure 4.10. In this, the Builder form contains a series of boxes. (Boxes A-E) Box A is the parent to Box B

which is the parent to Box C, etc... It simply shows that an object is a Parent if it contains another Builder "Part". Only the parts "Box" and "Multi Box" can be parents, while all the items on the parts list could be Children. Again referring to

(Figure 4.10)



HYPERLINK

Figure 4.10, rather than say Box A is the parent of Box B, you could say that Box B, is the child of Box A. Hopefully this makes it clear to you that there is no limitation upon a child having children objects of its own.

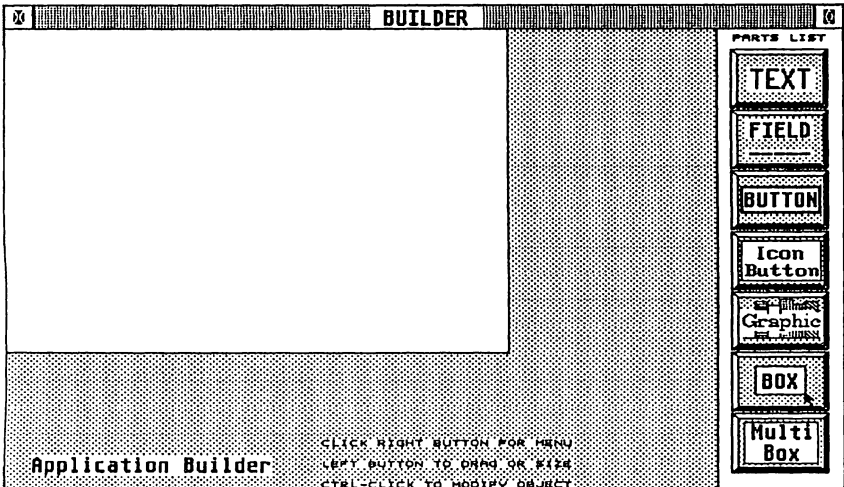
Some rules about children and parent objects... Once they are placed within a parent, that is where a children object expects to stay. If you try to drag a child object out of its parent, you will notice it unable to leave the parent's boundaries. To signify to HyperLINK that you wish to change the "family tree" as it were, and remove a child from its current parent, simply hold down the <SHIFT> key as you drag the child. HyperLINK then is watching for a change in parent status and allows you to move the child wherever you wish. Also, if you move an object into a location whereby it would become a children object, HyperLINK will enquire if you want the object's parentage status to change. Even though you may move a Text Part from another part of your Builder form into a Box, you still have the option of whether or not this becomes a child of the box.

One of the ways parent objects can be useful when you are creating an application is in the design of your HAP's layout. If you decide to move a group of fields or text around on the form, (trying to make your design a little better), when you move a parent object, it moves all of its children parts right along, actually keeping them in the exact same space relative to the parent.

While you may not totally understand this concept of "Parents" and "Children", play around inside the Builder, dragging an object into a Box, and see how HyperLINK handles it. It won't be long before the idea becomes clear.

FOUR

The Builder is comprised of many different components, or as we will refer to them, the "Parts" or "Objects". You will notice as you first enter the Builder screen, that up will pop a blank form with the "Parts List" running down the right side. You transform this blank form by simply dragging any of these parts to the form, in the location you wish it to appear. Quite quickly you will notice your application beginning to take shape. If you decide afterwards to move this object, simply click on it and hold the left mouse button down while you drag it to its new location.



(Figure 4.11)

Much of the remainder of Part B refers to this illustration...

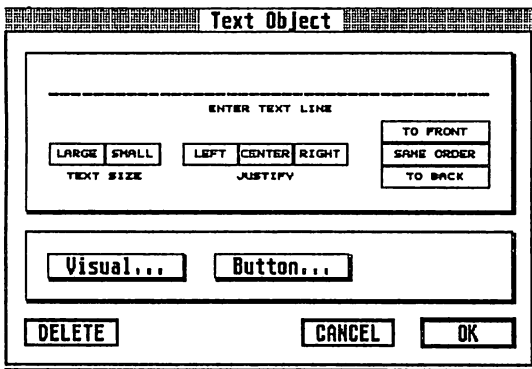
The Blank Workspace - There is not a whole lot to say. This is simply the working area in which to create your HyperLINK applications. While the window displayed is the default size of your form, this is not fixed. You can re-size your workspace by clicking and holding the left mouse button on the lower right corner of your form. (In

HYPERLINK

Figure 4.11, the Blank Workspace is the large white rectangular space.)

Now let's take a look at the Builder's "Parts List"...

TEXT PARTS - A text part, while often essential in creating a HAP, is inherently a "Sit around and do nothing" part of any form. It merely provides a description of other objects in your HAP, or of your HAP as a whole. When you drag this object to a Builder Form, the Text Object Dialogue Box comes up (Figure 4.12). Taking you through it line by line...



(Figure 4.12)

ENTER TEXT LINE : This is the text that will appear in your form. Enter it exactly as you wish it to appear. If you enter a blank line, HyperLINK assumes you had made a mistake and does not place a blank text object on your form.

TEXT SIZE : You have a choice between two font sizes when you are using text in the Builder. The standard screen font size you are used to seeing is actually the LARGE selection of these two, while the SMALL selection is a font approximately half the size. LARGE is the default setting, so if you want a smaller font, don't forget to change this.

JUSTIFY : You have three choices to select from, "LEFT", "CENTER", or "RIGHT". These are formatting options for how your text will

FOUR

appear in relation to the object it is contained within. For instance, should your text entry only be three characters long, but the sizing on your form for this text object be 8 characters long, you have the option of setting the text flush with the left margin of your object, flush with the right margin of your object, or centered within both margins. The default setting is LEFT as this is the most commonly preferred.

TO FRONT / SAME ORDER / TO BACK :

This allows you the option of setting the text on your form in relationship to other objects that may already be there. As an analogy, consider the making of Lasagna. You put down one layer of pasta, then a layer of meat, then a layer of sauce. As you put down each new layer, it hides from sight everything below it. Any new object placed on a form is automatically at the front. When an object is in Front of another, it does usually hide anything below it however if the out front object is made TRANSPARENT through the Visual configuration screen, then you would be able to see objects below. As you experiment with HyperLINK you will find some good uses for this feature, maybe not so much with Text Objects, but with backgrounds having Invisible Button overlays.

VISUAL : Clicking on this button, sends you to a screen discussed a little later (see Figure 4.19) in this section. If you wish to change the Border around your Text Object or its Fill Pattern, you would do so on the Visual Screen.

BUTTON : Selecting this item in the Text Object screen transforms the basic characteristics of any text entry you may have made. If you select Button, your Text Object is replaced by a Button

HYPERLINK

Object with the same Visual Characteristics and actual text content. (See **BUTTON PARTS** for more details.)

DELETE : When you first drag a Text Part to the Builder Form, this Delete button is hidden on the Text Object screen. However, if you Double Click on any Text Item already in place on your form, this Text Object screen reappears allowing you the ability to edit the content, the appearance, etc. or even **DELETE** the item from your HAP.

CANCEL : When you are first creating a Text Object for your HAP, **CANCEL** does exactly that. It is like telling HyperLINK; "Forget it, I changed my mind." If the Text Object has already been created and you've called this screen back up to edit the object, **CANCEL** will merely ignore any changes you may have made and return you to the Builder.

OK : Once you've created or edited a Text Object to your satisfaction, select **OK** to confirm the settings to HyperLINK and actually place the object on your form.

(Figure 4.13) **FIELD PARTS** - These represent Database Fields that you are going to be placing in your HAP. Not only can data fields contain the information that makes your application worthwhile, but they are also capable of creation action, in way of links, to other HAPs

Field Object

FIELD LENGTH: ___ DECIMALS: ___

TYPE: **ALL** **CAP** **NUM** **DAT**

FIELD NAME: | _____

TO FRONT
SAME ORDER
TO BACK

Expression

Visual... **Linking...**

DELETE **CANCEL** **OK**

FOUR

as well. When you drag a Field Object to your form, up pops the Field Object Dialogue Box (see Figure 4.13).

FIELD LENGTH : This is the number of characters to allow for your data field. Make sure that your form is large enough to contain the field length you select. **NOTE** : You must give this a value greater than zero (unless it is going to be a DAT field), otherwise HyperLINK will ignore your entry.

DECIMALS : You need only complete the information for this field if your data field is going to be a numerical value. This tells HyperLINK how many decimal points you want displayed in the data field. If your data field is a numerical value, and you leave the decimal entry blank, HyperLINK will assume you want a floating decimal point.

TYPE ALL / CAP / NUM / DAT : This tells HyperLINK what your data field is going to be. An entry that can accept any keyboard character (ALL), a field that will only accept Capital Consonants (CAP), a field that will only accept numerical values (NUM), or a date (DAT) field. In the case of the date field, when you select this option HyperLINK automatically places the field in a / / format.

FIELD NAME : Like any database program, HyperLINK insists that data fields have a name. Here you enter the name for the field you are creating. If you leave this entry blank, HyperLINK will generate its own name for your field. How HyperLINK does this is by merely keeping count of how many objects you've placed in your form. If the sixth item you are placing on

HYPERLINK

your form is a data field and you fail to give it a name in this screen, when you go back to edit your field you will find that HyperLINK has named it FIELD6. This is not a very descriptive name! Therefore you should get in the habit early of remembering to create names for all your data fields.

TO FRONT / SAME ORDER / TO BACK : As this item has already been explained under the TEXT PARTS, please refer back to this section if you need an explanation. Regardless of whether a part is Text, Field, Icon, etc. this function behaves the same.

EXPRESSION : This does two things... First it marks this data field in your HAP as an expression. Therefore this field will no longer be editable when running your application because it expects to receive its content through the formula or function set. Second, by selecting this button you call up an additional screen (see Figure 4.20) where you can enter data expressions for your current field. These expressions can be mathematical formula, search routines, and more. It is through the completion of this expression that your data field expects to receive its contents.

VISUAL : See Figure 4.19 for more details on the screen this selection calls up.

LINKING : Data Fields can be configured, so that when you click on them with the left mouse button, they may "Link" to other HyperLINK Applications. If you wish to design a link from a data field, you would click on this button in the Field Object screen. It then call up your main linking screen (See Figures 4.21 - 4.24).

FOUR

DELETE / CANCEL / OK : Once again, these are options common to most of the Building Parts screens. If you need a refresher course on their functions, see each individual item under the TEXT PARTS description earlier in this section.

BUTTON PARTS - A Button within HyperLINK is where most of the action is! Buttons are used to initiate linking procedures, characterize records, provide information, and more. When you drag a Button Object to your form you are met with the standard Text Button Object screen (Figure 4.14).

The screenshot shows a dialog box titled "Text Button Object". It contains several sections: "ENTER BUTTON TEXT" with a text input field; "TEXT SIZE" with "LARGE" and "SMALL" buttons; "JUSTIFY" with "LEFT", "CENTER", and "RIGHT" buttons; "BUTTON NAME" with a text input field; a group of buttons including "TO FRONT", "SAME ORDER", and "TO BACK"; another group including "LINKING BUTTON", "MULTI BUTTON", "DATABASE ITEM", and "MULTI DATA"; and a bottom row with "Visual...", "Linking...", "DELETE", "CANCEL", and "OK" buttons.

(Figure 4.14)

Object though, any text entry here is optional. It is not necessary to have a text label within a button.

TEXT SIZE (SMALL / LARGE) : If you are putting a text label within your button, you have to choice of whether to use the standard LARGE font, or replace it with the SMALL font size.

JUSTIFY (LEFT / CENTER / RIGHT) : Once again, these are options common to most of the Building Parts screens. If you need a refresher course on their functions, see each individual item

HYPERLINK

under the TEXT PARTS description earlier in this section.

BUTTON NAME : HyperLINK needs to keep track of buttons much the same way as it would a data field. Therefore, it requires your buttons be given field names as well. Once again, if you don't enter a name of your own choosing, HyperLINK will generate one for you.

TO FRONT / SAME ORDER / TO BACK : Once again, these are options common to most of the Building Parts screens. If you need a refresher course on their functions, see each individual item under the TEXT PARTS description earlier in this section.

LINKING BUTTON / MULTI BUTTON / DATABASE ITEM / MULTI DATA : These four are the settings you use to designate the type of button you are creating. There are different character traits that go along with each selection....

LINKING BUTTON - This is the Default Button type. When it is set, you are indicating that HyperLINK will perform an action of some sort, (associated with the linking screens) everytime this button is selected from within a HAP. To just reiterate a little, this button provides ACTION to a HAP.

MULTI BUTTON - A Multi Button assumes the presence of more than one of this type of button within the same parent object. This is a Database item generally, and not an Action button. How it works is very straightforward. Consider a survey form sent out by a magazine to determine its readership base. On it, they may not want your

FOUR

exact age, but rather to know which age group you fall in : Under 19, 20-25, 26-35, 36-45 etc. The way you could do this within HyperLINK is create a Parent Box first, then within this box place a "Multi Button" representing each age range. When the HAP is being run, the selection of any one of these age buttons, cancels any selection previously made. In other words, if you clicked on "20-25" then clicked on "26-35", not only would the "26-35" button highlight, but the "20-25" would at the same time become de-selected. As was briefly mentioned, a Multi Button is stored as a database item within your HAP. One data field is kept, indicating which of the buttons are selected with each record. If you have four Multi Buttons within a parent, the database would keep track of a numerical field with a value of 0-4 for each record (0 being none of the Multi Buttons selected).

DATABASE ITEM - Generally this type of button is essentially a Yes/No type of data item. Again, like the Multi Button, a Database Item Button would provide the information to a field within your HAP. If one of these buttons is selected, this would be recorded as a YES in your record, and likewise if one of these buttons were not selected, this would be recorded as a NO.

MULTI DATA - As the name suggests, this is a combination of the Multi Button concept and of the Database Item. A Multi Data Button is BOTH a Multi Button, therefore a group of buttons wherein only one can be selected at any given time, and a Database item which keeps a field in your record for each of the buttons. To go back to our example of the ages ranges... If these were Multi Data Buttons, your database would keep a Yes/No field for each of the buttons in all your

HYPERLINK

records. While this designation does not really do a lot in a HAP, it can provide a better Report capability.

VISUAL : (See Figure 4.19)

LINKING : (See Figures 4.21 through 4.24)

DELETE / CANCEL / OK : Once again, these are options common to most of the Building Parts screens. If you need a refresher course on their functions, see each individual item under the TEXT PARTS description earlier in this section.

ICON BUTTON PARTS - This allows you to load IMG files or ICN files onto your HyperLINK form to fancy up the appearance of your HAP as well as at the same time provide a useful purpose to your application. (See Figure 4.15)

Graphics Button Object

WHAT TYPE OF OBJECT TO LOAD?

LOAD .ICN LOAD .IMG SIZE .IMG

TO FRONT LINKING BUTTON
SAME ORDER MULTI BUTTON
TO BACK DATABASE ITEM
MULTI DATA

BUTTON NAME

Linking...

DELETE CANCEL OK

(Figure 4.15)

LOAD .ICN : When you are loading a graphic, you have the choice of file types between ICN files and IMG files. ICN files are icons created in any one of a number of Icon Editors, from the Resource Construction set, to Degas (with an Icon template loaded in), etc. We provide you

with a starting library of icons (32x32 pixels & 48x48 pixels) from which to select. When you choose this selection, you are faced with your everyday file selector, and you choose which icon you wish to load in. (The icons we provide are located on Disk 2 in sub-directories "ICON.32" and "ICONS.48") NOTE : As of the release of

FOUR

version 2.0 of HyperLINK, our own Icon Editor was going through final Beta Testing and is scheduled to be released soon.

LOAD .IMG : IMG file formats are the standard for scanned images in the ST world. This graphic format is supported by such software as Touch Up, Easy Draw, DT Paint, and many more. When you choose to load an IMG file, it places the entire image, in your builder at the location you selected. With this selection, you have no control over the sizing of the image. It will load in, it's actual size. If there is not enough room in your current form, some of the image will be lost. Therefore it is important to make sure your HAP form is big enough to display the image you are loading in. If it is not...

SIZE .IMG : The provides you with an alternative when loading in large IMG files. If their actual size is larger than the size you want appearing in your form, then you would select "SIZE .IMG" instead of "LOAD .IMG". After you have selected which IMG file to load in, you find yourself back on the main Builder screen sizing your image. Because HyperLINK has to make a little extra effort to re-size an IMG to your specifications, this option takes a little longer than your standard "LOAD .IMG". (NOTE : Once an IMG is in your form there is no way to alter its size. Therefore, you must delete the IMG Object and re-load it in, trying this time to get the sizing more to your liking.)

TO FRONT / SAME ORDER / TO BACK : Provides you once again with positioning control of your object as it relates to other items already in your HAP.

HYPERLINK

BUTTON NAME : When you are using this "GRAPHIC BUTTON", you are really just using a button with a fancier appearance than normal. As such, you must fill out all the information here, that you would under the regular button screen. Remember, HyperLINK needs a name for buttons, so if you don't give your button a name, HyperLINK will.

LINKING BUTTON : This is more than likely, the button type of any Graphic Button. When this is selected you are telling your HAP to perform an action you have set up in the Linking screen. For more details on the various button types, see **BUTTON PARTS** (Figure 4.14) earlier in this section

MULTI BUTTON : This would imply that you have a group of Graphic Buttons within the same parent, providing data info for your HAP with only one of them selectable at any given time. (For more details see Figure 4.14 - **BUTTON PARTS**.)

DATABASE ITEM : This would allow you to select or de-select a graphic image or icon to represent a Yes/No type record in your database. (For more details see Figure 4.14 - **BUTTON PARTS**.)

MULTI DATA : Again, this would designate a group of IMG's or Icons as a data entry in your application. While there is a data field for each graphic, only one would be selectable at a time. (For more details see Figure 4.14 - **BUTTON PARTS**.)

LINKING : Really only useful if your graphic is a Linking Button, here you would configure the

FOUR

action you want your HAP to take when the picture is selected. (See Figures 4.21 through 4.24 for more details.)

DELETE : Just to reiterate what was covered in **TEXT PARTS**, when you first drag any object part to the Builder Form, this Delete button is hidden on your editing screen. However, if you Double Click on any Item already in place on your form, the appropriate editing Object screen reappears allowing you the ability to change the content, the appearance, etc. or even **DELETE** the item from your HAP.

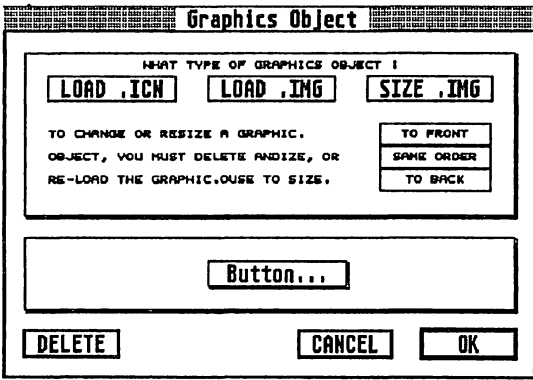
CANCEL : When you are first creating any Object for your HAP, **CANCEL** does exactly that. It is like telling HyperLINK; "Forget it, I changed my mind." If your Object has already been created and you've called this screen back up to edit the object, **CANCEL** will merely ignore any changes you may have made and return you to the Builder with your original part still intact.

OK : Once you've created or edited an Object to your satisfaction, select **OK** to confirm the settings to HyperLINK and actually place the object on your form.

GRAPHIC PARTS - You will notice a distinct similarity between these Graphic Objects (Figure 4.16) and Graphic Button Objects (Figure 4.15). The difference is that a Graphic Object is meant only to improve the appearance of your HAP and not provide any added feature, such as linking or database info.

LOAD .JCN : These three buttons across the top of the Graphic Object screen simply tell HyperLINK the format of the image you are

HYPERLINK



(Figure 4.16)

going to be loading in. This entry selects an Icon format.

LOAD .IMG : As with "LOAD .ICN", once again you are telling HyperLINK the format of the picture you wish to place in your HAP. This entry selects the ever popular IMG file format.

SIZE .IMG : While you are here also selecting an IMG file format. By selecting this option, you are given the ability to configure your own sizing of the image on your HAP. Just "LOAD .IMG" brings in the IMG file in the screen size it was created. If you wish to enlarge or shrink this original graphic, then you would select this item.

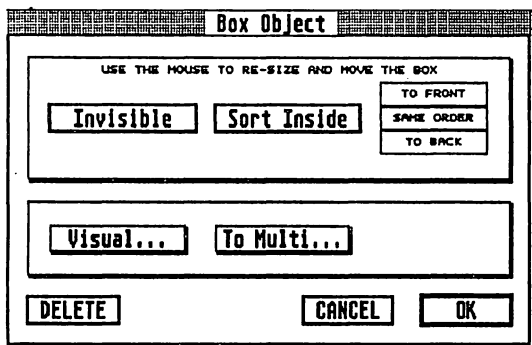
TO FRONT / SAME ORDER / TO BACK : Once again, merely a tool to provide flexibility in how this object relates in position with other objects already placed in your application.

BUTTON : If after you load a graphic into your HAP, you decide that you would like to make it a GRAPHIC BUTTON (see Figure 4.15), selecting this option will do that for you. Remember, once this is done, you can not go back to regular Graphic status.

DELETE / CANCEL / OK : These items have been adequately covered under previous Building Parts.

FOUR

BOX PARTS - A Box Part is simply a rectangular graphic which you can place, size, and control the appearance of, anywhere on your HAP. It provides a very useful tool as a Parent to Multi Buttons (see Figure 4.15) and it can be used also just to improve the overall look of your HyperLINK Application. When you drag a Box Part to your form, up pops the Box Object Dialogue (Figure 4.17).



INVISIBLE : If you select this Invisible option, you are simply taking a short cut on the Visual Appearance of your Box. Invisible sets the appearance to be Transparent Fill with No Border. If you go back and edit an Invisible Box (ie. deselect

(Figure 4.17)

Invisible), you will notice that nothing has changed in your form. NOTE : To change your box back to Visible, not only do you have to turn the Invisible Button off, but then you must go in to the Visual Screen (See Figure 4.19) and select either a visible Border, or a visible Fill.

SORT INSIDE : This selection is not available when you are first creating a box, it only becomes available upon editing. By selecting "Sort Inside" you are arranging the order of any children objects within your box. Order is first determined within a parent by the placement order on your form. Therefore, if when you were originally creating your HAP, you placed fields in your form in a different order than they ended up being placed, with this selection you can re-arrange the order children fields will be edited in. (You will

HYPERLINK

notice if your Field order needs sorting right away when running your HAP. Your cursor might move down two fields, then up three, then down four... etc, when you are entering data.)

TO FRONT / SAME ORDER / TO BACK : These entries have been covered on several occasions already in this section of the manual. Please refer back, if you need details on this concept.

VISUAL : (See Figure 4.19)

TO MULTI : This button transforms your Box Object into a Multi Parent Box Object, with all the characteristics associated therein. (For more details see Figure 4.18)

DELETE / CANCEL / OK : See earlier PARTS in this section if you are still unsure of these selections.

MULTI BOX PARTS - This designates a specialized Box Object, marking any children buttons within it as Multi Buttons (See Figure 4.14 for more info on Multi Buttons). The use of

(Figure 4.18)

The image shows a dialog box titled "Multi-Parent Box Object". At the top, it says "USE THE MOUSE TO RE-SIZE AND MOVE THE BOX". Below this, there are two buttons: "Invisible" and "Sort Inside". To the right of these is a vertical stack of three buttons: "TO FRONT", "SAME ORDER", and "TO BACK". Below the "Invisible" and "Sort Inside" buttons is a text field labeled "DATA ITEM NAME:" followed by a series of underscores. Below the text field is a line of text: "THIS OBJECT IS TO BE USED AS A PARENT TO MULTI-TYPE BUTTON OBJECTS ("RADIO BUTTONS").". Below this is a button labeled "Visual...". At the bottom of the dialog box are three buttons: "DELETE", "CANCEL", and "OK".

a Multi Box can often save you time and effort in the creation of your HAP. Rather than set all your buttons to Multi, you need merely place them in a Multi Box and this classification is made automatically. When a Multi Box is dragged to a form, the Multi Parent Box

FOUR

Object Dialogue appears (Figure 4.18).

INVISIBLE : This selection is identical to that in a regular Box Object. It is a short cut to be used in lieu of going to the Visual screen (Figure 4.19), setting the Border to NONE, and the Fill to CLEAR.

SORT INSIDE : This sets the editing order within a parent to match the order of the fields as they appear on screen, regardless of the order in which they were placed on your form.

TO FRONT / SAME ORDER / TO BACK : See previous PARTS for a description on these placement functions.

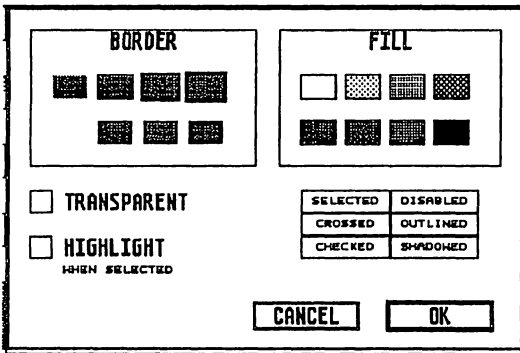
DATA ITEM NAME : Unlike a regular Box Object, a Multi Parent Box Object requires a field name. This is because it is exercising control over its children objects and providing database information as well. Remember, when HyperLINK requires a field name and you don't give it one, the program will generate its own.

VISUAL : (See Figure 4.19)

DELETE / CANCEL / OK : If still in need of clarification on these functions, see "TEXT PARTS" earlier in this section.

THE VISUAL SCREEN - (Figure 4.19) The appearance in your HAP of Text, Field, Button, Box, and Multi Box objects can be edited to your own liking. In the Visual Screen (called up from the Object Editing screens just listed), you control the Border around your object, the Fill Pattern of your object, and more... NOTE : You can not

HYPERLINK



(Figure 4.19)

change the Visual Appearance of Icon Buttons, or Graphic Objects within HyperLINK.

BORDER : The border of your objects can be set to various thicknesses or to be non-existent. Text and Field Objects default to an invisible border,

while Button, Box, and Multi Box objects default to a visible border. Of course you can change the Border Appearance of your object simply by clicking on the border selection you wish.

FILL : All objects default to a white Fill Pattern, but with boxes in particular, you may wish to change that. Click on the new Fill Pattern, and that's what your object will use in the HAP.

TRANSPARENT : Primarily used for buttons, this allows you to place one object over top of another object in your HAP and still allow the object below to be seen.

HIGHLIGHT : When this option is enabled, your HAP will display an object in reverse colour when it is selected.

Next to be described in the Visual Screen are a series of Six Buttons. These buttons are similar to ones found within Atari's own Resource Construction Set. While there is not much use for half of these buttons, we included them anyway...

SELECTED : When you choose this option, you are putting your Object into a reverse video. That

FOUR

is, for a Text Object you will find the text now white, and the fill, black. This is the only way of reversing the video colour of text within HyperLINK.

CROSSED : Probably one of the least useful of selections... When you configure your object as CROSSED it puts an X through your Object from corner to corner. If there is no content, (ie. Fill or Text) then you will not even see this Cross.

CHECKED : This option displays a Check Mark in the upper left corner of your Object. Once again, while no use for this becomes rapidly apparent, it is there in case you ever need it.

DISABLED : This provides a very interesting effect that you can use within HyperLINK. While normally your choice of text colour is black (or white on a black background if you have "Selected" your object), using DISABLED you can achieve a scaled grey effect on your HAP.

OUTLINED : This adds an external Border around the outside of your object. It works best with Boxes containing a Fill Pattern other than clear.

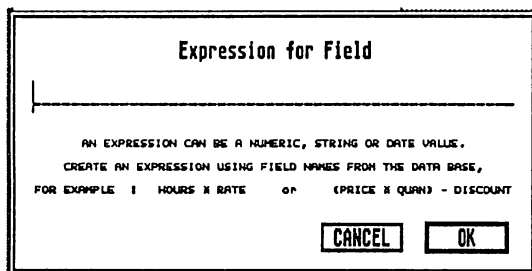
SHADOWED : This effect also works best with Boxes but is independent of your Fill Pattern. It provides a Shadow going down and to the right of your object, and is quite a useful tool in improving the appearance of your HAP.

CANCEL : Ignores any changes you may have made to your object's appearance and drops you back to whichever Object Editing screen you were first in.

HYPERLINK

OK : Accepts any changes you have made to the physical appearance of your object, and takes you back to that object's editing screen.

THE EXPRESSION SCREEN - (Figure 4.20)
This screen is accessed through the Field Object Editor. Its purpose is to allow you to create mathematical formula, or other search criteria, and to send the results to the data field which is marked as an expression.



Expression for Field

AN EXPRESSION CAN BE A NUMERIC, STRING OR DATE VALUE.
CREATE AN EXPRESSION USING FIELD NAMES FROM THE DATA BASE,
FOR EXAMPLE: 1 HOURS X RATE OR (PRICE X QUAN) - DISCOUNT

(Figure 4.20)

EXPRESSION FOR FIELD : An expression can perform mathematical calculations based on any combination of other fields in your HAP, or given numerical values. For instance, if you

were to create an Inventory HAP, listing your stock, and the prices you paid for each item, you could have an expression field making a recommendation on a selling price based on whatever factor you enter. Such an expression could resemble "COST x 1.5". (COST would have to be one of the other data fields in your HAP) This formula would give you a 1.5 Markup (or a 33% margin) on your sales. Using the same HAP, you may have another data field which gives you your total cost value of any inventory item. "COST x STOCK" (where STOCK is a numerical field listing how much of any product you possess.) When a data field is using an Expression, it can only get its data from the successful completion of the expression entered. HyperLINK does not allow you to mark a data

FOUR

field as an expression and still go in and change the calculated value.

CANCEL / OK : After you have entered your expression you can either choose **CANCEL** to have HyperLINK ignore you settings, or **OK** for HyperLINK to keep whatever expression you may have just entered.

THE LINKING SCREENS... While there is only one linking screen within HyperLINK, the options within that screen change as you select different types of LINKs. Therefore in this section we will treat the linking screen as if it were actually four distinct screens (Figures 4.21 - 4.24). The differences between the linking screens do not appear until you select the Type of Link. (NOTE : If you need a refresher on the basic theory behind the linking, see Appendix A.)

LINK TO A HAP - (Figure 4.21) This is the most straightforward linking screen that exists. You would select it if you were wanting to perform a straight link calling up another HAP.

(Figure 4.21)

SELECT TYPE OF LINK

Link to a HyperLINK App (HAP)

Link to a HAP and do a SEARCH

Link to a HAP and pass data

COMMAND Link

LINK ON:

SINGLE CLICK DOUBLE CLICK STARTUP

DESTINATION HAP INFO

LINK TO HAP NAME: _____

USE CURRENT DATA FILE

USE FILE NAME: _____

GET FILE NAME FROM LOCAL FIELD

FIELD NAME: _____ PREFIX: _____

DESTINATION INDEX ONE: _____

LINE # _____ X-LOC: _____ Y-LOC: _____ V-LOC: _____

CANCEL OK

LINK ON : This designates to HyperLINK when your link is activated. You can design your link so that it is either activated by a Single Mouse Click, a Double Mouse Click, or is started up automatically everytime you run your HAP.

HYPERLINK

LINK TO HAP NAME : Here you would enter the name of the HAP you wish to call up in the link. If this particular HAP is not already loaded in to your main HyperLINK Menu, it prompts you whether or not you wish to load it and any associated modules that may or may not also need loading in. For instance, if you are linking to the Graphics HAP (to display an image), and it is not loaded in, HyperLINK will ask you if you wish to load this HAP. Likewise if the module HL_IMG which controls the Graphics HAP is not loaded in, HyperLINK will ask if you wish to load this in as well. Remember, a controlling module HAS to be loaded in to allow you to run a HAP which uses it.

USE CURRENT DATA FILE : In each HAP (Database HAPs anyways) you have a default data file for that HAP to use. This is entered in to your HAP at the APP INFO Screen (Figure 4.26). When you are performing a link to a Database HAP, this option tells HyperLINK to use the data file which is currently the default of the HAP you are linking to.

USE FILE NAME : This selections tells HyperLINK not to use the regular default data file of the HAP it is linking to, but rather another data file. If you select this option, then you MUST tell HyperLINK which alternative data file to use. If you do not give HyperLINK the name of the alternate file, it reverts back to using the default.

GET FILE NAME FROM LOCAL FIELD : What used to be called an "Indirect Link" in earlier versions of HyperLINK has now been simplified. You use this selection if the button that initiates your link, does not contain enough info on completing the link. For instance, in the

FOUR

Baseball HAP included with HyperLINK there is a GRAPHIC BUTTON of a Camera. When running this HAP, when you click on the Camera Icon, HyperLINK calls up a link to the Graphic HAP, opens a window, and displays ____? Where does HyperLINK determine what to display in this window? Well, in this case the link was set to get the file name to complete the link from one of the local fields, CARD. Therefore HyperLINK knew to load the Graphics Hap, open a window for it, and display an IMG file that matched the card number shown in the current record. (ex. A record in your Baseball HAP for card number 144, when you call the link to the Graphics HAP, it knew to look for a file 144.IMG since all Graphics files have to have an IMG extension.)

FIELD NAME : When you are using the "Get File Name From Local Field", the Field Name MUST be filled out. Otherwise HyperLINK will not understand what you are trying to do, and no link will be performed.

PREFACE : This is to provide users with more than one data file associated to any local field. The best way to explain this is through an example. Once again let us use the Baseball HAP with the Camera Icon linking to an IMG file (a scanned image of the card). Well, that IMG was the card's face. Suppose we wanted another IMG file of the card's back. Both links are getting their linking info from the local field "CARD", but if you insert a prefix into the link to the card's back, it will look for a slightly different destination file to load in to the Graphics HAP. For instance, if your Prefix for the link to the card's back is B, then rather than looking for an IMG file named 144.IMG, this link would look

HYPERLINK

for B144.IMG.

DESTINATION LINK ON : If you are linking to any database HAP, you do not necessarily have to use that database's default INDEX. You can configure your link to use any index that has already been created in your destination HAP. Again, if you want to use an alternative to the default, make sure that you enter the name of the index to use, otherwise HyperLINK will once again refer to the default.

LINE # : If you are creating a link to the Editor HAP, this entry allows you to select the actual line number of your destination text file that will appear at the top of your Editor's window. If you do not enter this when linking to the Editor, you will still load in the appropriate text file, but you will find the Editor displaying it from the beginning of the file.

X-LOC : This is if you wish to provide the X screen coordinate whereby the upper left corner of your destination HAP's window will be located.

Y-LOC : This is the Y screen coordinate of the upper left corner of your destination HAP's window. You will have to experiment with these coordinates to get your windows appearing exactly where you want them to.

CANCEL : This causes HyperLINK to ignore any changes you've made to this linking screen.

OK : This tells HyperLINK to accept all the changes you've made to the current linking screen.

FOUR

LINK TO A HAP AND DO A SEARCH - (Figure 4.22) Most of the items in this linking screen are the same as in a link that does not perform a search. The difference being that you have to complete the searching parameters as well. We will just concentrate on these LOOKUP/SEARCH DATA SOURCE fields, as they represent the differences between this screen and Figure 4.21.

Under the LOOKUP SEARCH entry...

GET FROM LOCAL FIELD : If the record you are looking for in your destination HAP contains a matching field with any field contained within your source HAP, enter the name of the field in your source HAP from which to get the data to conduct your search. If

(Figure 4.22)

that explanation confused you, you're not alone... Here's an example of what was meant. Consider the Message and Address HAPs that are included with HyperLINK. When running the Message HAP, if you Double Click on the "From" field you are invoking a link to the Address HAP. For search information, HyperLINK has been told to get the search parameters from the data contained in the Message HAP's "From" field and see if it can find in the Address HAP's "Lastname" field a matching record. Okay, this means that "FROM" was the local field which provided the data to complete the search.

DIRECT PARAMETER : This allows you to

HYPERLINK

perform a search on any fixed string (Parameter) you may wish. Unlike data fields which may contain a different text string in every record, and thus cause a different link for every record; by entering a "Direct Parameter" your link would be exactly the same every time, regardless of the contents of your source HAP. As an example, if your Direct Parameter is "Mouse", whenever you invoke a link, HyperLINK will search for "Mouse" under the current index in your Destination HAP.

EXPRESSION : In the Link & Search screen, the Expression entry allows you to perform your linking search based upon multiple indexes. That is, if you have created an index in your Destination HAP based on multiple fields, for instance Name + City, then you can enter this multiple index as an Expression.

LINK TO A HAP AND PASS DATA - (Figure 4.23) Once again most of the linking screen appears the same. The difference being that this type of link is geared to calling up a blank record in your destination HAP and filling in

(Figure 4.23)

The screenshot shows a window titled "LINK TO A HAP AND PASS DATA" with several sections:

- SELECT TYPE OF LINKS:** Four radio buttons: "Link to a HyperLINK App (HAP)", "Link to a HAP and do a SEARCH", "Link to a HAP and pass data" (selected), and "COMMAND Link". Below are buttons for "SINGLE CLICK", "DOUBLE CLICK", and "STARTUP".
- DESTINATION HAP INFO:** Fields for "LINK TO HAP NAME:", "USE CURRENT DATA FILE" (checked), "USE FILE NAME:", "SET FILE NAME FROM LOCAL FIELD" (checked), "FIELD NAME:", "PREFIX:", "DESTINATION INDEX ON:", "LINE #", "R-LOC", "V-LOC".
- LOOKUP SEARCH OR PASSED DATA SOURCE:** Fields for "GET FROM LOCAL FIELD:", "DESTINATION FIELD FOR PASSED DATA:", "DIRECT PARAMETER:", and "EXPRESSION:". Below are "CANCEL" and "OK" buttons.

information into any one specified field in this blank record. Let's concentrate on the differences involved in this type of link...

Under the *LOOKUP SEARCH OR PASSED DATA SOURCE* heading...

FOUR

GET FROM LOCAL FIELD : This provides the field from which HyperLINK would take the information to place in the selected data field in the destination HAP.

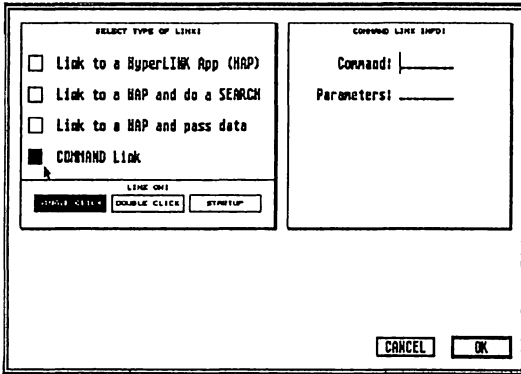
DESTINATION FIELD FOR PASSED DATA : This is the data field in the Destination HAP which will accept the data which is passed from the source HAP.

DIRECT PARAMETER : Allows you to hardwire the entry into the Destination HAP which will be passed. For instance, if you were linking to the Address HAP, a Direct Parameter of "SMITH" would call up a blank record in the Address HAP and insert SMITH into whichever field the Address HAP is currently being indexed on.

EXPRESSION : The Expression field in the Pass Data Link screen can either be a string expression or a numeric one. Whichever it is, is calculated and the the result of that calculation constitutes the data passed to the receiving HAP.

COMMAND LINK - (Figure 4.24) Ah! Rather a drastic change in the appearance of your linking screen occurs when you select this type of link. A Command Link is exactly what it implies. Rather than linking to another HyperLINK Application, this allows you to create buttons that perform actions like going to the next record, or any of the other commands currently supported by HyperLINK. In Command Link Info we must complete...

HYPERLINK



(Figure 4.24)

COMMAND : This is the actual HyperLINK command to perform when this link is invoked. (For a listing of all the HyperLINK Commands available, see Appendix B.)

PARAMETERS : A parameter associated with a command link is extra information which

the command will use when performing its function. If the Command entered is "Print", for instance, HyperLINK's first step would be to use an item selector to allow you to choose which Report you want printed. If you enter the name of the Report you want printed in this Parameter field, then HyperLINK need not resort to calling up your item selector, and you've saved yourself a few steps.

THE BUILDER MENU : (Called up by clicking on the Right Hand Mouse Button any time you are in the Application Builder)

(Figure 4.25)



This menu (Figure 4.25) provides you the file functions necessary for use with the Builder (ie. "Load" / "Save") as well as some other key options that are specific to the Builder.

LOAD : Used to load in an existing application in to the builder for modification. This item calls up your file selector and is keyed to files with a .HAP extension. Note the alternative keystroke <ALT> + <L> that can be used any time the Builder screen is active.

FOUR

NEW : If you have begun work on a form and you are not happy with the results, rather than deleting every item you have already in place, this menu option erases every object currently in the builder. It provides a quick method of starting from scratch.

APP INFO : This menu item calls up a very important screen (See Figure 4.26) that **MUST BE FILLED OUT PROPERLY FOR EVERY HAP!** If you do not fill in the information for your HAP before you save it for the first time, HyperLINK will guess at what this info should be. While HyperLINK is intelligent enough to understand what you are trying to do, it might be misunderstanding your intentions. Therefore it is always a good idea to manually fill out this information for every HAP you create.

SAVE : Saves the application you are currently constructing. If you have not given your form a name already, your item selector will appear and you will be expected to enter a name for your new HAP.

SAVE AS : This menu item also saves the application you are currently constructing but always calls up the item selector, thus giving you the chance first to give your HAP an alternate name or to save it to a different disk path.

THE APPLICATION INFO SCREEN : This screen logically should follow the discussion of the Builder Menu as this is the source for it. You can either select the App Info item from the Builder menu or use the keyboard to call up this screen <ALT> + <I>.

HYPERLINK

Application Information

MUST BE FILLED OUT

HAP Name : _____
Uses MODULE : _____

USER DEFINED INFO
AUTHOR ID1 : _____ NUMERICAL ID1 : 0____
HWP REVISION1 : 0,____

DATABASE INFORMATION
 STANDARD HYPERLINK DATABASE
 CUSTOM OR NO DATABASE
DATABASE NAME : _____ DEF INDEX : _____

VISUAL INFORMATION / OPTIONS
 LOG IN ICON WHEN HWP LOADED ?
 AUTOMATICALLY LOAD FIRST RECORD ?
FIRST BODY FIELD : _____

HAP ICON
LOAD

WINDOW FEATURES
NAME SHOWN CLOSE BOX
HOWER BAR FULL-SCREEN

SPECIAL / CUSTOM INFORMATION
SECONDARY FILENAME : _____ FILE SELECT

CANCEL OK

(Figure 4.26)

HAP NAME : This is the name your HyperLINK Application will use both as its file name when it is saved, and the name it displays on the HyperLINK Desktop Menu when your HAP is loaded in.

USES MODULE : This is to tell HyperLINK which

module your HAP will be using. (ie. the Database Module, HL_EDIT, or HL_IMG, etc.)

Below this box, there is the Database Information of your HAP. This area only need be completed if your HAP uses the Database Module...

STANDARD HYPERLINK DATABASE : This is the only type of file format HyperLINK currently supports in its Database Module. Whenever the Database Module is being used by a HyperLINK Application, this defaults to being selected.

CUSTOM OR NO DATABASE : If your HAP does not use the Database Module, then this option becomes the default setting. (In the current version of HyperLINK you do NOT need to alter either of these two buttons from whichever setting they're defaulting to.)

DATABASE NAME : This information only needs be filled out if your HAP is using the Database Module. If it is, then this is the name of the data file first created, and then used by your HAP. The extension on database files should be

FOUR

DBF. (ie. If you have a database file for your Message HAP, then a likely name for it would be MESSAGE.DBF)

DEF INDEX : This is short for Default Index. It allows you to choose the default criteria on how your Database HAP is sorted. You simply enter in this field the name of the index that should be used automatically every time this HAP is run. (NOTE: Indexes are not created in this screen, they must be created in the Application Defaults screen, Figure 4.27)

HAP ICON / LOAD : In this little box on your App Info screen you are shown the icon which your HAP currently would use when it is loaded into the HyperLINK Main Menu. If you do not like the current selection, click on LOAD (up will come your file selector) and choose an alternative icon. NOTE: 48x48 icons provide a better look in the Main Menu than 32x32 icons.

Dealing with the overall appearance of your HAP's window, are the settings within "Window Features"... The following four items are simply preference selections on the appearance of your application's open window.

NAME SHOWN : When this item is selected, the name of your HAP appears across the top of its open window. If this option is deselected, then the name of your HAP is not displayed anywhere in the window.

CLOSE BOX : When selected, your window contains a button in the upper left hand corner of its frame which when clicked on during the running of your HAP, will close that HAP's window. When this option is deselected, then no

HYPERLINK

such button appears in your window's frame. (There is a keyboard equivalent to the use of this button. If you wish to close your ACTIVE HAP, hit the function key <F10>.)

MOVER BAR : When selected, across the top of your open HAP window would appear a shaded frame (which also may or may not contain the name of your HAP). If you wish to move the screen location of your ACTIVE HAP, click and hold down the left mouse button over this bar, and drag your HAP to its new location. If you have disabled the Mover Bar, then there is no way to change the screen location of your HAP.

FULL / MIN : When enabled, this option places a button in the upper right corner of the frame around your HAP window. When disabled, the button does not appear. This button can be used for two functions :

First, by clicking on this button in an open HAP, you "Minimize" your window. (This keeps your HAP at whichever record it is on, but places it in effect, in a holding pattern along the right side of your HyperLINK Desktop. To reactivate your HAP at its current record, just click on this Minimized icon.) There is also a keyboard equivalent for minimizing HAPs ; this is the function key <F9>.

Secondly, if you <SHIFT> click (ie. hold down the <Shift> key while you click) on this button in your open HAP, you would make your window Full Screen. (Note: Full Screen is only available for some HAPs : the Builder, the Report Builder, and the Text Editor. Other HAPs are only available to be viewed in the size they were created originally in the Builder.) There is no

FOUR

keyboard equivalent to performing this function.

Next comes the User Defined Info. While this information is not essential, it can help identify the creator of a HAP, and also provide a location for labeling which version number of a HAP you are currently using.

AUTHOR ID : In this field you can enter whatever you wish to help identify you as the creator of this HAP.

NUMERICAL ID : This is an optional field for giving your HAPs additional identification information. While this is not commonly used, it is there if you need it.

HAP REVISION : This provides you a field to identify the current version of your HAP. As changes are made, and functions are added to your HAP, you would change your revision number to identify such.

The next area of the App Info screen to be completed is the Visual Information / Options.

LOG IN ICON WHEN HAP LOADED : If this option is enabled then your HAP's icon will be displayed in the HyperLINK Main Menu when it is loaded in. If deselected, then no icon will appear in the Main Menu. As an example of when you might use this option : If you have created an application that is only available as a supplement to another HAP, ie. you do not want it to be used separately, but merely as a HAP only available through a link from another HAP, then there is no need for this HAP to be loaded, or "Logged In" to the Desktop.

HYPERLINK

AUTOMATICALLY LOAD FIRST RECORD

: This option is only applicable to a Database HAP. If it is enabled, then whenever you run an application which uses the Database Module, the first record (based on the default index) will load in. If this option is deselected, then whenever you run a database HAP it defaults to loading up a blank record.

FIRST EDIT FIELD : This allows you to select which field in a database HAP defaults to being the location of the editing cursor whenever a record is loaded within that HAP. If no field is specified here, then the default is understood to be the actual first database field in your HyperLINK Application.

The only remaining data to complete regarding your HAP comes under the heading of Special / Custom Information. This is strictly optional info.

SECONDARY FILENAME : Currently there is no Module support for this option, but essentially it allows for future expansion whereby a HyperLINK Application may use more than one datafile. As an example, consider a Movie Module : Within one file would be the actual picture data, and within a "Secondary" file would be the sound data. When the HAP actually runs, it merges the pictures with the sound.

FILE SELECT : If this button is selected, then whenever you run this specific HAP, the item selector will first appear allowing you to tell HyperLINK which file to use. In other words, this makes HyperLINK need you to choose the datafile it will use for this session with the HAP every single time you call up this application

FOUR

from the Main Menu.

CANCEL / OK : As with all HyperLINK screens the Cancel button gives you the opportunity to abort any changes you may have made, while the OK button confirms and saves whatever changes you've made.

THE APPLICATION DEFAULTS SCREEN : There is also one further screen that is relevant to the BUILDER, the APPLICATION DEFAULTS screen (Figure 4.27). This screen is not called up from within the BUILDER though. Rather it is called up from the Main HyperLINK Desktop. Instead of just clicking on an application in the Main Menu, by holding down the <ALT> key when you click on any application, you bring up the Application Defaults for that HAP. Some things on this screen are duplicated from the APP INFO screen within the BUILDER, such as which data file to use, which default index to use, whether or not to load the first record when your database is called up, but also on this screen you can create new indexes for your database, or pack your database to eliminate deleted records, etc.

If you change any of the information items which have been loaded in from the Builder's App Info, then these are only valid for your current HyperLINK session, and the next time this HAP loads, the default settings from the App Info screen will return.

As you will notice, some of the information on this screen has a *Disabled* appearance. This indicates info that can NOT be modified. (See the Application Info screen for further reference - Figure 4.26) Therefore we will concentrate only on the editable fields in this screen.

HYPERLINK

APPLICATION DEFAULTS

GENERAL INFORMATION
HAP Name : BUILDER
Author ID : JRG Ver 200
Module Name : BUILDER
DB Type : APPDEF

MODIFIABLE PARAMETERS
Database File : _____
Default Index : _____ LIST
Secondary Name : _____
First Edit : _____

INDEX FILE CONTROLS
MAKE NEW INDEX : Create INDEX
FIX/REDO INDEX : RE-INDEX
REMOVE DELETED : PACK
REMOVE INDEX : Delete INDEX

LOCATION : X Y (IN CHARS)
 AUTO LOAD 1ST REC FILE SELECT

CHECK FIELDS ETC : CHECK Date

CANCEL OK

(Figure 4.27)

The first area we will look at are the Index File Controls... These items are only applicable if your HAP uses the Database Module.

CREATE INDEX : Select this option if you wish to create an index based on any field, or combination of fields, within a Database HAP

that has not already been created. When you create an index, you must also name it. It is important to remember the characteristics of your index, so name it appropriately. (Creating an index also creates a file on your disk "NAME.NDX" which contains the data on this index.)

RE-INDEX : If you suspect that somehow your database file has become corrupted or your index has lost its integrity, this option will rebuild your index from scratch.

PACK : For those of you familiar with Dbase commands, you will recognize the significance of this command. Basically, when you delete a record from your database file, it is just flagged as deleted and is actually still contained in your file but is no longer displayed in reports etc. When you pack your database, you are physically removing the contents of these "Deleted" records from your file. Once you perform a PACK you can no longer recover deleted records.

DELETE INDEX : This option will delete from your list of available indexes, whichever index

FOUR

you choose. It will also delete the associated NDX file from your disk.

This brings us next to your Modifiable Parameters. Remember, any selections you make in this area are only valid in the current session of HyperLINK, and the next time you load this HAP these choices will revert to the defaults selected in the App Info screen of the Builder.

DATABASE FILE : If you wish your HAP to use a file other than its default, enter the name of this file here.

DEFAULT INDEX : This allows you to select a different index for this session of your HAP.

LIST : By selecting this button, a list of all the indexes that have been created for this HAP, is displayed.

SECONDARY NAME : Not currently used by HyperLINK, but when it is supported, it will allow for one HAP to use more than one file. (For more info on this item, see the APP INFO Screen - Figure 4.26)

FIRST EDIT : Allows you to select for this session, the data field where your edit cursor will default to as each record is called up.

LOCATION (X/Y) : Provides the screen location where the window for this HAP will appear. Location is based on the upper left corner of your application window, and is given in characters.

AUTO LOAD 1ST RECORD : If you wish HyperLINK to load the first record for your HAP

HYPERLINK

(based on the current index), select this option. Otherwise when you run your HAP, a blank record will first appear.

FILE SELECT : If this is selected then HyperLINK will prompt you with your file selector every time you run this HAP in the current session, providing you a choice of which data file to use.

If you have added a field to your database, make sure you "Check Data" before you enter any data into it. Otherwise, the info in this new field will not be recorded into your database file.

CHECK DATA : This is one of the new features to HyperLINK 2.0. When selected it rebuilds your database file, checking to see if you have added or modified any fields within your HAP since the time when the database file was first created.

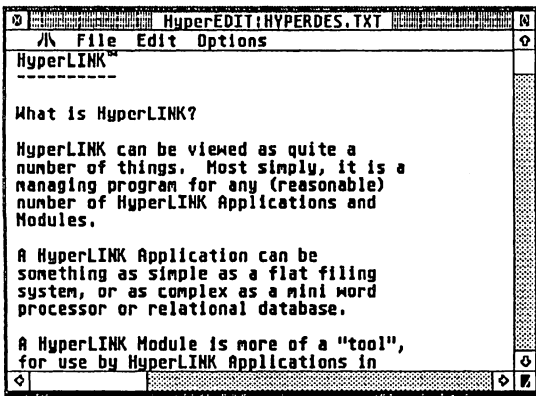
CANCEL / OK : Once again, the Cancel button gives you the option of ignoring any changes you may have made on this screen while the OK button confirms your selections.

FOUR

PART C

THE TEXT EDITOR :

The concept of a Text Editor should be familiar to most. Essentially it is a scaled down Word Processor. HyperLINK's editor (HyperEdit - Figure 4.28) is very straight forward in its operation and makes an excellent tool in providing additional information capabilities to your HAPs. As well, it can be used as a stand-alone feature for allowing simple note taking.



There is very little that can be considered confusing about HyperEdit. Its actual window follows the same rules as any other on the HyperLINK Desktop. That leaves just its various menu items to be explained...

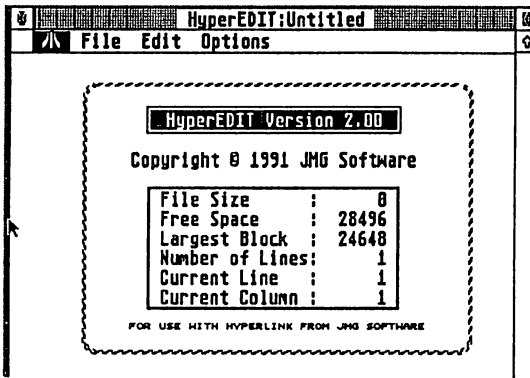
(Figure 4.28)

TEXT EDITOR MENUS :

Unlike typical ST GEM Menus, if you wish to select one of the window menus of the editor, you have to click the left mouse button on the desired menu to activate it.

THE FUJI MENU : Only one item currently resides under this menu selection, "HyperEdit Info". This calls up a status screen on the text file currently loaded in (Figure 4.29).

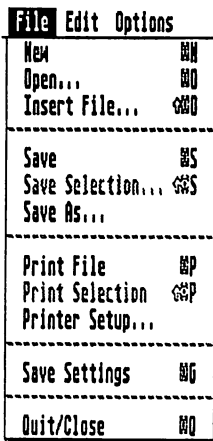
HYPERLINK



(Figure 4.29)

As you can see, this status screen lists the File Size (in bytes) of the text file currently loaded in, the Free Space in memory still available, the Largest Block, the Number of Lines in your text file, and the current location of your cursor based on Line Number and Column.

EDITOR FILE MENU : (Figure 4.30) This is the location for any file operations within the editor, such as Load, Save, etc.



(Figure 4.30)

New File - Clears whatever text file you are editing out of memory for you to work on a new file. If you haven't saved the file you were working on, you are given the chance to save it before the current file is abandoned. An alternative keystroke that can be used in place of this menu selection is $\langle \text{ALT} \rangle + \langle \text{N} \rangle$.

Open File - Similar to above option, but calls up your file selector asking which text file you wish to load in. Also gives you the option at this time of saving your current work before it is lost. The alternative keystroke is $\langle \text{ALT} \rangle + \langle \text{O} \rangle$.

Insert File - Calls up the file selector asking for which text file you wish to Append to the current document being edited. Append occurs at current location of the cursor. Alternative keystroke is $\langle \text{SHIFT} \rangle + \langle \text{ALT} \rangle + \langle \text{O} \rangle$.

FOUR

Save - Does not abandon current document, just gives you the option of saving the text file currently being edited. (Remember, it is always a good idea to save your files often.) Alternative keystroke is <ALT> + <S>.

Save Selection - If you have a marked block within your text file, this menu item allows you to save out only the selected text to a separate file. Alternative keystroke is <SHIFT> + <ALT> + <S>.

Save As - Saves the file, first calling up the file selector to give you the option of saving a file under a different name or in a different path. (If you continue using the same name every time you save a file, you just write over the last version of that file. Therefore it is a good idea to change the name every so often when you are saving to help insure the safety of your data.) There is no alternative keystroke for this menu selection.

Print File - As it says, sends the current text file to the printer. Alternative keystroke is <ALT> + <P>.

Print Selection - If you have a marked block within your text file, this menu item allows you to print out only the selected text. Alternative keystroke is <SHIFT> + <ALT> + <P>.

Printer Setup - Used for setting up the page format for your printouts. Lines per page, Margin, this is where you set up things like this. There is no alternative keystroke for this menu selection.

Save Settings - This is used for saving the default configuration of your editor window. Window

HYPERLINK

sizing, window positioning, can all be saved here. While this feature is not available in the current version of the editor, it will be available soon. The alternative keystroke is <ALT> + <G>.

Quit/Close - This selection closes the active editor window giving you the option to save first. Alternative keystroke is <ALT> + <Q>.

EDITOR EDIT MENU : (Figure 4.31) Much the same as any editing functions you are familiar with in other software, to select a block of text for editing simply hold down the left mouse button and drag the cursor over the text you wish to highlight.

NOTE : In this section, a "CLIPBOARD" is referred to many times. This is used by the text editor as an external file for temporary storage. Only one item can be in this buffer at one time. If you send a second item here, then the first one is written over and lost. The Clipboard being saved as a temporary file and not just allocated to memory allows it to be kept even if you exit the HyperLINK program.

Edit Options	
Cut	⌘X
Copy	⌘C
Paste	⌘V
Delete	DEL

Select All	⌘A

Reformat Paragraph	
Reformat to End	

Find...	⌘F
Find Next	⌘A
Change...	⌘H

(Figure 4.31)

Cut - Moves any selected text from where it is in your file to the Clipboard. This is used for moving text blocks about in a document. Alternative keystroke is <ALT> + <X>.

Copy - Similar to "Cut" but the copy function does not delete the text out of the original document from which it is moving to the memory buffer. This is used when you might be using the same block of text somewhere else, either in this document or another. Alternative keystroke is <ALT> + <C>.

FOUR

Paste - Inserts whatever text is currently in the Clipboard, starting at the present location of your cursor. If there is no text currently in the clipboard then this menu option is disabled. Alternative keystroke is <ALT> + <V>.

Delete - Erases the currently selected (highlighted) block of text from the document. Once again this option is available only if you have selected a text block. Alternative keystroke is appropriately the <DELETE> key.

Select All - Marks the entire text of the current document. Alternative keystroke is <ALT> + <A>.

Reformat Paragraph - If you've inserted text to a paragraph, this option reformats the text to match the current configuration of the editor. Only reformats the current paragraph where your cursor is located. Alternative keystroke is <ALT> + <R>.

Reformat to End - Similar to reformatting a paragraph, this option is not limited to just one paragraph, but reformats the document from the point of the cursor to the end of the file. Alternative keystroke is <SHIFT> + <ALT> + <R>.

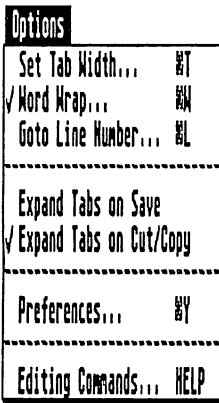
Find - Searches for an occurrence of a specific text pattern. Used primarily for finding specific words or phrases within a document. Alternative keystroke is <ALT> + <F>.

Find Next - Does not change the criteria of the last search, merely goes on to look for the next occurrence. Alternative keystroke is <SHIFT> + <ALT> + <A>.

HYPERLINK

Change - This is a "Search and Replace" option, whereby the user can tell HyperEdit to change all occurrences of one specific text content to something else. Alternative keystroke is <ALT> + <H>.

EDITOR OPTIONS MENU : (Figure 4.32) The options contained under this menu deal with the overall performance of the Editor and are not specific to any one file.



(Figure 4.32)

Set Tab Width - Allows the user to enter the interval at which tabs occur on a line in your text. For instance, a tab width of 4 allows for four spaces between tab marks, therefore there would be tab marks at column 5, 10, 15 etc. Also under this selection you can configure the HyperLINK Editor to display the tab marks in any open window. Alternative keystroke is <ALT> + <T>.

Word Wrap - This is used primarily to set the line length of your document. Being that the HyperLINK Editor window can be sized variably, you would usually set the "Wrap Column" width to what would fit in your open window. When your "Word Wrap" is enabled a check mark will appear to the left of this menu selection. Alternative keystroke is <ALT> + <W>.

Goto Line Number - This moves your cursor to whichever line number in your text file you specify. Alternative keystroke is <ALT> + <L>.

Expand Tabs on Save - If this option is enabled, (a check mark beside this menu item will indicate if it is enabled) then when you save a file, every tab mark is expanded and replaced by spaces. There is no alternative keystroke for this menu function.

FOUR

Expand Tabs on Cut/Copy - The same as for "on Save" as it relates to things being sent to the Clipboard. Once again there is no keyboard alternative for this menu item.

Preferences - (Figure 4.33) Allows you to modify the point size of the screen font to be used with HyperLINK. Also allows the user to configure the amount of computer memory allocated for maximum size of Hyper Editor files.

For instance, if the Memory Buffer Size is set to 32K then the maximum size of the file loaded in to the editor is 32K. A third setting under this option is the location (or disk drive path) of the editor's Clipboard. Alternative keystroke is <ALT> + <Y>.

Memory Buffer Size (32-999):	32 K
Drive to use for Clipboard :	C
Point size for display font:	18
Window/ACC Title Prefix :	HyperEDIT:
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>

(Figure 4.33)

Editing Commands - Calls up a HELP screen (Figure 4.34) for commands within the

(Figure 4.34)

SHIFT-↑	Go up one screen
SHIFT-↓	Go down one screen
SHIFT-→	Go to end of line
SHIFT-←	Go to beginning of line
CTRL-→	Move right by 1 word
CTRL-←	Move left by 1 word
Shift-Delete	Delete the current line
Shift-Backspace	Delete to end of line
Home	Go to beginning of text
Clr	Go to end of text
CTRL-Return or CTRL-J starts a new line Indented the same as the one above it.	
Drag the mouse to select text for Cut/Copy/Paste/Delete, or use a SHIFT-Click to extend the selection.	
<input type="button" value="Hit any key to resume editing"/>	

HyperLINK Editor. Alternative keystroke is <HELP>. Most of the Help given on this screen refers to cursor movement within your text file. NOTE : <HELP> will always look to see if the ACTIVE HAP has a Help Screen.

HYPERLINK

PART D

CREATING TEXT LINKS

The Text Linking screen (Figure 4.35) is called up when you are creating a link from an existing text file. You can form a link to a Database HAP, another Text File, a Graphic File, or any other module that may be available in future for HyperLINK. To original call up this screen, you need merely double click (or <ALT> click) on any word in the file currently loaded in the text editor. For more details on "Linking" in general, see both the Reference Section and the Appendices.

There are a couple of background notes on the Text Linking that should be made clear before we proceed any further :

1) The TextLink Builder is a HAP, albeit a very specialized one. Therefore, if you want to create a TextLink, this HAP **MUST** be loaded in. Otherwise no linking screen will be called up.

(Figure 4.35)

2) Whenever there are links from a text file, HyperLINK is in the background creating a Database File for your Text File. In this DBF it links the occurrence of a word, with the information to perform the link.

LINK TO HAP NAME:
This is the name of the HAP you want your text

DEFAULT

TEXT LINK BUILDER

LINK TO HAP NAME : _____ LOOKUP AFTER LINK

FILE NAME TO USE : _____ (BLANK FOR CURRENT FILE)

INDEX TO USE : _____ (BLANK FOR DEFAULT)

TO LINE : 0 (WHERE APPLICABLE)

LOOKUP STRING : _____

FOUR

to link to.

LOOKUP AFTER LINK : This button has the same effect as a "Link to a HAP and do a Search". In other words, if this option is enabled, when you link to a HAP, a search is performed for a record matching the text string the link was called from. Remember, the field in your destination HAP which you want searched, has to be indexed.

FILE NAME TO USE : This gives you an option of using (in your Destination HAP) a data file other than the default. If you don't want the default file used, you **MUST** enter the alternative file name here.

INDEX TO USE : If you wish to perform a search on an index other than the destination HAP's default, enter the alternative index name here.

TO LINE : If you are linking to a Text File, you have the option of linking to a specific line in the destination file.

LOOKUP STRING : This gives you the alternative of hard wiring a search, based on something other than the source text string.

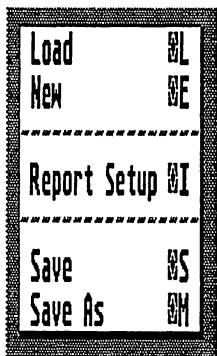
PART E

THE REPORT BUILDER :

Like the Application Builder, the Report Generator is not just comprised of one component. It is very straightforward to create reports that are just spewing pre-canned data to a printer, but it gets a little more complicated when you also wish for sorting, searching, and calculations done. As such, the more involved reports call up an additional screen to handle these parts.

THE REPORT BUILDER MENU :

(Similar to the Builder, the menu for the Report Generator is called up by clicking with the Right Mouse Button.)



(Figure 4.36)

LOAD - Used to load in an Report in to the Report Builder for modification. The alternative keystroke for this menu item is <ALT> + <L>.

NEW - The selection made to start creating a new Report in the builder. This loads in a blank form from which you can start creating your own Report Form. The alternate keystroke is <ALT> + <E>.

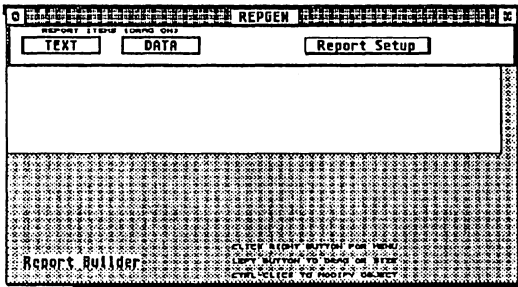
REPORT SETUP - Calls up another screen for doing the general setup as well as the intricate details of your Report. This is gone in to in much further detail later in this section. The alternate keystroke is <ALT> + <I>.

SAVE - Saves the Report you are currently creating. Alternative keystroke is <ALT> + <S>.

FOUR

SAVE AS - Saves the Report you are currently constructing but gives you the chance first to give it an alternate name or an alternate path. The alternate keystroke is <ALT> + <M>.

MAIN REPORT BUILDING SCREEN : (Figure 4.37) The HyperLINK Report Generator is the place to design all of the Reports that you want your HyperLINK Applications able to access and print.



(Figure 4.37)

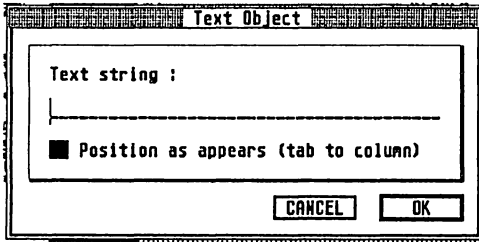
It works in much the same manner as the Application Builder itself and when first called up is merely a blank form. Simply by dragging **TEXT**, and **DATA** to the form anywhere you want, you can generate an application. Not only though is the Report Builder used to create the look of a Report but more importantly this builder is where all the Searching and Sorting and Calculations for your Report are configured.

Make sure that you also read the Appendices concerning the Report Builder to learn all the flexibility of this important part of HyperLINK.

REPORT TEXT PARTS - As in the Application Builder, whenever you drag a specific object to your form you are met with a distinct dialogue box in which to enter the details of your object. Unlike the Application Builder though, dragging **TEXT** to a Report involves much less configuring. **TEXT** on a Report is simply a non-changing constant wherein the same text will appear in the same place every time you print

HYPERLINK

your report. It is useful for headings, and any descriptive explanation you wish to have included with your customized report. When you move a Text entry onto your Report you are met with the Report Text Object screen (Figure 4.38)...



(Figure 4.38)

TEXT STRING : Here you need only enter the text exactly as you wish it to appear on the Report.

POSITION AS APPEARS: This option allows you to control the spacing of text on

your report as it relates to preceding Text or Field Objects. If enabled, your text will fall on the next Tab Mark from where the previous Object ended. If disabled then your text will immediately follow the preceding text or field without tabbing.

REPORT DATA PARTS - The fundamental difference between TEXT and DATA in your report is that DATA is an ever changing item. When you are printing a report, it is the DATA portion that makes it unique. DATA can be many different things, from a field in a database, to a mathematical expression, to any number of other logical functions the computer can perform. To help explain this concept, let's take a look at the Report Data Screen (Figure 4.39)...

FIELD NAME : If you are printing in this space on your report the contents of a field in a database, here you would enter the field name to be printed. You must both select this option and enter the field name.

FOUR

Report Object

Field Name : _____
OR
 Expression String : _____

Maximum Length : _____ Decimal Spaces : _____
(BLANK FOR NONE) (BLANK FOR FLOATING)

Print Format : Justify to :

STRING	DATE	YES/NO	LEFT	RIGHT
INTEGER	FLOAT	EXPONENT	POST'N AS APPEAR	

CANCEL OK

(Figure 4.39)

EXPRESSION STRING :

If you choose this option then your report will perform the command you enter and then print the results of your expression. (For more details on the expressions which can be performed by HyperLINK, see the Appendix on Expressions.)

MAXIMUM LENGTH :

This sets a maximum field length in your report for your Field Object. If you leave this entry blank, then your report will keep adjusting the field length of this object, based on the actual length of the data it contains. In other words, you have a free floating length and it will adjust its size according to the contents ; If the data field contained "Smith" then it would adjust to a length of 5, but in the next record the contents could be "Johnson" which would set the field size to 7 characters.

DECIMAL SPACES : If your field is a numerical value, you can here configure the number of decimal spaces allowed as a maximum in your report. This operates in the same manner as "Maximum Length". That is, if you leave this field blank, the Decimal will be floating (up to your maximum field size).

PRINT FORMAT : These buttons are for you to inform HyperLINK of what type of data is expected in your report so that HyperLINK can handle it appropriately.

HYPERLINK

STRING FORMAT - A string is just any collection of characters. These can be either alphabetical or numerical.

DATE FORMAT - I should think that this type of format is pretty self explanatory... When HyperLINK sees a Date Format selected, then it automatically knows to set up the report field as / / .

YES / NO FORMAT - With this format selected, HyperLINK expects a single character "Y" or "N" to be the contents of your field.

INTEGER FORMAT - With this option, HyperLINK expects a positive or negative numerical value with no decimal places as your field.

FLOAT FORMAT - This is short for Floating Decimal. When this is selected, HyperLINK expects the contents of your data field to be a numerical value with as many decimal places as you specify.

EXPONENT FORMAT - This represents a field containing an exponential numerical value.

JUSTIFY TO : (Left/Right/Post'n as Appears)
This gives you the option of having your report field justified to your preference.

This now lead us back to one of the screens associated with the Report Builder's Menu. If you will recall, one of the selections on the menu was "Report Setup" (Figure 4.40)...

REPORT SETUP - It is on this screen and one of its sub screens that the true power of the

FOUR

Report Builder is achieved (mostly the sub screen)...

Report Setup

Print All Using Default Sort
or
 Use Custom Sort and Search Options

FORMFEEDS
BEFORE BETWEEN AFTER

Page Length : 66
Left Margin : 0 Right Margin : 88
Printer Font : SMALL LARGE

CANCEL OK

(Figure 4.40)

PRINT ALL USING DEFAULT SORT :

This option uses your overall Report Form as it stands, printing all the records from your database file in the report.

USE CUSTOM SORT AND SEARCH OPTIONS :

If you do not wish all the records in a database to print in

your report, or if you would like HyperLINK to perform extra sorting functions, you would select this option. It then calls up the Report Sort and Search Screen (see Figure 4.41).

PAGE LENGTH : This field requires you enter the length of the "Page" you are printing your report to. As a couple of examples : a standard 8.5 x 11 page is 66 lines long, while a standard address label is 6 lines long.

FORMFEED : (Before/Between/After) This allows you to position a page feed (based on your page length) around a record in your report. If none of these options are selected, then you are choosing "No" formfeed.

LEFT /RIGHT MARGINS : As a further Report Formatting feature, you can set the outer limits of your report printing.

PRINTER FONT : This last feature on the

HYPERLINK

Report Setup screen allows for you to improve the appearance of your report through the use of different font sizes.

THE CUSTOMIZED REPORT SORT & SEARCH SCREEN - (Figure 4.41) This must be completed if you wish your Report to do some sorting or record searching which your HAP is not fully set up for. For instance, if you wish to use an index other than your HAP's Default Index, or if you only wish to print a limited range from your database you would click on the "Custom Sort" option in the Report Setup screen and up would pop...

Report Sort and Search

Name of Sort Index : _____

Print All Records

Print a Range : _____

Starting at : _____

To End of File
 or While Expression is True : _____

PRINT ONLY RECORDS THAT MATCH SEARCH EXPRESSION :

AND	OR	_____
AND	OR	_____
AND	OR	_____

CANCEL OK

(Figure 4.41)

NAME OF SORT INDEX : In this field you would enter the name of the datafile's index you wish to use. Leaving this field blank causes the Default Index to be used.

PRINT ALL RECORDS : This is the default setting for this screen primarily because

it becomes inconsequential to even bother accessing this screen if you do not want to perform custom sorts and searches in your report. The real reason for calling up this screen would be to disable this setting and enable...

PRINT A RANGE : When this option is selected all of a sudden this screen becomes one of the most powerful options in all of HyperLINK. Remember, if you do select this option it is important to fill out the remaining info on this

FOUR

screen to take proper advantage of the capabilities herein.

STARTING AT : This is the beginning point of the range of records you want printed. For instance, if you wanted to print all the entries in your Address book where the Lastname is S-Z, then the appropriate entry would be to enable this "Starting at" selection and simply put "S" in the following field. (Remember, your Database file HAS TO BE SORTED by whatever criteria you are using so the "Name of Sort Index" at the top of this screen would have had to be "Lastname".)

TO END OF FILE / OR WHILE EXPRESSION IS TRUE : These are the options on when to end your printing range. Again using the example of printing all records where the lastname is S-Z, the "Starting At" would be "S" and you would enable the "To End of File" selection because "Z" would be the end of the file. If on the other hand you wanted to print a report for names from S-T, then you could select "Or While Expression is True" and enter in the subsequent field "<U" (record is less than U) so that when HyperLINK came to a lastname starting with U it would end the report.

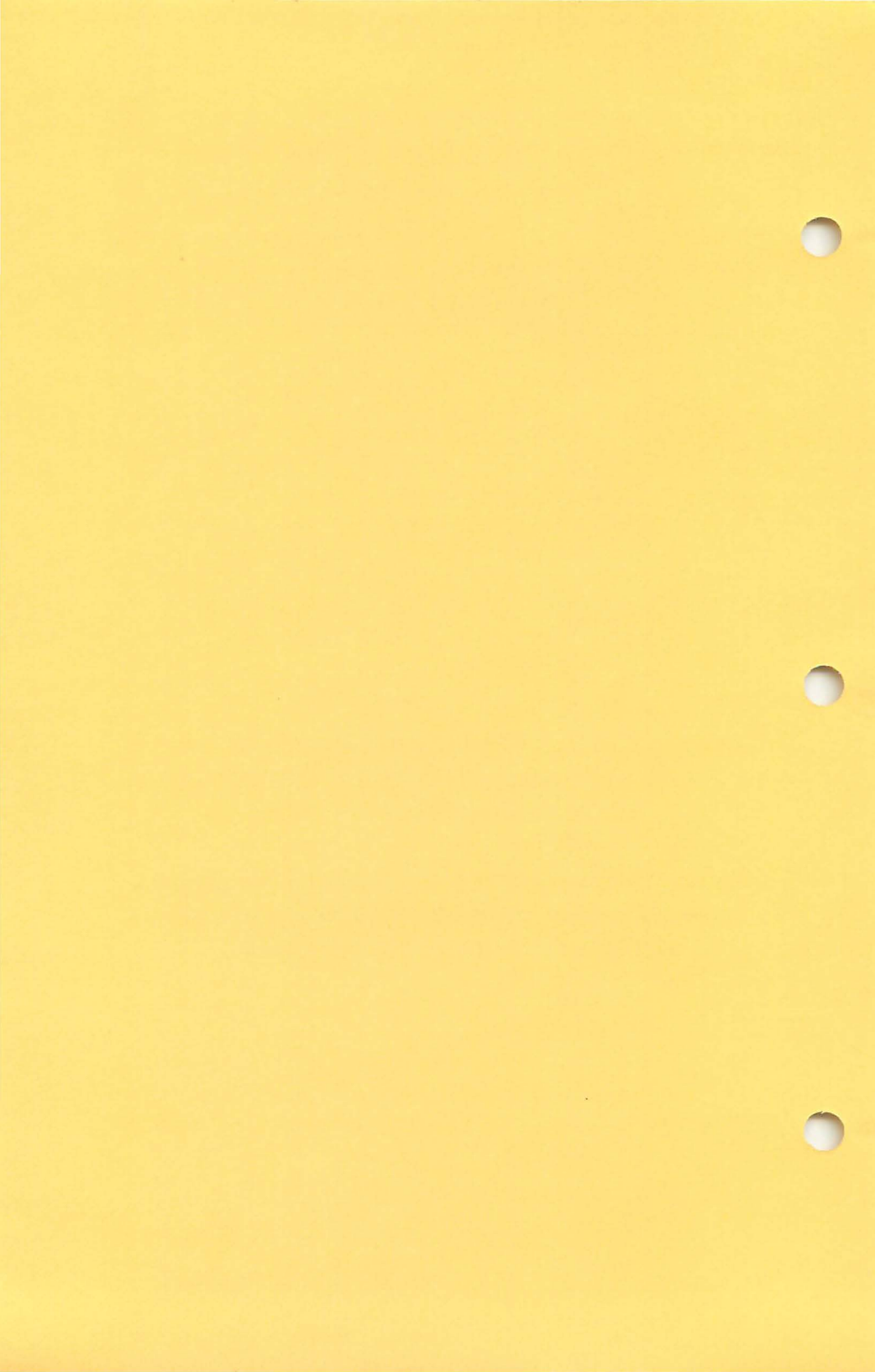
PRINT ONLY RECORDS THAT MATCH SEARCH EXPRESSION : This section gives you the option of entering up to four more expressions for your search criteria and of also qualifying them as And/Or Expressions. (Remember to check out the Appendices for a more detailed listing of the Expressions HyperLINK supports.)



HYPERLINK APPENDICES

This section of the manual provides some of the more technical info on the concepts surrounding HyperLINK.

As well though, we use some analogies here to better explain some of the more difficult areas of HyperLINK.



APPENDIX A

LINKING - A Different Approach

The concept of "Linking" is at the very heart of the HyperLINK program and of its modules. Due to the versatility HyperLINK provides in linking dissimilar items and objects, this entire area of the program can become confusing and difficult to grasp. Hopefully now we can help clear up any such confusion, and shed more light on this very powerful feature of the HyperLINK system.

Also, with the revised Linking Screens now in place in Version 2 and above, hopefully some aspects of creating links in the Builder will be easier to follow.

Before we begin explaining what linking really is about, we should make clear two points: First, each link is controlled by information that you provide and manipulate. In general, the two places where this happens are in the Linking Screens (called up through the various Part Objects in the Builder), and in the TextLink editing form. Second, Linking technology is being continually developed and improved, and as HyperLINK evolves even more in the future, Linking will grow to be more "intelligent", probably to the point where HyperLINK will try to figure out what you want to do and fill in the details automatically. For now, however, the details must be provided by us mere humans.

APPENDIX A

A HyperLINK Analogy

Since the ways in which HyperLINK handles "linking" can be complex to describe from scratch, we decided that an analogy might be in order here. The one we chose is that of a little company operation, with a shipping desk and a receiving desk. Inside each company are a bunch of little managers, each in their own offices. The company represents a HyperLINK Module. Each little manager operating within it are HyperLINK Applications. there may only be one, or there may be many of these guys in each company. The shipping and receiving departments handle shipping (sending) and receiving LINKING PACKETS, which you can think of as little forms that each company sends and receives by courier. The courier is, of course, HyperLINK itself, directing these packets from company to company (module to module).

So, let's take a trip down to the DATABASE company... If we look in, we'll see that there are a few managers in residence - reading signs on the office doors, we can see ADDRESS department, MESSAGE dept, and BASEBALL dept (three of the sample database applications that come with HyperLINK). As we walk into the BASEBALL office, we see the little manager at his desk, doing what the user is asking of him. Since this is the DATABASE company, he has no trouble calling up a data record, showing it on the screen, taking input as it's typed, and saving it back. Why, this little guy can even do searches, delete records, and other fun database things. But then the user clicks on the button on the

HYPERLINK

BASEBALL form that is marked STATS. The manager looks at the button and sees that this isn't a database item - the item info says that this item is LINKED if selected. At this point the manager takes out a memo form and fills in a series of blanks. The information he puts in these blanks is taken right from the item info list of the STATS button, info that was recorded when the BASEBALL application was first created. Our little manager then rushes out and hands this form to the Shipping desk, saying "Send this out right away!"

The shipping department takes the memo; depending upon what the manager wrote, the shippers might have to add some information to the form, but usually they'll just call a courier and have it sent off right away. When the courier comes, they'll take the packet and look up the full address. (All the managers are bit lazy, you see. They'll only write the destination office name on the memo, not the company that the office is in. Luckily the courier has no problem looking up which company the office is in, and of course no two offices can ever have the same name.)

In the case of this memo from the STATS button, the destination office is EDITOR, and the company is HL_EDIT (that's the name of the HyperLINK Module that controls the EDITOR HAP). So the courier rushes out to the HL_EDIT company and hands the memo to their receiving desk. They look at it, rush it up to the EDITOR office and the manager there follows the instructions on the memo. The memo will tell him to open up the EDITOR window, and then

APPENDIX A

load a specific file into it. After he does this, the memo is shredded, having done its job.

Now supposing the user clicks on the camera icon, the BASEBALL manager rolls into action again... same procedure, but the CAMERA button lists Graphics as the destination application (office). Once again, the courier looks up the address and sends the memo off to the correct company (HL_IMG in this case). Once the GRAPHICS manager gets a hold of the memo, he follows the instructions, loading up the file name he was given and showing it in a window.

Before discussing the details of these memo forms (the Links in HyperLINK), we should detour a bit and look at the types of companies involved... Some, like DATABASE, do all sorts of things. They have a shipping department and a receiving department (sending and receiving links), so they are called ACTIVE companies (Modules). Others, like HL_IMG, only have a receiving department - they never actually send anything out, they only receive memos from others. These are PASSIVE companies (Modules). Another difference between some companies is the offices (Applications) themselves - DATABASE has many, such as the ADDRESS database, the MESSAGE database, the BASEBALL database, and you can add as many more offices as you wish, very easily by loading in or creating new database HAPs. These are also then referred to as GENERIC companies. Others, like HL_EDIT and HL_IMG, are a bit different. They only have one office, never any more! Unlike a database, which can be modified

HYPERLINK

and built in thousands of different ways, or an Animation player, which can have a different animation film in each office, companies like HL_IMG can only do one specific thing. HL_IMG gets a picture file from disk and shows it in a window. HL_SND get a sound file from disk and plays it through the speaker. They each do only one specific thing, and so they only need one office and one manager. These are referred to as SPECIFIC applications.

Now back to our memo system. At present, only three companies

actually send out memos ("Active Modules"), these being DATABASE, HL_EDIT, and NUL. We'll run through some examples of these... First, let's take that that STATS button memo that was sent out earlier. Just what was on that memo form? Well,

SELECT TYPE OF LINK!

Link to a HyperLINK App (HAP)

Link to a HAP and do a SEARCH

Link to a HAP and pass data

COMMAND Link

SINGLE CLICK DOUBLE CLICK STARTUP

DESTINATION HAP INFO!

LINE TO HAP NAME! EDITOR

USE CURRENT DATA FILE

USE FILE NAME!

GET FILE NAME FROM LOCAL FIELD

FIELD NAME! CARD PASSPAGE!

DESTINATION INDEX ONE!

LINE 0 0 X-LOC! 0 Y-LOC! 0

CANCEL OK

(Figure A.1)

each memo form has different lines to be filled in:

LINK TO HAP : Essentially this means an indication of which office is the memo going to. There are some modifications to this in the memo as well. Not only can you just send a memo to an office, but once it arrives there, you can instruct that manager to "Link to a HAP and do a SEARCH", or "Link to a HAP and Pass Data". Of course, with either of these last two options

APPENDIX A

you must be specific on what fields you want searched, or which fields you want filled out by the manager. You can even give screen coordinates where you wish the receiving office to locate its response.

COMMAND LINK : This is best seen as an inter-office memo. An office manager here is telling himself to do something. It can be something as simply as "Go to the next record", or a mathematical calculation. Whatever it is, in general, it is staying in the same office.

So, where did our little BASEBALL manager get information to fill out this form? Simply he looked at the Item Information for the STATS button, which was entered when the link was first created in the Builder. Let's take a look at the Linking Info Screen for the STATS button (Figure A.1) :

As you can see, we are sending our memo to another office (or HAP), the EDITOR. Automatically the courier (or HyperLINK) knows which Company (or Module) to address this memo to - HL_EDIT.

Now the other information needed in the memo, is telling the EDITOR Manager which file to load. There is a field in the BASEBALL database called CARD which contains the number for each card. Since this is different in each record we

(Figure A.2)

SELECT TYPE OF LINK

Link to a HyperLINK App (HAP)

Link to a HAP and do a SEARCH

Link to a HAP and pass data

COMMAND Link

LINE #

SINGLE CLICK DOUBLE CLICK SETUP

DESTINATION HAP (HAP)

LINK TO HAP NAME: ADDRESS

USE CURRENT DATA FILE

USE FILE NAME: _____

GET FILE NAME FROM LOCAL FIELD

FIELD NAME: _____ PREFIX: _____

DESTINATION INDEX (HAP) AD.LINK

LINE # B X-LOC B Y-LOC B

LOOPUP METHOD OR PASSED DATA SOURCE

GET FROM LOCAL FIELD: NFRQ

DIRECT PARAMETER

EXPRESSION

CANCEL OK

HYPERLINK

can use this as part of the instructions to the EDITOR manager. The last thing that is necessary on this Linking screen is an instruction about when this memo is sent. For instance, does a user have to Double Click on the STATS button to send the memo, or is a Single Click enough.

As you can see in our example, a Single Click is enough.

The screenshot shows a 'MESSAGE' window with a title bar and a close button. The main content area is divided into two sections. The left section contains the following text: 'To Earle, Geoff', 'Date 04/30/91 Time', 'From Geczy, George of JMG', 'Phone', and a set of checkboxes for 'Telephoned', 'Returned Call', 'Wants Visit', 'Wants to Contact', 'Please Call', and 'Will Call Again'. Below this is the text 'Received by GJ' and 'Message What's new?'. The right section is an address card for 'JMG' with the following fields: 'LAST NAME: GECZY', 'FIRST NAME: George', 'COMPANY: JMG Software Int'l', 'ADDRESS: 892 Upper James St', 'CITY: Hamilton', 'STATE: ONT', 'COUNTRY: Canada', 'POSTCODE: L9C 3A5', 'PHONE: 416-575-3200', 'FAX: 416-575-0283', and 'NOTES: Chief Programmer/Bottlemasher'. The address card has a tab labeled 'JMG' and navigation arrows.

(Figure A.3)

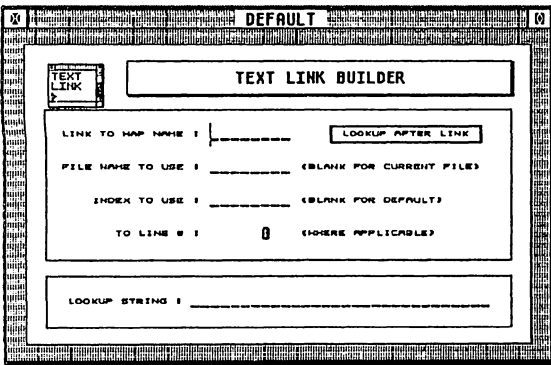
So what is really happening when you click on STATS? The BASEBALL Manager looks at the current record and makes a note of which Card Number you are currently looking at. He then rushes a memo to Shipping that is couriered to the EDITOR at the HL_EDIT company. Once that manager gets the memo, he opens up a window for the Editor and loads in a TXT file with the same name as the card number. Once that's done, he shreds the memo... His job's complete.

For another example, let's look at the MESSAGE office. This is also part of the DATABASE company. It uses links to another office in the DATABASE company, the ADDRESS office. When the MESSAGE office was first created, links were installed in the FROM and OF fields - these both link to the ADDRESS office and perform a lookup on either the NAME or COMPANY fields respectively. Take a look at the Linking Info for the FROM field in the

APPENDIX A

MESSAGE company (the actual name of this field in the Message HAP is "MFROM").

As you can see, the link is invoked by Double Clicking on the MFROM field itself. It has been configured then to link to the ADDRESS HAP, and once there, to search the ADD_LNAM (Last Name) field, looking for a record matching the contents of the MFROM field in the MESSAGE HAP. Using the Message record displayed in Figure A.3 and in the parlance of this analogy... When you are in the MESSAGE office of the DATABASE company, looking at a record of a message to "Earle, Geoff" and you Double Click on the name in the FROM Data Field "Geczy, George", you are sending a memo which in part contains the name of whom the current message is from. This memo is sent to the ADDRESS office, also in the DATABASE company, and instructs the manager of this office to open up a window with the ADDRESS HAP in it, search for a matching record wherein the contents of his "Last Name" (ADD_LNAM) field match "Geczy, George" and display this record.



NOTE : One would think that since the memo is staying within the same company (ie. the DATABASE company) that it would merely have to be walked next door from the MESSAGE office to the

HYPERLINK

ADDRESS office. Well, not quite. MESSAGE still takes the memo to the Shipping desk, the courier still picks up the packet and checks the address (to see which company ADDRESS is really in), and then hands it to the Receiving desk at the DATABASE company. They then take it up to the ADDRESS office, and the manager there gets to work.

So far we've discussed links from DATABASE applications, but links from other sources still follow the same basic rules. As we've mentioned at the start, the only other link creators ("ACTIVE" Modules) are the HyperEditor and the NUL Module.

Let's look now at the HyperEditor. When you call up a link by Double clicking on a word in the editor, a very similar process occurs as with the database. The major difference between the two is the source of the information to fill in the link memo with. Under DATABASE, all linking information is provided in the Application Builder. With the EDITOR, all the same information is entered by using the TEXTLINK form; when you <ALT> Click on a word in the text editor, this form comes up and allows you give all the details necessary in accomplishing the desired link:

The layout is a bit different, but the result is mostly the same. When you select a word, the EDITOR office looks to see if a TextLink entry exists for the word. If it does, the TextLink manager goes and fills in a link memo form just the same as the BASEBALL or ADDRESS manager did. This manager gets the info for the

APPENDIX A

memo from the details in the TextLink record. Note that there is no option here to look elsewhere for information on completing a link; Searches are still allowed, but the info from them must be either contained in the word which is the source of the link, or by the user actually typing a search key in to the TextLink screen. When the link memo is filled in, it is sent, just as before, to the Shipping Desk and then to the courier and then to the destination office.

Another difference people may notice between a DATABASE link setup and the TextLink form is that only the former offers you the choice of how a link is originally called (or "Link on..."). The reason that the EDITOR does not give you this option is that the Editor only has one way to select a linking action, a Double Click. (or <ALT> Click). Single Clicking in the Editor merely moves your text cursor to that location.

That brings us to the last Active Company which we so far have ignored : NUL. This built-in HyperLINK Module functions almost exactly like the DATABASE module. The difference is that it does NOT create or manipulate any database files. The point of this is to allow creating a "Control" type of application, one where there is not data input or output but other features, primarily linking, are still active. Consider a Canadian map application : The main map screen has no data input or output, it only contains an IMG backdrop of a Canadian map with a number of invisible buttons overlaying it. By clicking on any of the provinces you are really clicking on the invisible button overlaying that province. This

HYPERLINK

button simply launches another HAP based on that province that could be a Database, Text File, another Graphic, etc. This type of form is a "NUL" based application, though it is still built with the Builder and operates like a database form does.

We hope that this analogy has shed a little bit of light on the subject of Linking within HyperLINK. It is one of the most powerful elements of the program, and linking can be as simple or as complicated as you choose to make it. Understanding the basic principles as described above can go a long way towards allowing you to create complex links between different applications and different objects.

APPENDIX A

This Page Left Intentionally Blank

APPENDIX B - EXPRESSIONS

Expressions have a number of uses in HyperLINK. For instance:

- In the Report Generator, an expression can evaluate whether to print a certain record or not. For instance, if you wanted to print all products in an inventory list that were out of stock, you would print a list asking for all items where "STOCK = 0". This is an expression, that resolves into a True/False ("yes it is", or "no it isn't") answer.

- In the Report Generator, you can print an expression instead of just a simple field as a print-out item. Your database may contain two fields, one called COST and the other QUANTITY. Your report could then print out a value of "COST * QUANTITY" so you could see the total dollar value of each item (cost multiplied by quantity).

- In the Builder, you could specify an expression field for automatically filling in the current date, or automatically adding a total or calculating a tax amount.

- An expression is used to create a sort index. While often sorts are based on a single field, it is also possible to combine fields (such as LASTNAME + FIRSTNAME) to create sorts based on multiple fields. As well, other operations and functions can also be used on fields used in a sort expression.

APPENDIX B

- You can also use expressions in a link, though this is less often used. It is possible to pass an expression both as data (to fill into a field in another HAP after a link), or to use as a search.

What is an expression? An expression may contain one of two things: operators (ie plus, minus, etc) and functions. Each can be combined as well. The primary use of expressions is to do something with the contents of fields in your HAP, whether it be math or string operations.

For the following examples, assume you have a sample database layout as follows:

<u>Field name</u>	<u>Contents</u>	<u>Type</u>
ITEM	"WIDGET"	Character
MADEBY	"WID-R-US"	Character
COST	"19.50"	Numeric
QUANTITY	"10"	Numeric
LASTIN	"04/15/92"	Date
REORDER	"Y"	Logical (Data Button)

The simplest item in an expression is a FIELD NAME. This is simply, as the title suggests, the name of the field you want to reference.

eg. expression: ITEM
result : "WIDGET"

Next are the simple operators. These consist of operations such as addition, combining strings, and the like. The effect of these operations depends on the type of data you're dealing with,

HYPERLINK

whether numeric or characters.

NUMERIC OPERATORS

+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation

CHARACTER OPERATORS

+	Combine strings
-	Combine strings, put spaces at end.

These operations can also be combined with the use of constant values and parentheses.

eg.	expression :	COST * QUANTITY
	result :	195.00
	expression :	QUANTITY * (COST + 5)
	result :	245.00
	expression :	ITEM + MADEBY
	result :	"WIDGET WID-R-US "
	expression :	ITEM- MADEBY
	result :	"WIDGETWID-R-US "
	expression:	ITEM + COST
	result :	error - can't mix characters and numerics

APPENDIX B

Next down the line are the **RELATIONAL OPERATORS**. These return a TRUE or FALSE value only, and therefore are most ideal for report filters (ie the "which records to print" selection). They can also be used to provide a value for Data Button objects, which also store only a True or False value. It should be noted that True values are stored as "Y" (yes), and False values as "N" (no). As well, True and False can be represented inside an expression with the value constants .TRUE. and .FALSE.

RELATIONAL OPERATORS

=	Equal to
<>	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
\$	Contains string

eg. expression : COST < 20
result : Y

expression : QUANTITY = 10.01
result : N

expression : MADEBY \$ "R-US"
result : Y

Closely related to Relational Operators are **LOGICAL OPERATORS**. These keywords affect a logical result, usually used in combination with the above.

HYPERLINK

LOGICAL OPERATORS

.AND.

.OR.

.NOT.

eg. expression: COST < 20 .AND.
 QUANTITY > 5

result : Y

expression : COST < 20 .OR.
 QUANTITY > 50

result : Y

expression : .NOT. COST < 20

result : N

Finally, expression functions can be used to manipulate data or get special information out of the database routines of HyperLINK. A list of the currently supported functions is as follows:

CTOD(*string*) : Converts a string to a date value. The string **MUST** be in the form CCYYMMDD, ie "19820415".

DTOC(*date*) : Converts a date field into a normal string. A value must be a string to be compared to or combined with other strings. The string is returned in the form CCYYMMDD.

APPENDIX B

- DATE()** : Returns the system date as a date value.
- IIF(*logical_value*, *true_result*, *false_result*)** :
Based on the logical (ie True/False) value supplied, it returns one of two different results. These results can be any value type (string, numeric, date, or logical).
- RECCOUNT()** : Returns the total number of records in database.
- RECNO()** : Returns the record number of current record.
- STR(*number*, *length*, *decimals*)** :
Converts a number to a string value.
- SUBSTR(*string*, *startpos*, *numchars*)** :
Return a portion of the provided string.
- TIME()** : Returns a character string of the current time in the form HH:MM:SS.
- UPPER(*string*)** : Converts a string to all upper case.
- VAL(*string*)** : Converts the string to a numeric value.

HYPERLINK

eg. expression :	STR(COST,10,4)
result :	" 19.5000"(string value)
expression :	IIF(COST<20, "CHEAP", "PRICEY")
result :	"CHEAP"(string value)
expression:	QUANTITY * IIF (COST > 10, 5, 10)
result :	50(numeric value)
expression:	SUBSTR(ITEM, 3, 3)
result :	"GET"(string value)
expression:	LASTIN=CTOD ("19920415")
result :	Y

It is important to remember that you cannot combine different data types, such as adding a string to a number, comparing a date to a string, or comparing a logical (true/false) value to a string. Each data type is unique, though you can use functions such as STR(..), VAL(..), CTOD(..) and DTOC(..) to convert between types.

Just as important is to remember that expressions serve two very different purposes: you can use them to create a new value (string or numeric), OR you can use them to provide a logical result (True/False). The different uses for each are described at the start of this section. A few more examples may help a bit:

APPENDIX B

VALUE RESULT EXPRESSIONS

expression: ((COST + 10) * 20) + VAL("1.5")
result : 3901.5

expression : "The item is a " + ITEM
result : "The item is a WIDGET "

expression : "This item costs " + STR(COST,6,2)
+ " dollars."
result : "This item costs 19.50 dollars."

expression : IIF(QUANTITY < 10, "Less than
10", "10 or More")
result : "10 or More" (string value)

LOGICAL RESULT EXPRESSIONS

expression : COST < VAL ("20")
result : Y

expression : ITEM = "WIDGET"
result : Y

expression : ITEM < "W"
result : N

expression : LASTIN > CTOD("19920401")
.AND. REORDER = .TRUE.
result : Y

HYPERLINK

That gives a brief overview of expressions. In some cases the best way to find out the power and flexibility of expressions is to use them, both in your reports and in your HAPs, and see what you can accomplish.

APPENDIX B

This Page Left Intentionally Blank

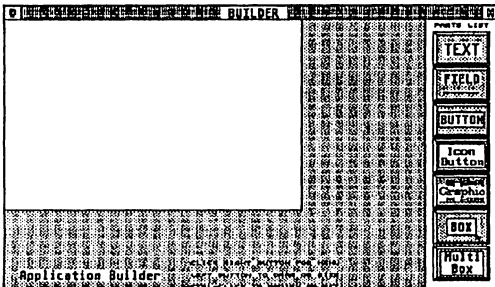
HYPERLINK

APPENDIX C - COMMAND LINKS

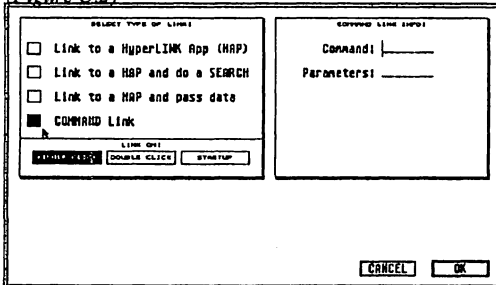
A very useful features of HyperLINK is the ability to create COMMAND LINKS. These are a form of link that you can attach to buttons (whether they be text or icon buttons) that call up "commands": either system type commands (such as window close, etc), or a shortcut to call up selections off a menu or popup.

While available commands depend on what module is involved, most commands are used in Database applications. There, commands such as NEXT, SAVE and such provide a user-friendly, interactive way to customize and automate your own applications.

(Figure C.1)



(Figure C.2)



So, first of all, to create a Command Link: In the builder, create either a BUTTON object or an ICON BUTTON object. Access the linking screen for this object, and then select COMMAND LINK as the linking type. At this point, the right hand side of the dialog will contain two fields, one for the command, and one for a parameter.

The parameter can be used to pass extra information,

APPENDIX C

such as a file or sort name, to a command. Otherwise, if the parameter is blank, HyperLINK will ask for the extra information (such as a popup menu to pick an index, or a file selector to pick a file name).

Once the Command Link button has been established, simply clicking on the button or icon will result in the action specified.

LIST OF COMMANDS

Database Commands: These commands are available to any HAP that uses the DATABASE module (ie any HAP that is a database oriented HAP).

- NEXT** Go to Next record in Database.
- PREV** Go to Previous record in Database.
- FIRST** Go to First record in Database.
- LAST** Go to Last record in Database.
- SAVE** Save any changes to current record.
- NEW** Clear all fields, ready to input a new record.
- FIND** Call up the find dialog.
- SORT** Select a new index name for this database.

HYPERLINK

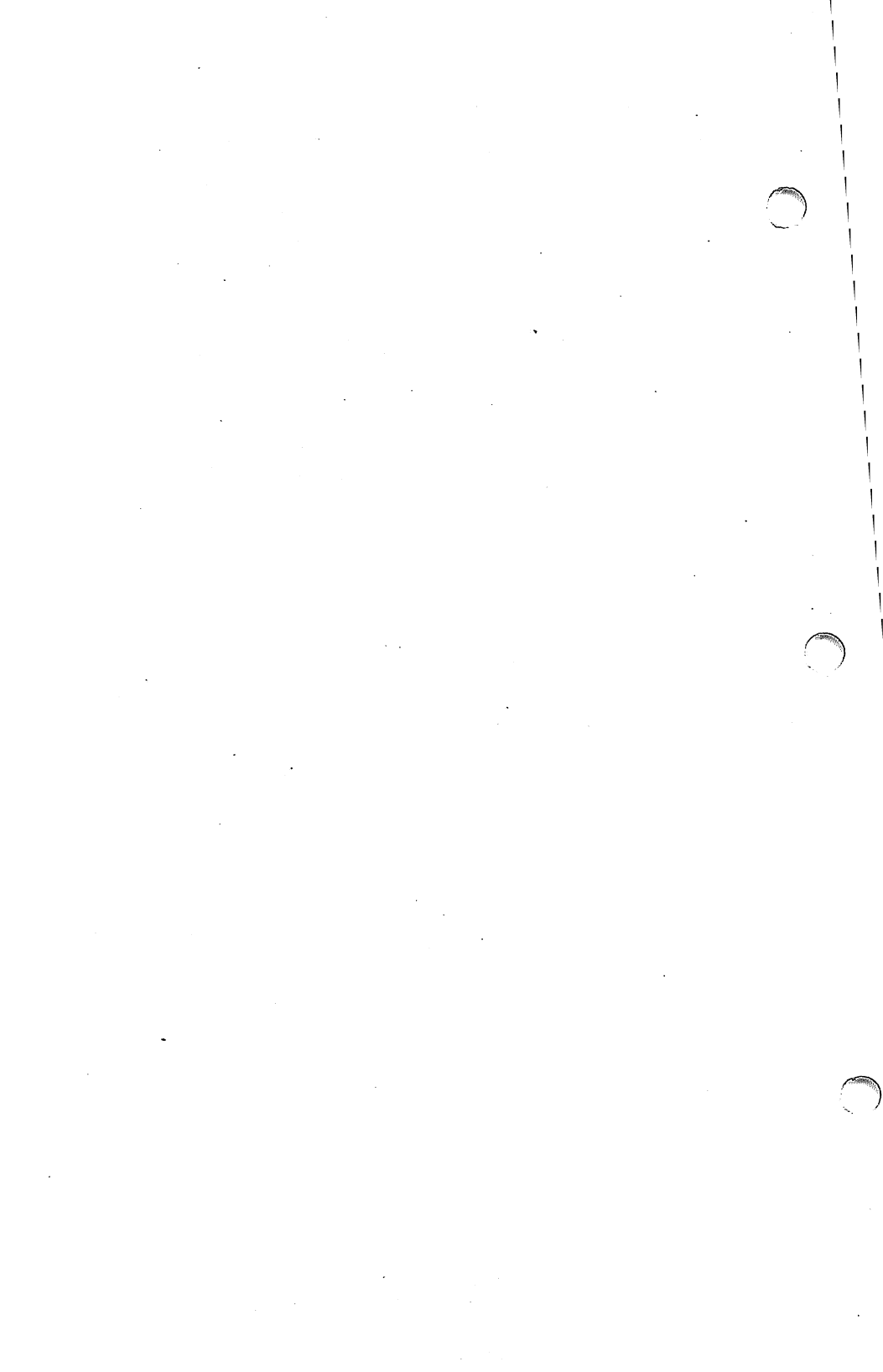
PRINT Call up the report printing routine using this database.

System Commands: These commands are more general, and available to both DATABASE and NUL applications.

WCLOSE Close window.

WMIN Minimize window, ie shrink to icon.

Additional or updated modules for HyperLINK may add more commands that can be used via a Command Link. See the information included with any additional modules or updates for any new commands that may be available.



INDEX

<i>Animation Player</i>	45
<i>Applications</i>	9-11
<i>App Defaults</i>	33, 63, 116-119
<i>App Info</i>	22, 32-33, 60-63, 110-116
<i>App Status</i>	19
<i>On Screen</i>	21
<i>Author ID</i>	114
<i>Borders</i>	99
<i>Builder</i>	21-33, 50-63, 80-119
<i>Builder Menus</i>	21-22
<i>Building with...</i>	
<i>Boxes</i>	28-29, 96-98
<i>Buttons</i>	25-27, 88-91
<i>Fields</i>	24-25, 85-88
<i>Graphics</i>	28, 94-95
<i>Icon Buttons</i>	27, 91-95
<i>Text</i>	23-24, 83-85
<i>Buttons</i>	
<i>Database Item Buttons</i>	26, 90
<i>Icon Buttons</i>	27, 91-95
<i>Linking Buttons</i>	25, 89
<i>Multi Buttons</i>	25-26, 89-90
<i>Multi Data Buttons</i>	26, 90-91
<i>Check Data</i>	119
<i>Checked Objects</i>	100
<i>Clipboard</i>	37, 123
<i>Command Listing</i>	162
<i>Crossed Objects</i>	100
<i>Database Status</i>	19
<i>Desktop</i>	68-79
<i>Desktop Menu</i>	18-19, 73-75
<i>Disabled Objects</i>	100

INDEX

<i>Expressions</i>	30, 101-102, 151-159
<i>Character Operators</i>	153
<i>Functions</i>	155-157
<i>Logical Operators</i>	154-155
<i>Logical Results</i>	158
<i>Numerical Operators</i>	153
<i>Relational Operators</i>	154
<i>Value Result Expressions</i>	158
<i>File Menu</i>	17-18, 71-73
<i>Fills</i>	99
<i>Fuji Menu</i>	71
<i>HALs</i>	16-17
<i>HAPs</i>	16-17
<i>HAP Revision</i>	114
<i>IMG Graphics Displaying</i>	44-45
<i>Indexes</i>	117
<i>Create an Index</i>	117
<i>Default Index</i>	112, 118
<i>Delete an Index</i>	117-118
<i>List Indexes</i>	118
<i>Re-Index</i>	117
<i>Introduction</i>	3-5
<i>Linking</i>	30-32, 58-60, 102-109, 139-149
<i>Command Links</i>	32, 108-109, 144, 161-163
<i>Link to a HAP</i>	30-31, 102-105, 143-144
<i>Link to HAP & Pass Data</i>	31, 107-108
<i>Link to HAP & Search</i>	31, 106-107
<i>Text Linking</i>	39-40, 127-128
<i>Log in Icon</i>	114
<i>Menus</i>	16-20, 71-78
<i>Module Status</i>	19
<i>Modules</i>	9-11
<i>Mouse Buttons</i>	17

HYPERLINK

<i>Network Menu</i>	20, 76-77
<i>Nul Module</i>	45
<i>Numerical ID</i>	114
<i>Objects</i>	
<i>Parents vs Children</i>	23, 80-82
<i>Sorting</i>	84
<i>Outlined Objects</i>	100
<i>Preferences</i>	39, 126
<i>Report Generator</i>	40-44, 129-136
<i>Report Generator Menu</i>	40-41
<i>Reports Building with...</i>	
<i>Data</i>	42-43, 131-134
<i>Text</i>	41-42, 130-131
<i>Report Searches & Sorts</i>	43-44, 135-136
<i>Report Setup</i>	43, 133-135
<i>Screen Locations</i>	105, 118
<i>Secondary Filename</i>	115
<i>Selected Objects</i>	99
<i>Setup (General)</i>	6-8
<i>Shadowed Objects</i>	100
<i>Size IMG</i>	92
<i>Sort Inside</i>	96-97
<i>Sub Directories</i>	7-8
<i>System Menu</i>	19, 75-76
<i>System Requirements</i>	6
<i>Task Status</i>	19
<i>Text Editor</i>	34-39, 120-126
<i>Text Editor Menus</i>	
<i>Edit Menu</i>	36-38, 123-125
<i>File Menu</i>	35-36, 121-123
<i>Options Menu</i>	38-39, 125-126
<i>Title Screen</i>	15-16, 68-70
<i>Transparent Objects</i>	99

INDEX

<i>Utilities Menu</i>	20, 77-78
<i>Visual Screen</i>	29, 98-101
<i>Windows</i>	78-79
<i>Closing</i>	78
<i>Minimizing</i>	79
<i>Moving</i>	78
<i>X-LOC</i>	105, 118
<i>Y-LOC</i>	105, 118

HYPERLINK

THE CREDITS :

HyperLINK is the product of many years of software development, suggestions, basic ideas, and in-depth design, both specifically with HyperLINK as the goal but also long before HyperLINK was a firm product idea.

We would like to thank all the little, and not so little people that have had a hand over the years (whether they knew it or not) in HyperLINK's development.

Hopefully all the effort behind HyperLINK is apparent, and that HyperLINK meets all the demands you have of it.

Programming :	George Geczy
Design Ideas :	George Geczy & David Thompson
HyperEditor :	Eric Rosenquist & George Geczy
Manual :	David Thompson & George Geczy
Testing :	The Entire Staff of JMG Software, Rob Quance, Phil Vaughan, Peter Zalesak, & Mark Chappel.

And a special appreciation to Ted Nelson, who first conceptualized Hypertext, and Xerox for inventing the Graphic User Interface.

HYPERLINK is Distributed by :

JMG SOFTWARE INTERNATIONAL, INC.

892 Upper James Street

Hamilton, Ontario

CANADA

L9C 3A5

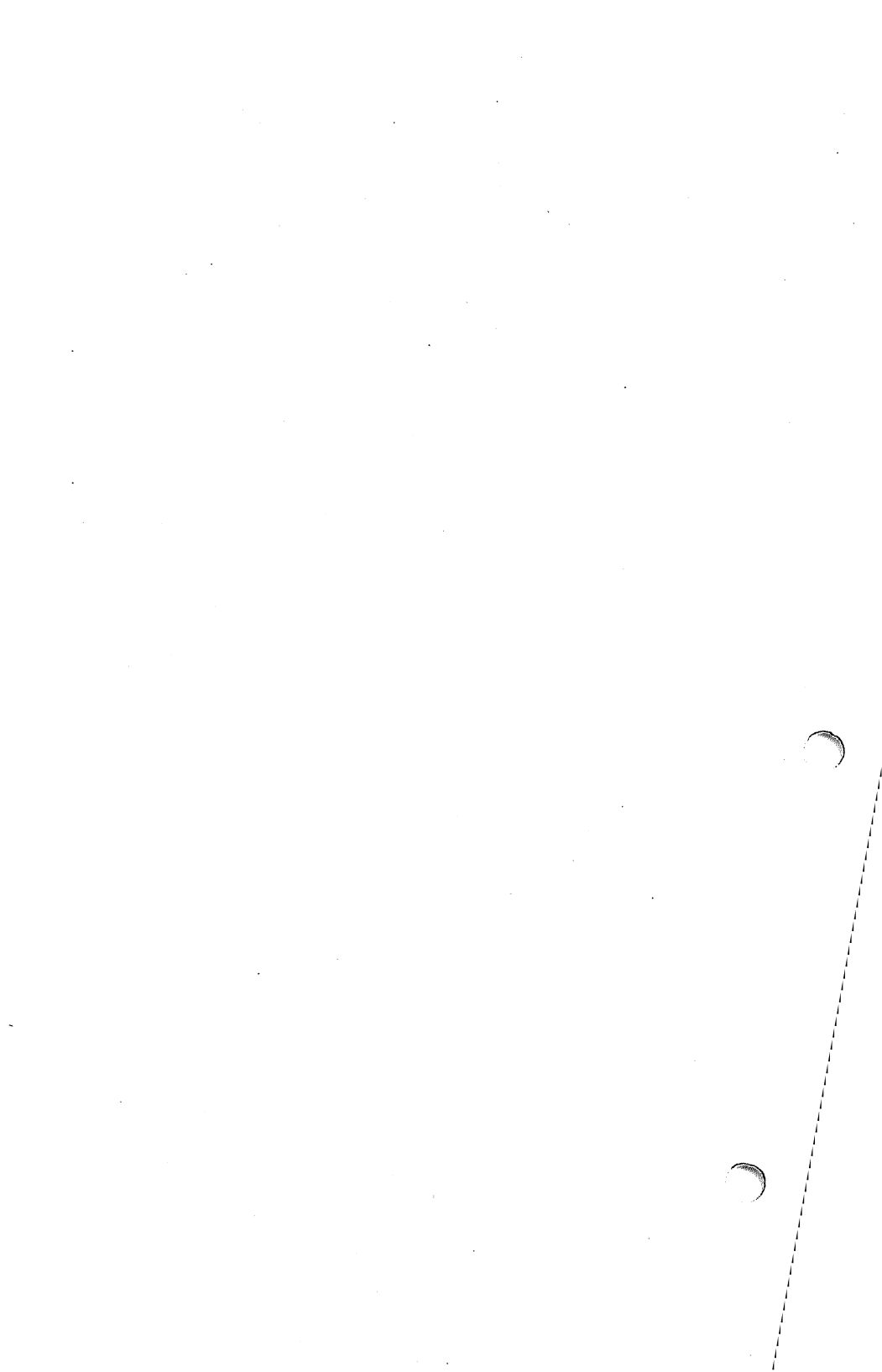
Phone : (416) 575-3201

FAX : (416) 575-0283

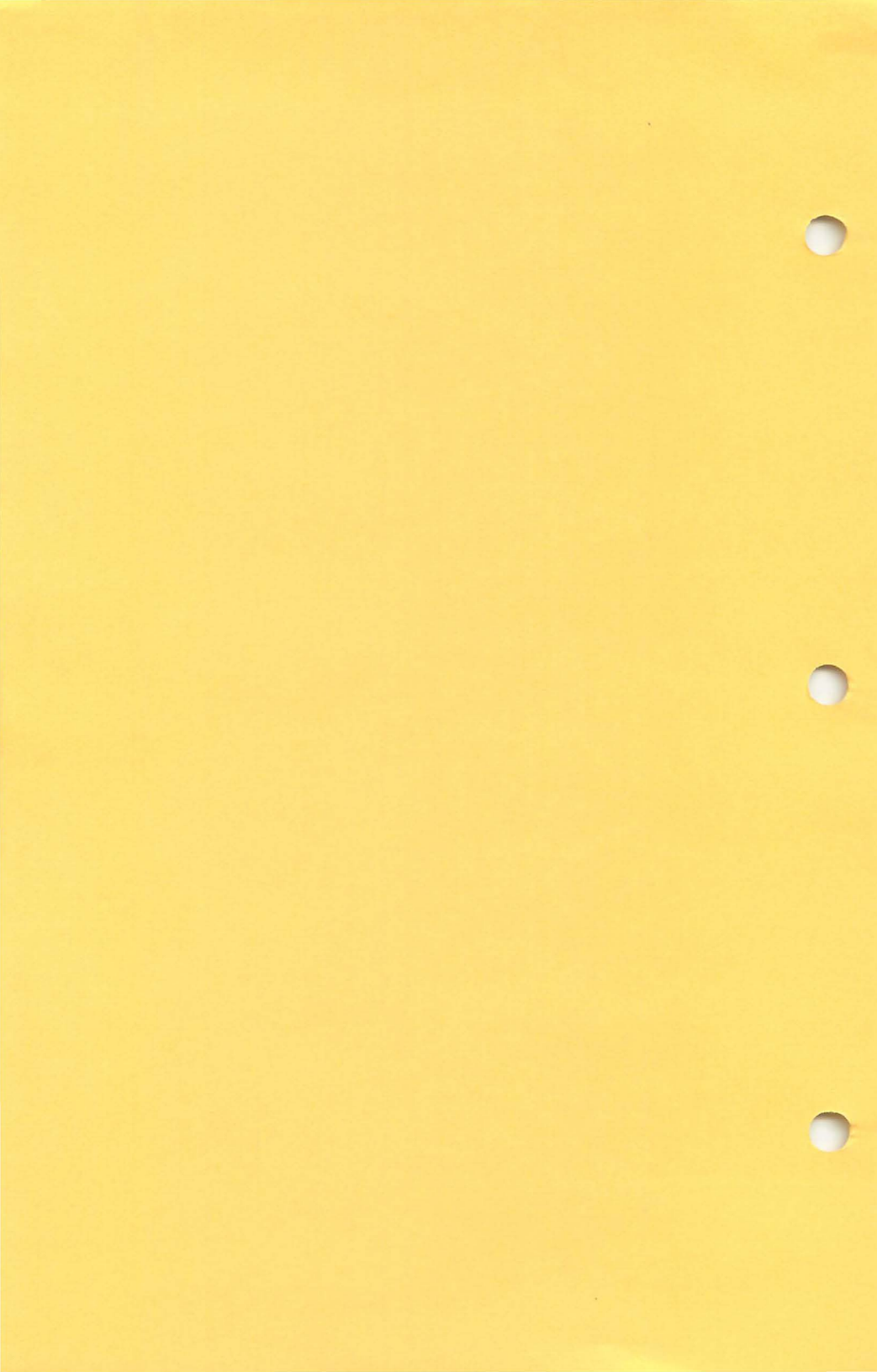
Tech Support : (416) 575-2867

BBS : (416) 389-9064

GENIE : JMGSOFT







HYPERLINK

A Multi-Purpose Object/Database Product

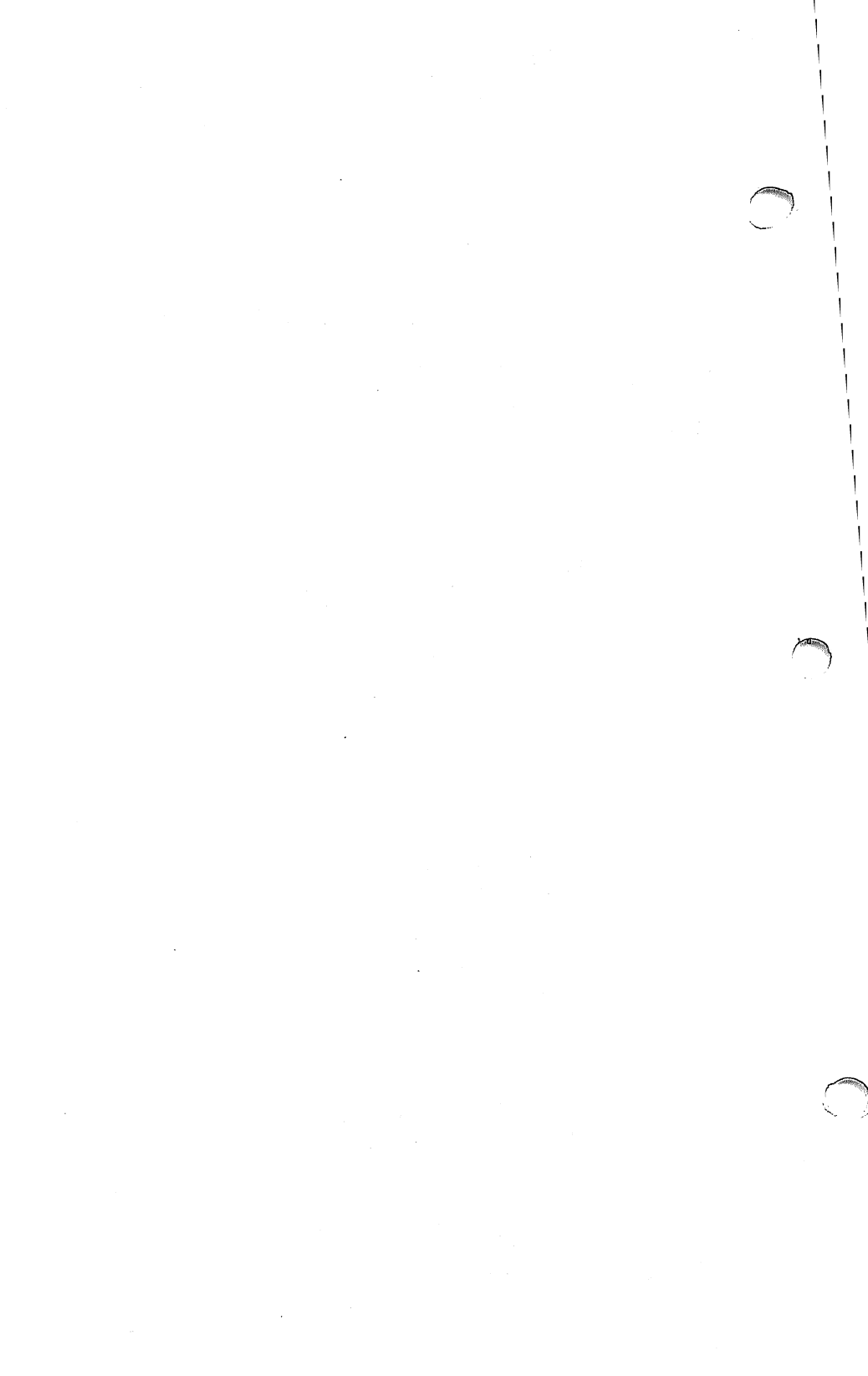
Featuring
Database Control
Relational Objects
Hypertext
Graphics, Sound
Unlimited expandability

and much more...

HyperLINK is a product of
Turning Point Solutions

Distributed by
JMG Software International Inc
892 Upper James Street,

416-575-3201



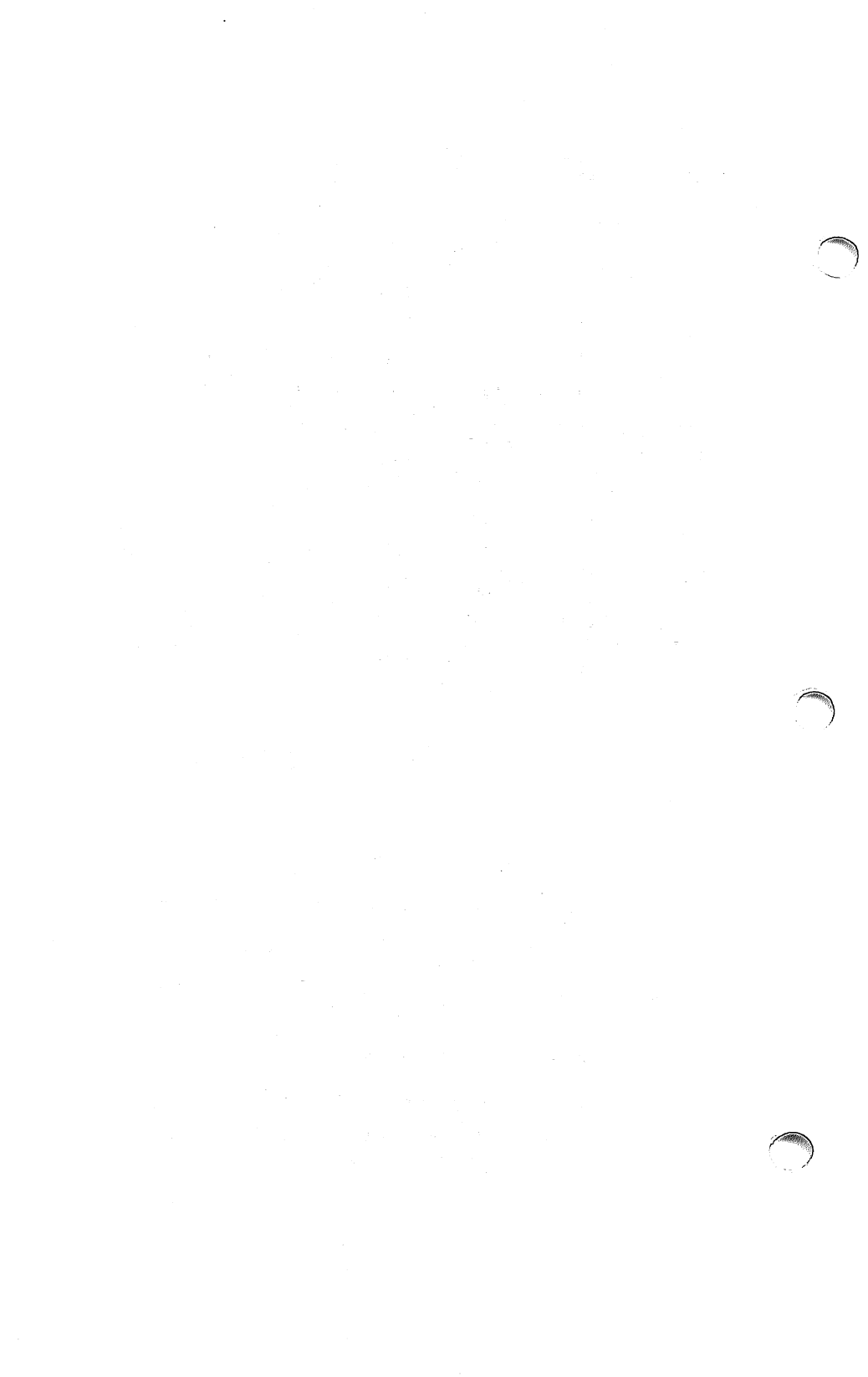
Some Legal Stuff

HyperLINK is not copy protected, so the first thing you should do is make a backup of the distribution disk. Most software publishers are trusting souls who put their life's work up to public scrutiny for the public's own benefit (and maybe some financial reward). Theoretically programmers are appropriately reimbursed for their efforts by people buying their product if it is found to be worthwhile, and they are penalized by people shunning their software if it is not. Also the Atari ST market is not the most fiscally rewarding software market in the world, and if Atari owners want to see quality software continue to be written for their machines they will PURCHASE AND NOT PIRATE software that serves them well. All copies of HyperLINK contain a Serial Number, and as such we will be able to trace back pirated copies to their registered source. As HyperLINK will be updated continually to registered owners who will be linked to their serial number, it is for your own best interests to get registered with us as soon as possible. We will do our best to track down the source of any pirated copies which may turn up and pursue any and all legal options available to us.

Just a couple of more nasty sentences and then we can get on with the business of using HyperLINK...

HyperLINK is Copyright (C), 1991, by Turning Point Solutions. All rights are reserved. This document or the associated software may not be copied or reproduced without the express written permission of Turning Point. This manual and the related software are sold "AS IS", without any warranty as to their performance. Turning Point assumes no liability for any loss or damage (personal or mental) to any individual as a result of using this product.

JMG Software warranties for a period of ninety days the media on which this product is carried. If after ninety days a problem should develop with the media, then return it to JMG Software with a check for \$5 to cover handling and shipping charges, and we will replace it.

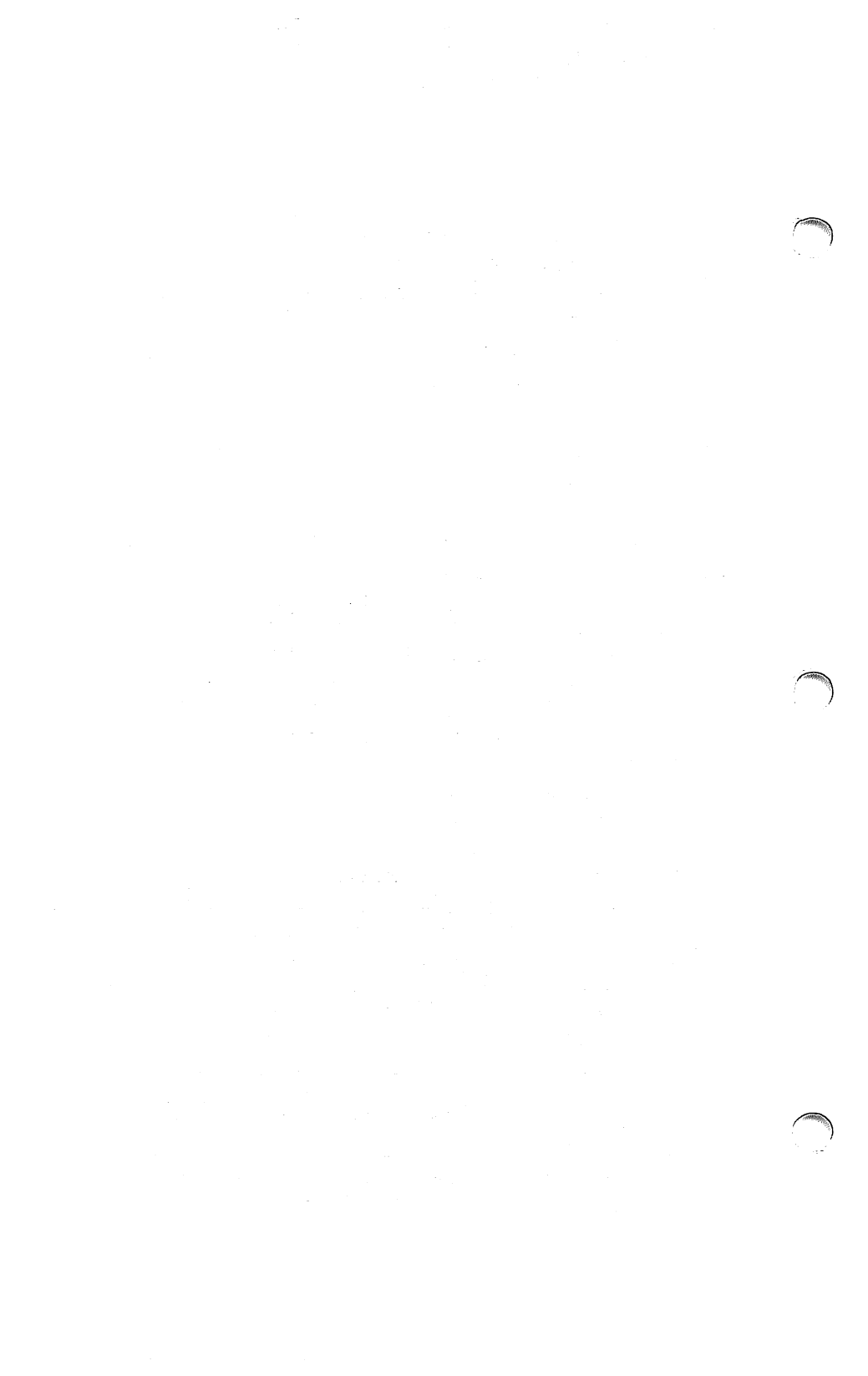


We hope that you find HyperLINK to be the productive tool that it was meant to be. And we would like to thank you for purchasing this product. We will do our best to provide many future upgrades and add-ons that make HyperLINK an even more sophisticated and useful product.

HYPERLINK...

HyperLINK is a Multi-Media Database Management Program that lets you create your own applications simply and quickly. HyperLINK is much more than a Database though. It is based on the concept of Hypertext which was developed mostly in the 1970's. Hypertext is a system that aspires to replace normal, linear text with smaller text topics "linked" to each other. This system can be applied past just text to encompass data, graphics, sound, and external control of other devices as well, hence the popular term Multi-Media.

HyperLINK is made essentially of two parts : HyperLINK Applications, and HyperLINK Modules. A HyperLINK Application can be something as simple as a flat database, or as complex as a mini word processor, relational database, or even a full Inventory/Point of Sale System. While a HyperLINK Module is more of a "tool" for use by HyperLINK Applications in manipulating or collecting data, or controlling external devices or actions. A simple example of a HyperLINK Module is a routine to convert graphics for display in a database application, or a driver for a CD-ROM or Laserdisk player, as well as sound, music, and speech Modules.



A NOTE ABOUT APPLICATIONS AND MODULES :

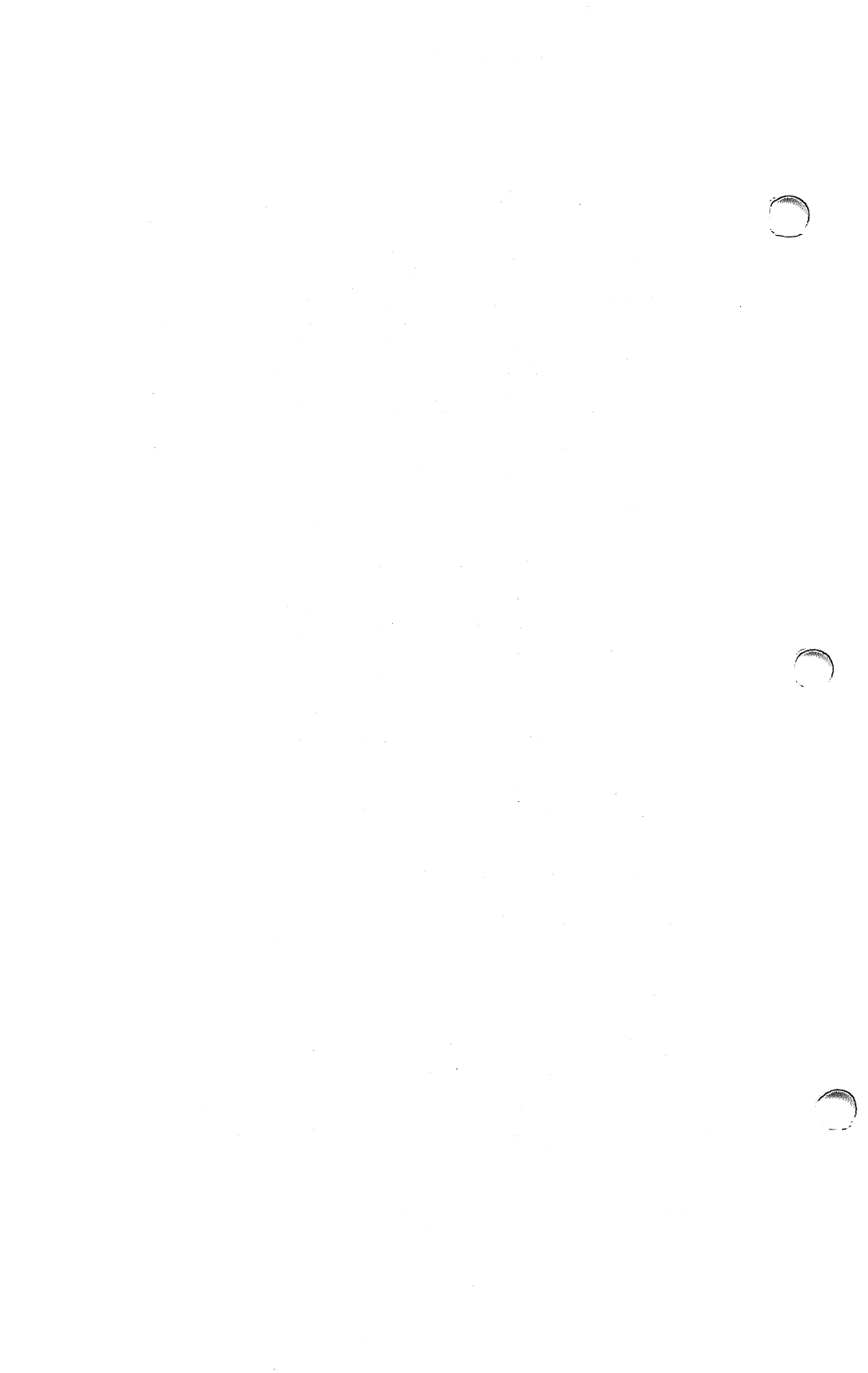
Essentially there are some keys to understanding different types of HyperLINK Applications (HAP's) and Modules : One such element is whether an application is **Passive** or **Active**. The module usually determines this trait in an application. Simply put, a **Passive** application calls up no outside links, (even though it may have links made of it from outside), and an **Active** application calls up a link to another application or module as well as being able to receive outside links.

Another consideration stems around the status of **SPECIFIC** vs **GENERIC** Applications and Modules. A **Specific** Application must be used in conjunction with the matching **Specific** Module. A **Generic** Module allows many different types of applications to be created from it. As an example, HyperLINK's Database is considered a **Generic** Module because you can create many applications from it. On the other hand, HyperLINK's Builder Application and Module are **specific** because both require the other.

CHART OF MODULES AND APPLICATIONS :

MODULE	"HAP"	STATUS
HL_EDIT	EDITOR	ACTIVE / SPECIFIC
DATABASE	----	ACTIVE / GENERIC
HL_BUILD	BUILDER	---- / SPECIFIC
ANIMATE	----	PASSIVE / GENERIC
HL_IMG	GRAPHICS	PASSIVE / SPECIFIC
HL_SND	DSOUND	PASSIVE / SPECIFIC
HL_TALK	TALKER	PASSIVE / SPECIFIC
NUL	----	ACTIVE / GENERIC

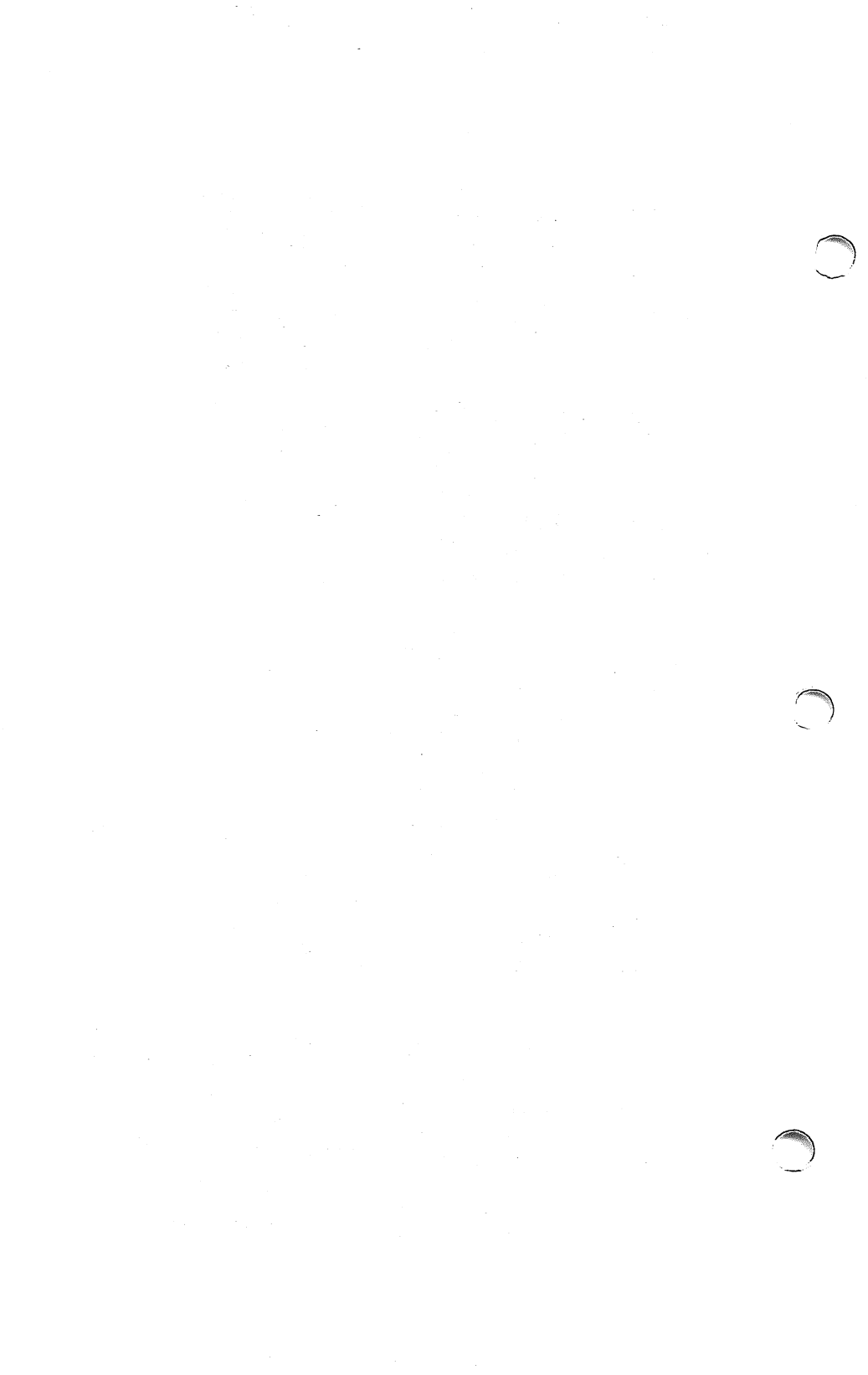
HAP stands for **HyperLINK AP**plication.



One of the simplest examples of a what HyperLINK can do would be the Message/Address Application set developed for HyperLINK. The Phone Message App lets you take standard phone messages, enter them on a graphics-based screen and store them in database. But then, to call up information on a company or a person named in that message, just click on the name and it will display the Address book form with any information available. You could even create more links, such as an appointments link that would display a calendar form for appointments made with that person. You could then zoom in on one particular day's schedule, and even look at the Address or Message record of someone else from that day.

HyperLINK can create links between any sort of data, from one Application to another, or even from one Module to another. All you have to define is how to do the link, and HyperLINK will do the rest. For instance, HyperLINK has a built-in dBase compatible manager. It automatically knows how to look up a name found in one database from within another database, you just tell it what fields to use. For more involved link, you can include custom modules for the application. An assortment of such modules is provided with HyperLINK, any any others can be added at any time (ie programmed in C and loaded into HyperLINK with its Module-load feature).

Each Application when invoked, opens its own window. This way a full history of links can be present on the screen at the same time in different windows. Having all linked applications open at once allows you then to edit any link in a particular "chain". One HAP may be opened more than once as well. In the case of the Message HAP for instance, you may want more than one message displayed on



screen at the same time (making this the first program to use the "cluttered desktop" metaphor).

NOTE : Only one HAP Window is ACTIVE (or editable) at any given time, regardless of how many HAPs are currently open.

What makes HyperLINK the useful tool it is, is the powerful Application Builder built in. As its name suggests, this feature lets you construct Application "Forms" and databases of your own, and change them later on. In addition, it allows you to specify the "links" between applications and modules, and provides a way to combine different functions and modules together. With the Application Builder, HyperLINK can be tailored to any number of uses.

One of the most popular uses for the "Link" function of HyperLINK is that of a computer-based encyclopedia. As already exists in the Mac and MSDOS worlds, a Multi-Media environment such as HyperLINK would allow you to search an encyclopedia textually then link it graphically and with sound. For example: Looking up in a computer encyclopedia the topic Blue Whale. By linking the text with graphics, a picture of the whale could be brought up on the screen. Further by linking these to a database of sounds, the digitized whale song could be listened to. All this through one program mixing data in different formats (sound, graphics, text) to provide a more complete overall view.



Specifications...

* Fully dBase III / dBase IV compatible database manager included.

* Support for Moniterm and other special monitors; support planned for all TT graphics and colour capabilities.

* Launch feature to run other programs from within HyperLINK; includes parameter passing and clipboard use if the destination program supports it.

* dBase data support, Text support, Graphics support, Digitized Sound support, Speech support, and HyperLINK Module support all standard.

* Document hypertext capability standard, allowing use of existing or new text files as hypertext documents or HyperLINK control forms. (ie link text to text, text to graphics, or text to another HyperLINK form, Application, or Module.)

* Ability to actively link two computers together to share, real-time, information. (Serial, Midi or other network capable, by loading "Network Modules". Full tracking of file sharing and record locking.)

* Support Modules for Atari's CDAR-504 CD-ROM player will also be included in future releases of HyperLINK as product availability for this medium increases.

* "HyperLINK Module Protocol" will be published and distributed to allow third parties and experienced end users to create HyperLINK Modules for special purposes. (Use of custom HyperLINK Modules will effectively allow virtually anything to be linked to anything.)

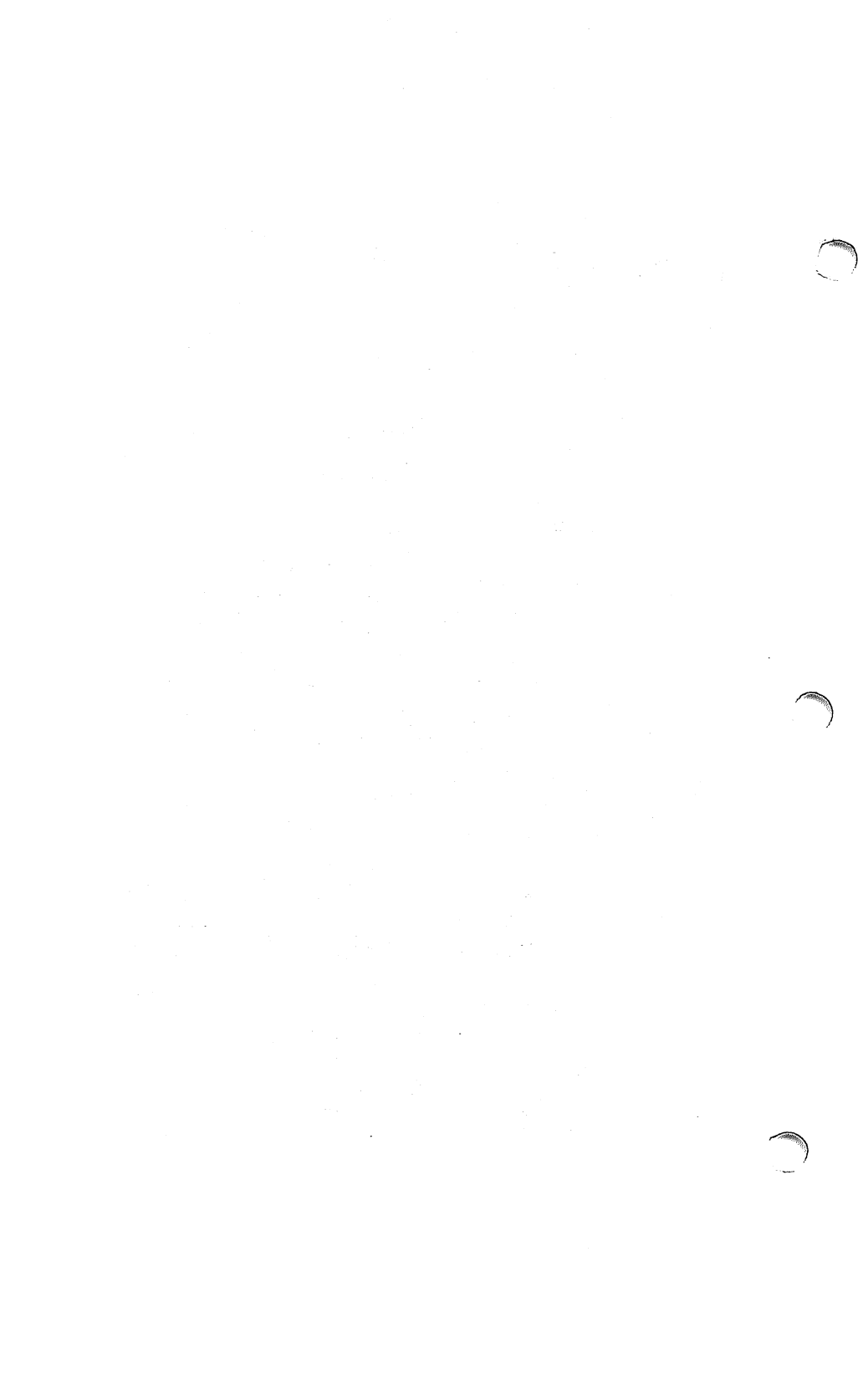


Getting Started with HyperLINK...

Some knowledge of the Atari ST, and of the GEM environment of the ST will be assumed throughout this manual. If you have any problems understanding the instructions, please consult your owner's manual on the use of the ST. If this still does not help, check with your local dealer or user group, or give us a call.

While HyperLINK will run okay from just Floppy Drive Systems, like any database program, the way it operates definitely lends itself better to Hard Drive Systems. If operating HyperLINK from Floppy, remember NOT TO CHANGE DISKS unless requested by the program itself. Also, the more memory your computer system has, the more HyperLINK applications and modules you can have loaded at one time. For any reasonable use of HyperLINK you should have at least 1 Meg of RAM.

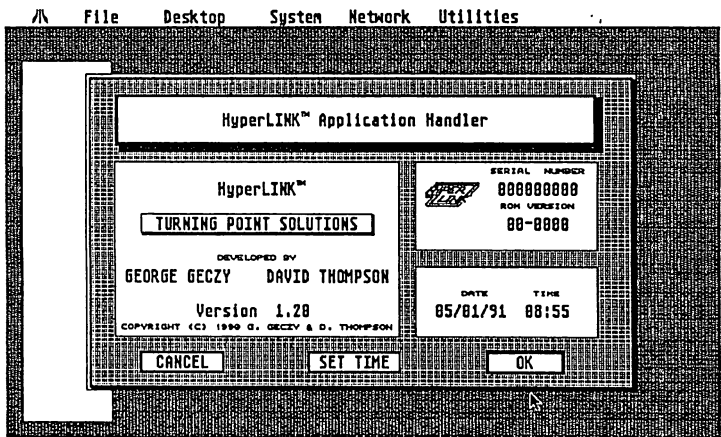
After you have backed up HyperLINK to another floppy (or to your hard drive), the next step is to get right in and start the program up. The program and all but one of its associated files, come on the original in a sub-directory called HYPRLINK. HyperLINK and its associated applications and modules should all be located in the same sub-directory of your working copy when room permits. The HyperEdit.ACC file though should either be on the floppy you boot your system with, or in the root directory of the C drive for those with Hard Drive Systems.



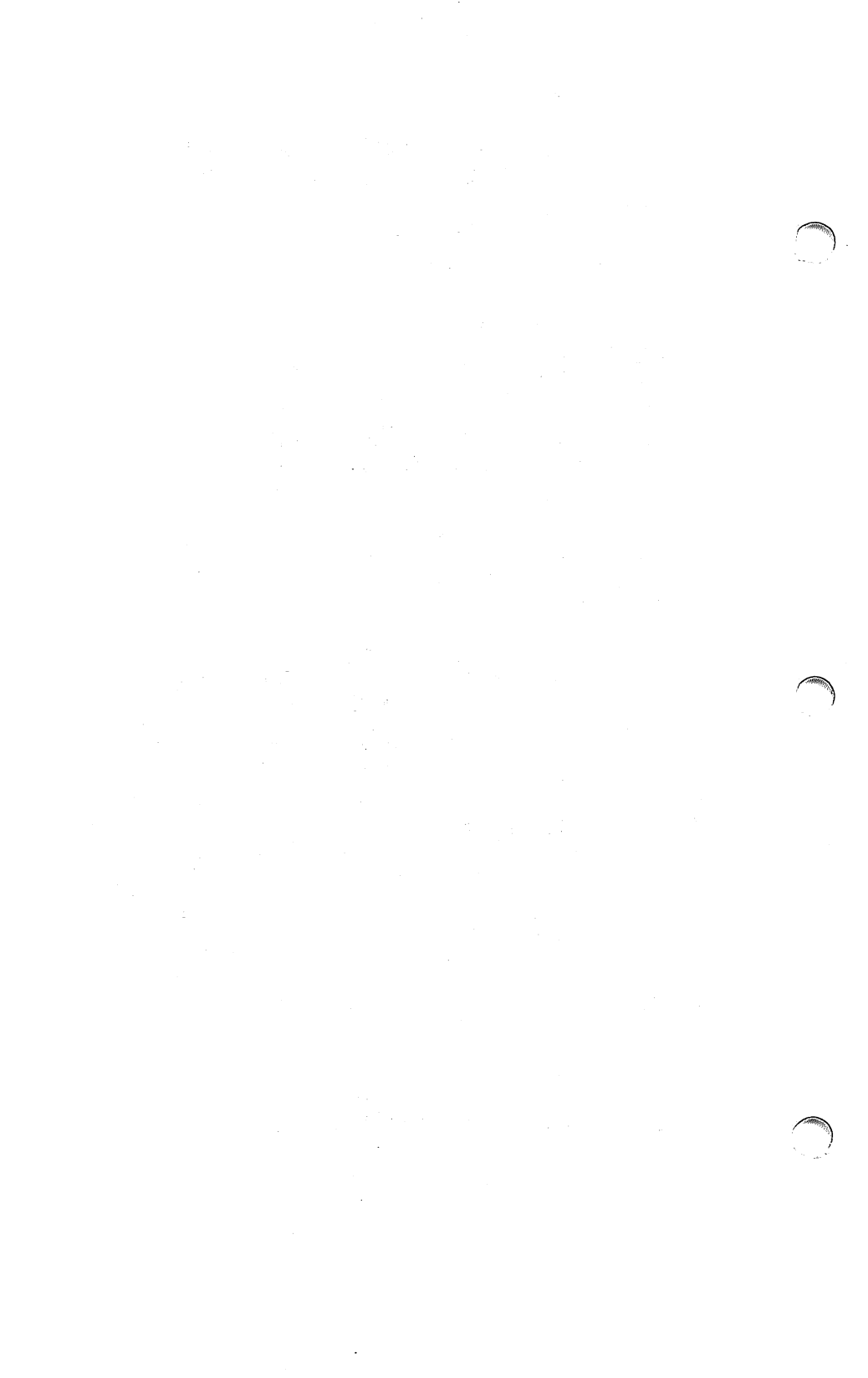
All Applications and Modules should be in the same sub-directory. While not required, this makes it easier to manage the startup and maintenance of HyperLINK. All Data files HAVE TO BE IN THE SAME DIRECTORY while the program is running.

Applications and Modules can be loaded in from other directories but you will have to update the default paths for them to be able to load in automatically upon start up of the program. One of the first updates to HyperLINK will provide for the data files to be in separate sub-directories from the actual program.

So now HyperLINK is essentially ready to go. To start up Hyperlink go into its sub-directory and double click on the H_LINK.PRG icon.



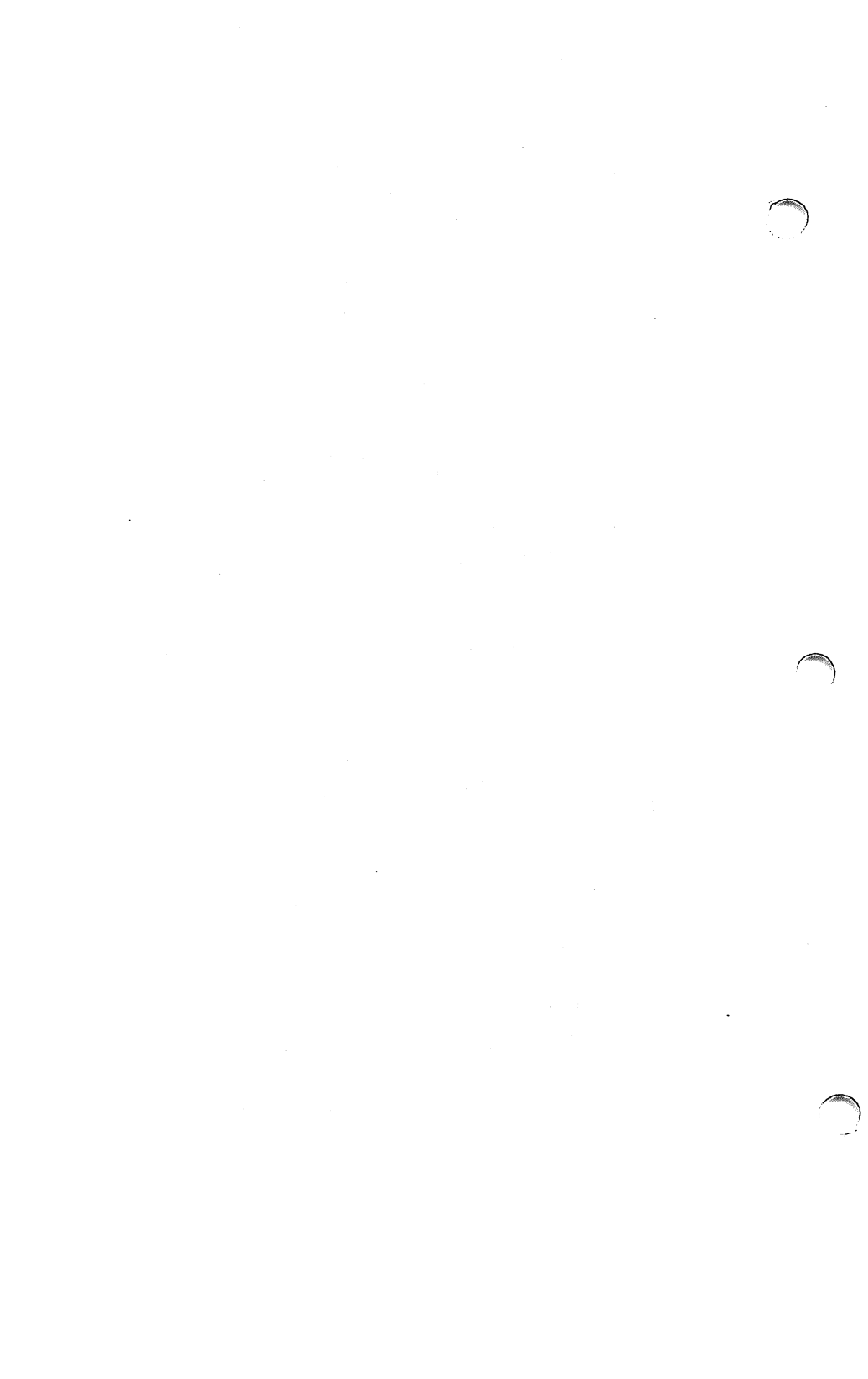
HyperLINK boots up and shows off its Title Screen which gives credit to all the little people that made it happen. The Title Screen of HyperLINK also provides an opportunity to set the Date and Time of your system. It is important that this be accurate as HyperLINK's data files will be able to use the system date and you will want them to be accurate! Once



you have entered the correct time and date, or just hit <Return> to go on, you will be in HyperLINK program itself and all the HyperLINK Applications and HyperLINK Modules that come on the distribution disk will have also loaded in. HyperLINK is now ready for use.

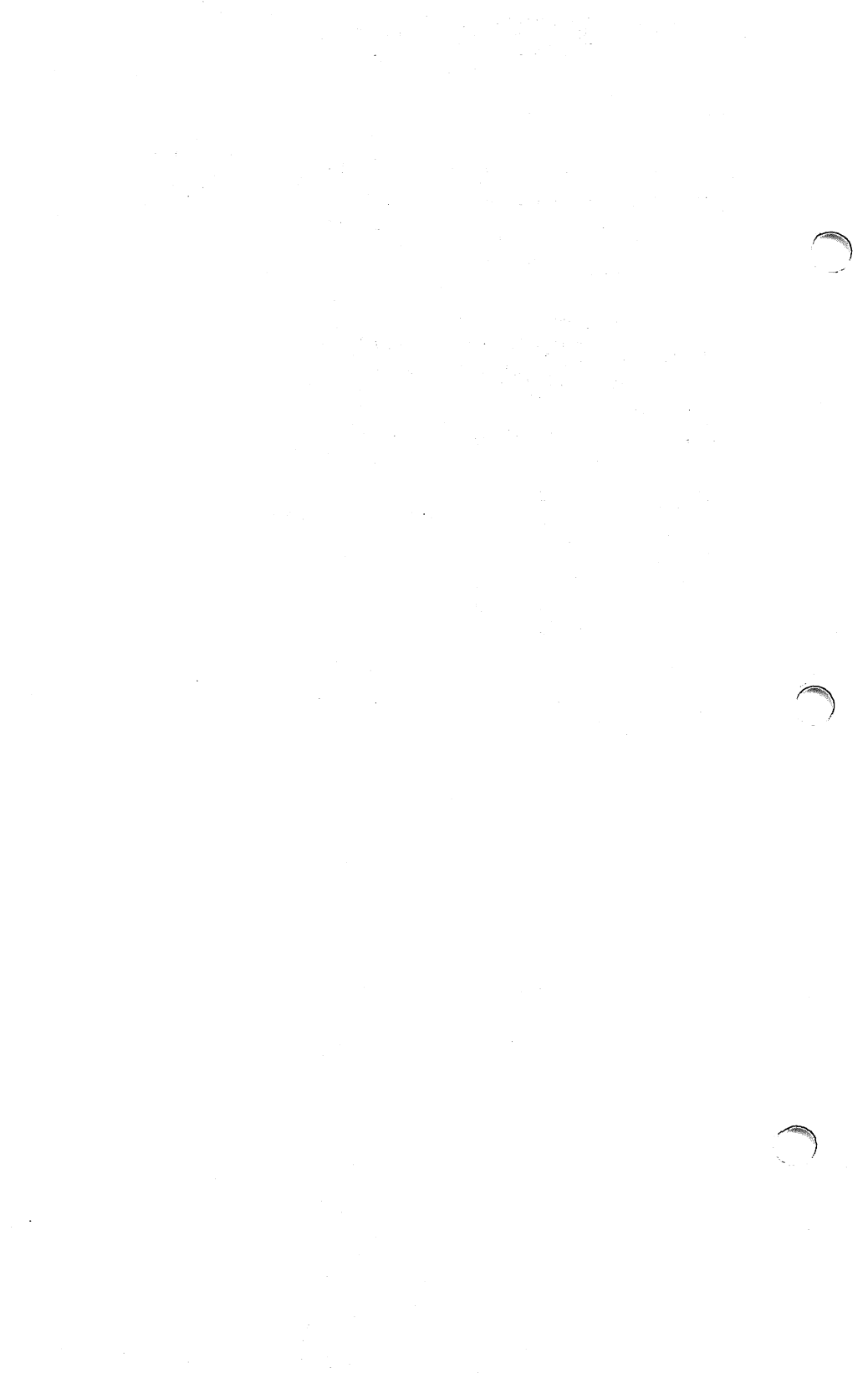
When the program starts up, all the HyperLINK default applications load in on the left hand side of the screen. To call up one of these applications, such as the Telephone Message Book, simply point at the application you want and click the left hand mouse button. As an example, should you click on the Message Book, then you will pop the standard message form. At this time you can enter a new message or bring up existing messages. After the message form is loaded up, click on the right hand mouse button to bring up your options menu. Selections that can be made from this menu are : (typical of apps using the database module)

- * Save - Saves application records into database
- * New - Gives you a blank form to add a new record
- * Find - Finds a record in the database
- * First Record - Goes to first record in application
- * Last Record - Goes to last record in application
- * Next Record - Goes to the next record in application
- * Prev Record - Goes to previous record in application
- * Sort By - Picks an index (already created) to use to view or sort a given application
- * Delete - Deletes a given record in your database
- * To Clip - Copies current application record to a clip board
- * Print - Prints out current application record



If you hold down the <Alternate> key when you first click on the application, that brings up that application's default settings screen. For each application you can view or set :

- * Application Name (pre-set)
- * HyperLINK Module Name (pre-set)
- * Database File application will use
- * Default Data Field to use as an index
- * Display a list of available indexes
- * General Filename for use by this application
- * Application Options
- * Set to default load of first record in database
- * Module Defaults
- * Create Indexes
- * Re-Index a database
- * Delete an Index
- * Pack Data
- * Check Data
- * Backup Data



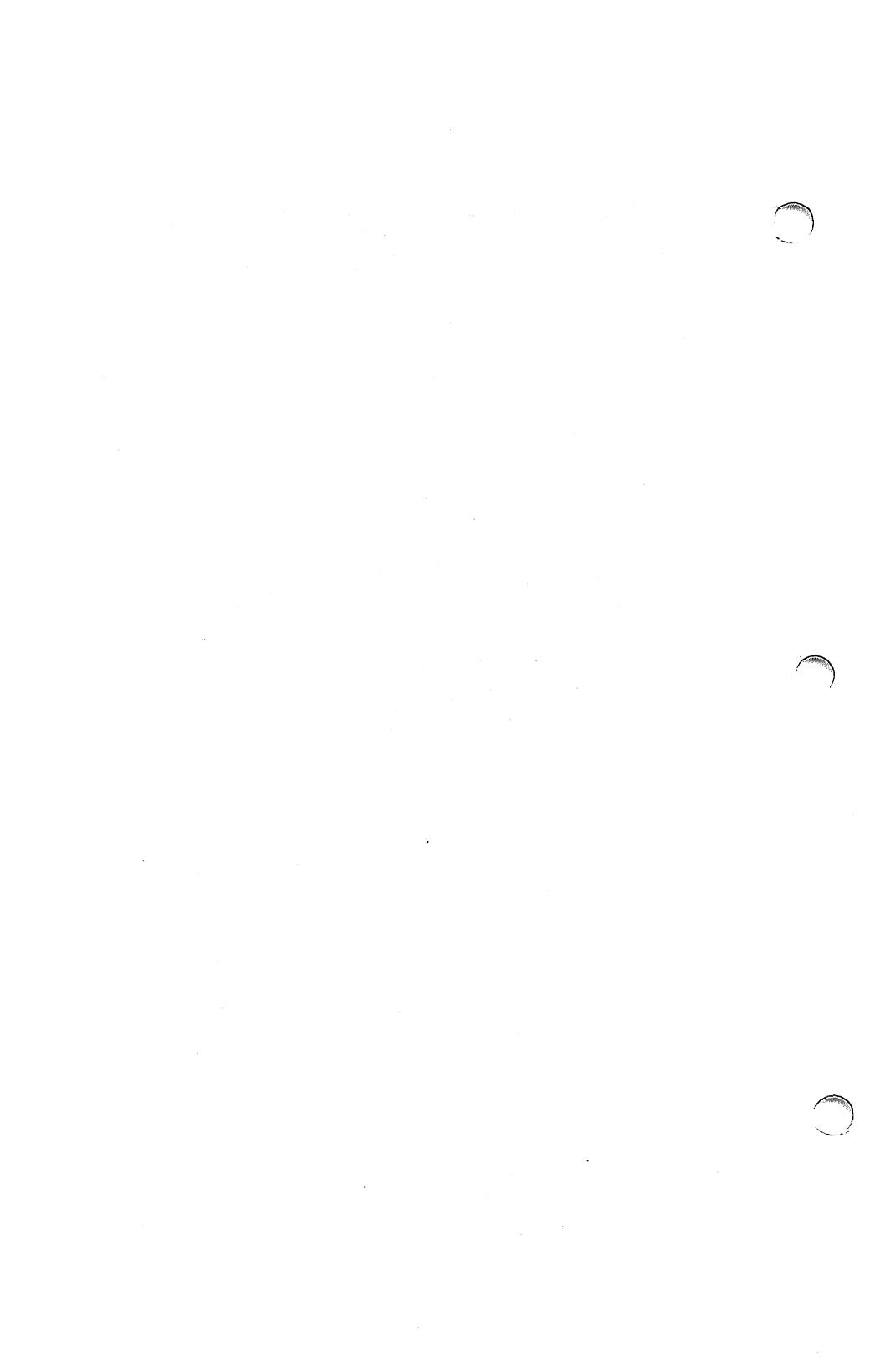
USING AN AVERAGE FORM :

(Does not apply to Text Editor-HyperEdit)

Common to all HyperLINK Applications are some very standard keys and buttons. First of all by pointing at an application on the left hand side of the screen and clicking with the left mouse button, a window for the application is opened on the screen (the limit for open windows in HyperLINK is seven at a time). You can then click on the same application again and you will notice another window is opened for this same application. You may want to have more than one window open at a time for some applications for many different reasons. For instance, you may want to have two address cards displayed on screen at the same time. Rather than opening just one window and having to go back and forth between records, simply open a second window and display both.

On each application window there is the title bar such as "Message". To the left of the title bar is a button that when clicked on with the mouse will close this HyperLINK window. To the right of the title bar is a button that when clicked on with the mouse will "Minimize" that window. What this means is it closes the window to another part of the screen keeping it ready at whatever record it has already loaded, to be called up again without the need for re-searching the database. When a window is minimized it does not count as an open window so is not part of the limit of seven.

Other than just opening a HyperLINK window, you can also call up the default settings for each application. To do this, hold down the <Alternate> key while clicking on the application in mind. As has



been previously mentioned, this will give you the opportunity to change various settings for each HyperLINK application. To go into a little more detail on what can be done here, let us look more closely at the options available :

APPLICATION DEFAULTS

APP NAME : (PRE-SET) The Name Appearing on an App's Title Bar.

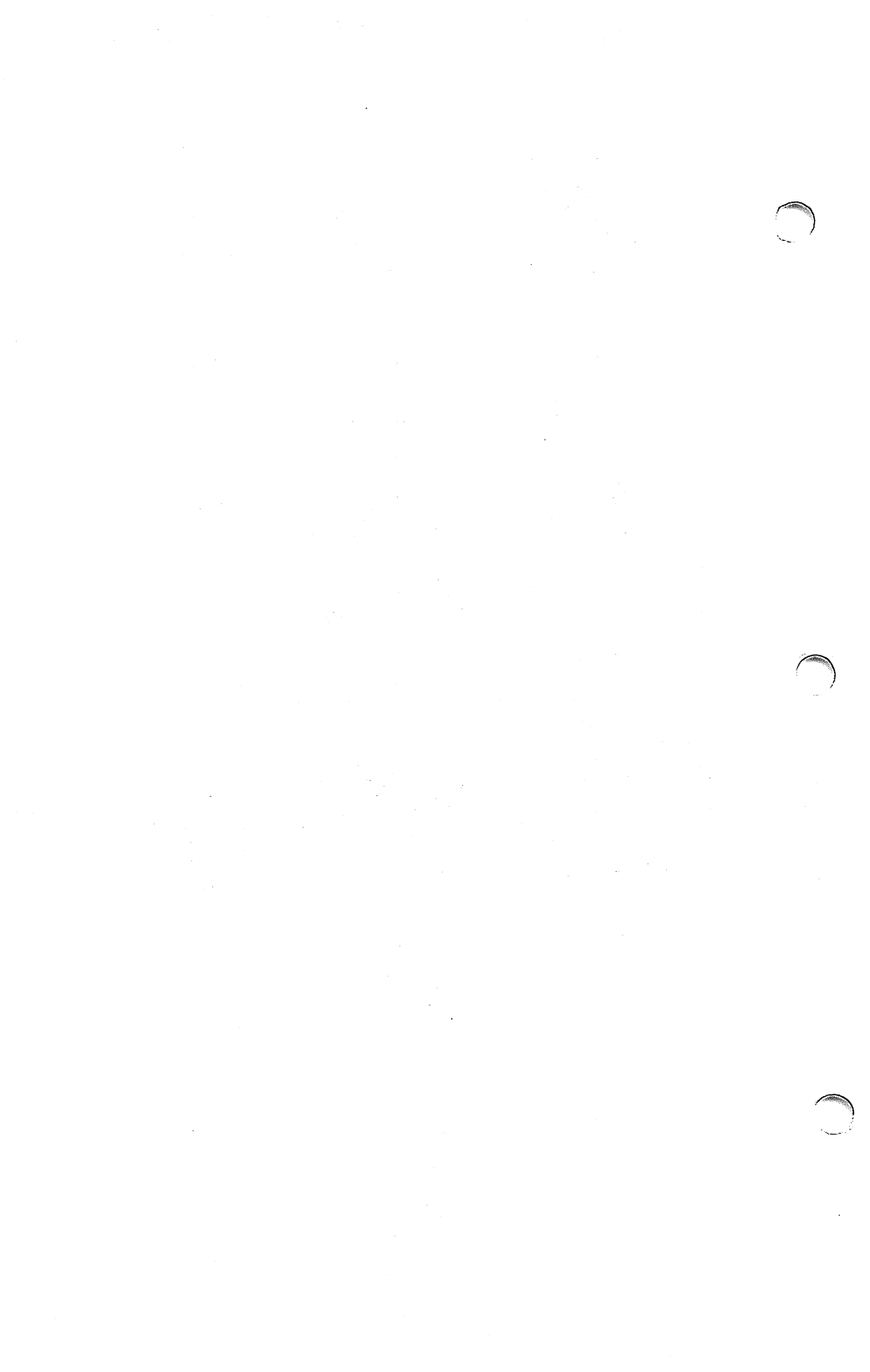
MODULE NAME : (PRE-SET) Which HyperLINK Module an App uses.

DATABASE FILE : The default database file that is associated with each HyperLINK Application.

DEFAULT INDEX : The default index on which a HyperLINK Application's database is sorted.

GEN FILENAME : The general filename that an Application can use if it is so configured.

DB TYPE : There is the option for several types of Databases within HyperLINK. (Of course since HyperLINK was written specifically for its own database, and since this database module comes with HyperLINK, it is the one recommended for use.)



APPLICATION DEFAULTS (cont'd)

APP OPTN :Once again this is something passed on to HyperLINK applications if it is needed. As a sample, the HyperLINK animation module uses this in some applications to set speed of the animation.

1ST REC LOAD :This sets whether or not a HyperLINK application will load its first database record when you open that applications window.

MODULE DEFS :Mostly for future use, this gives the options for future HyperLINK Modules to contain their own default settings as part of the module.

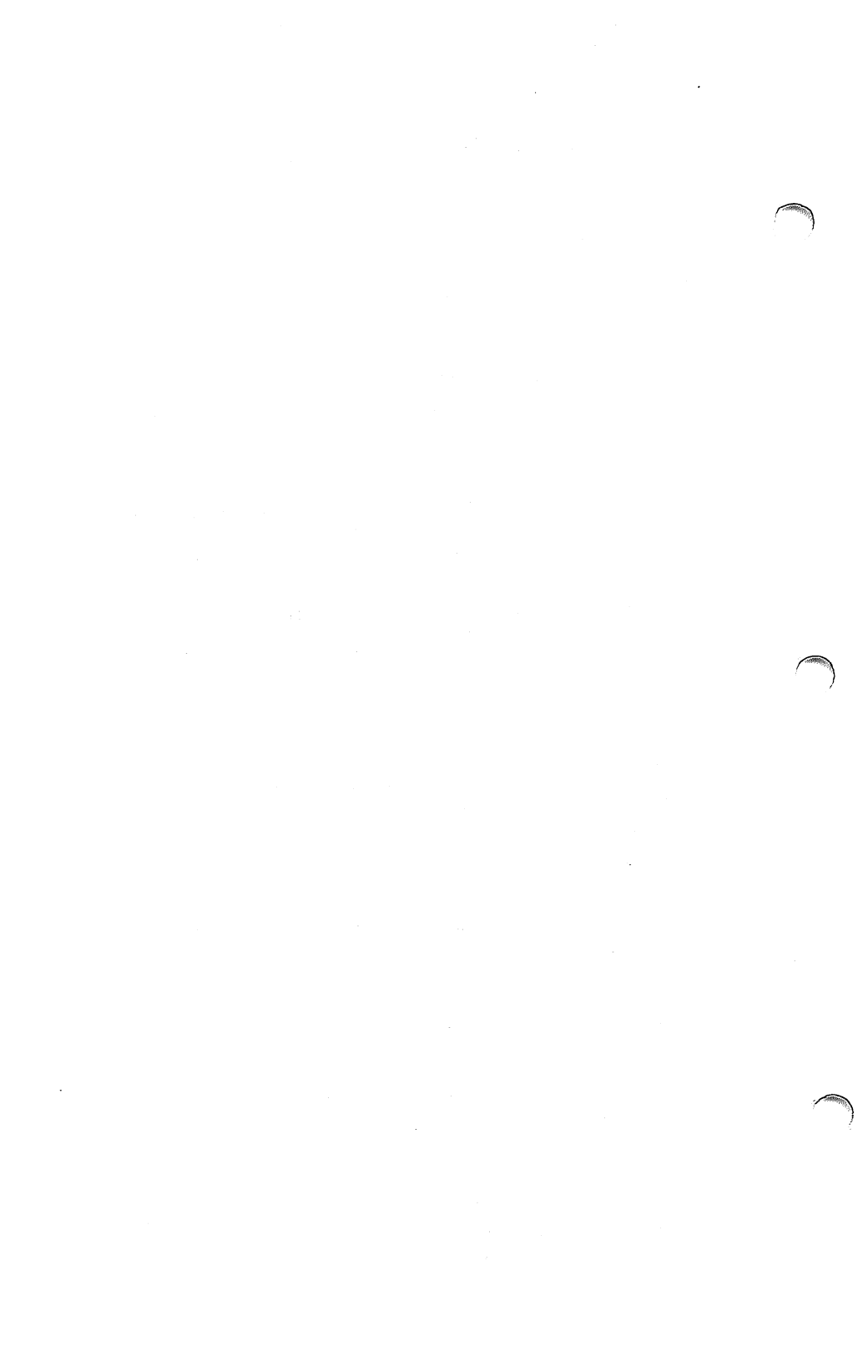
CREATE INDEX :This allows a user to create an index on any data field in a HyperLINK application. You have to tell HyperLINK which data field to index in an application, and you have to give the index a unique name.

RE-INDEX :As happens with all databases at one time or another, an index can get corrupted somehow. This option allows you to have the computer re-generate an index.

DELETE INDEX :Just as the name implies, this options deletes an index file that has already been created.

PACK :Pack is a DBASE III Command needed since the HyperLINK Database is DBASE compatible. When a record is deleted, it is only marked as a deleted record and is only actually wiped from your disk when your data file is packed.

CHECK DATA :Checks the integrity of the data and rebuilds the deleted record list if there are any un-packed deleted records.

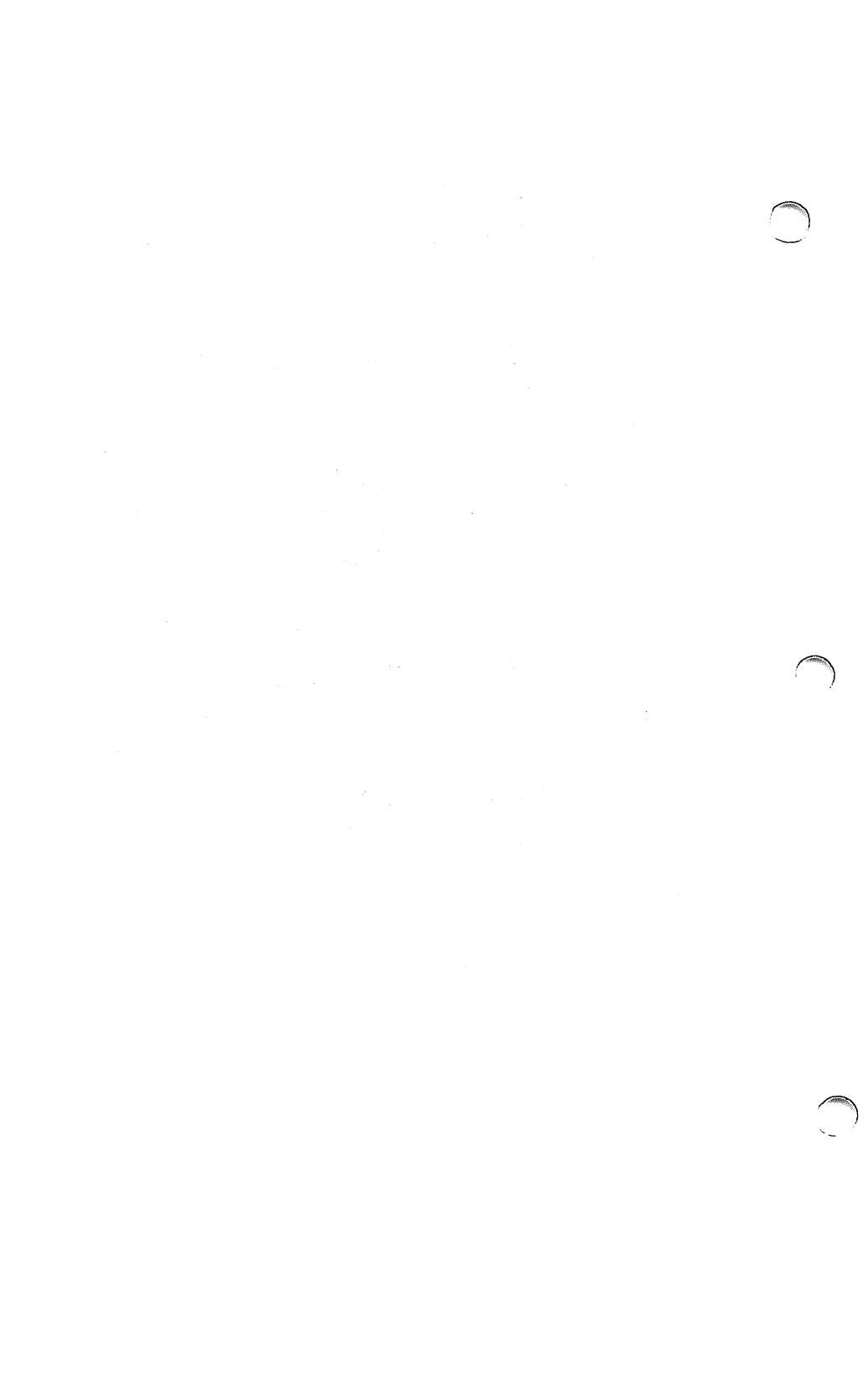


Starting with the default applications that come with HyperLINK, let us take a look at how you would go about using them...

The MESSAGE Application...

Simply put this is an application for keeping track of one's telephone messages. When you call it up you will see a form very reminiscent of the average message pad in everyone's office. Now if George Bush got back to the White House one afternoon and wanted to see who had called while he was out, using

HyperLINK all he would have to do is call up the message form. He may have already set up an index based on whom the message is to (eg. George Bush) so all he has to do is call up the message form, find "George" and then just step through his individual message.



The Data Field names in the MESSAGE Application :

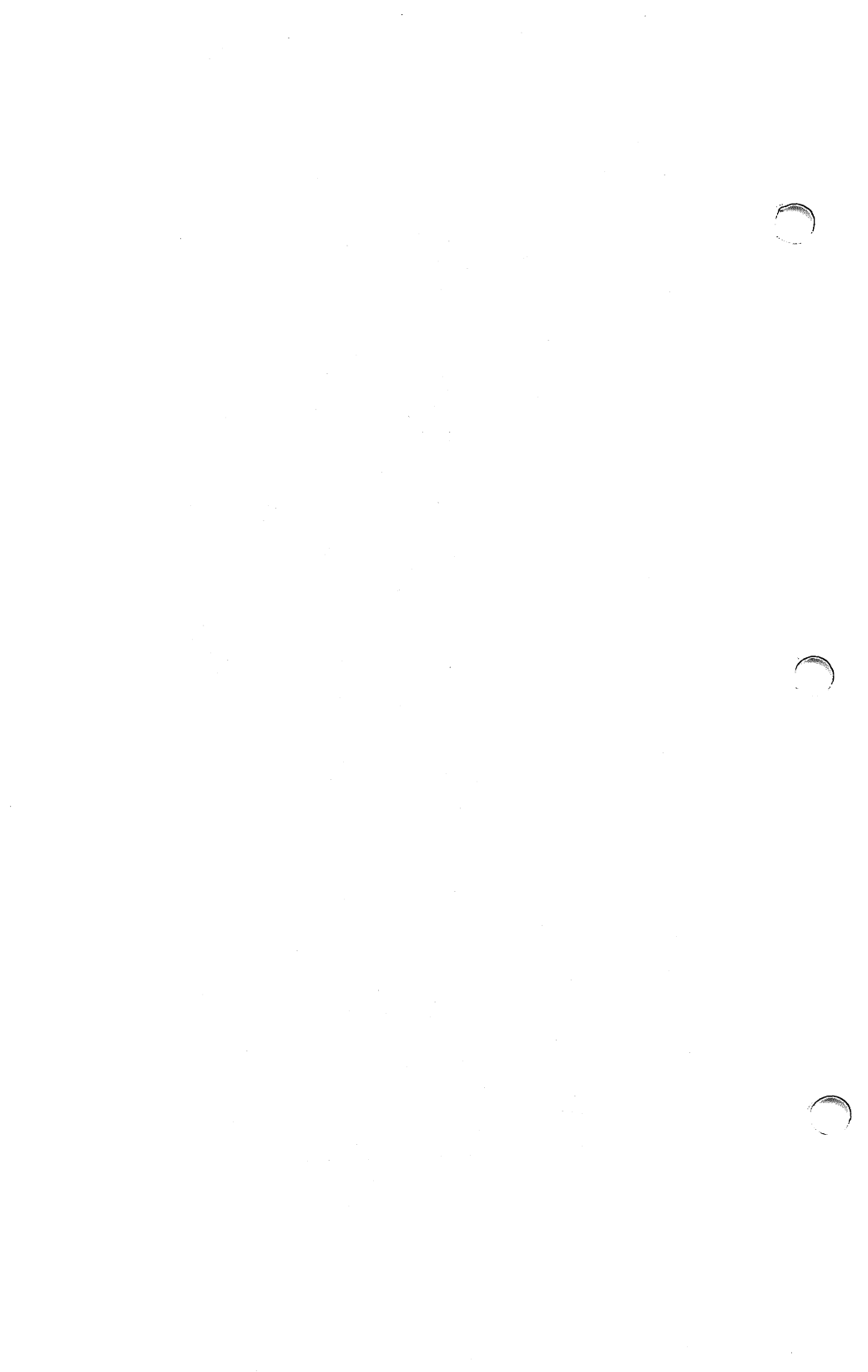
APPLICATION : Uses the Database "MESSAGE"
The File Name for this App is "MESSAGE.HAP"
MODULE : Uses HyperLINK Module "DATABASE"

To : Uses the Data Field "MTO"
Date : Uses the Data Field "MDATE"
Time : Uses the Data Field "MTIME"
From : Uses the Data Field "MFROM"
Of : Uses the Data Field "MOF"
Phone : Uses the Data Field "MPHONE"

Telephoned : Uses the Button Named "MB1"
Returned Call : Uses the Button Named "MB2"
Wants Visit : Uses the Button Named "MB3"
Wants to Come : Uses the Button Named "MB4"
Please Call : Uses the Button Named "MB5"
Will Call Again : Uses the Button Named "MB6"

Received By : Uses the Data Field "MRECV"
Message : Uses the Data Fields "MMSG1"
"MMSG2" & MMSG3"

Since the MESSAGE Application uses the HyperLINK DataBase Module, when a message window is open you have the added options provided by clicking on the right mouse button. This gives you your standard set of database commands such as FIND, NEXT, etc which are used to speed your way through the database records. It is also here where you can SAVE any changes made to this database, DELETE any records, set which index you wish to SORT by, send records to a CLIPBOARD, PRINT records out, or add a NEW record to the message database.



The ADDRESS BOOK Application...

As the title of this application suggests, this is used for keeping track of addresses, phone numbers, contact name at a company you deal with etc.

Like the Message Application, you can sort by any of the fields as well as search through the database for any needed records.

The screenshot shows a window titled "ADDRESS" with a form containing the following fields:

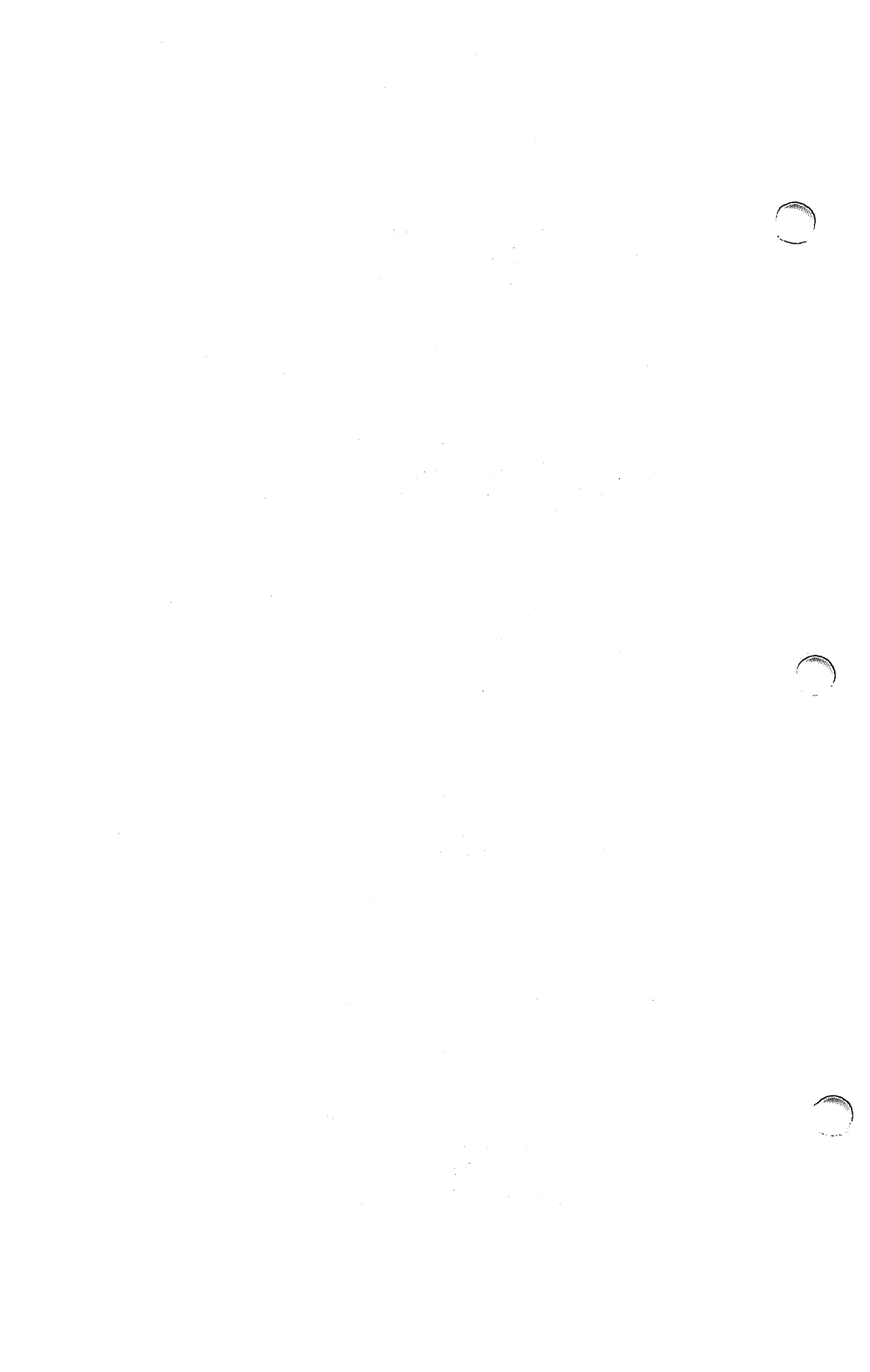
- LAST NAME : _____
- FIRST NAME : _____
- COMPANY : _____
- ADDRESS : _____
- CITY : _____ STATE : _____
- COUNTRY : _____ POSTCODE : _____
- PHONE : _____
- FAX : _____
- NOTES : _____

The Data Field names in the ADDRESS BOOK Application :

APPLICATION : Uses the Database "ADDRESS"
The file name of this app is "ADDRESS.HAP"

MODULE : Uses HyperLINK Module "DATABASE"

Tab : Uses the Data Field "TAB"
Last Name : Uses the Data Field "LNAME"
First Name : Uses the Data Field "FNAME"
Company : Uses the Data Field "COMP"
Address : Uses the Data Field "ADR"
City : Uses the Data Field "CITY"



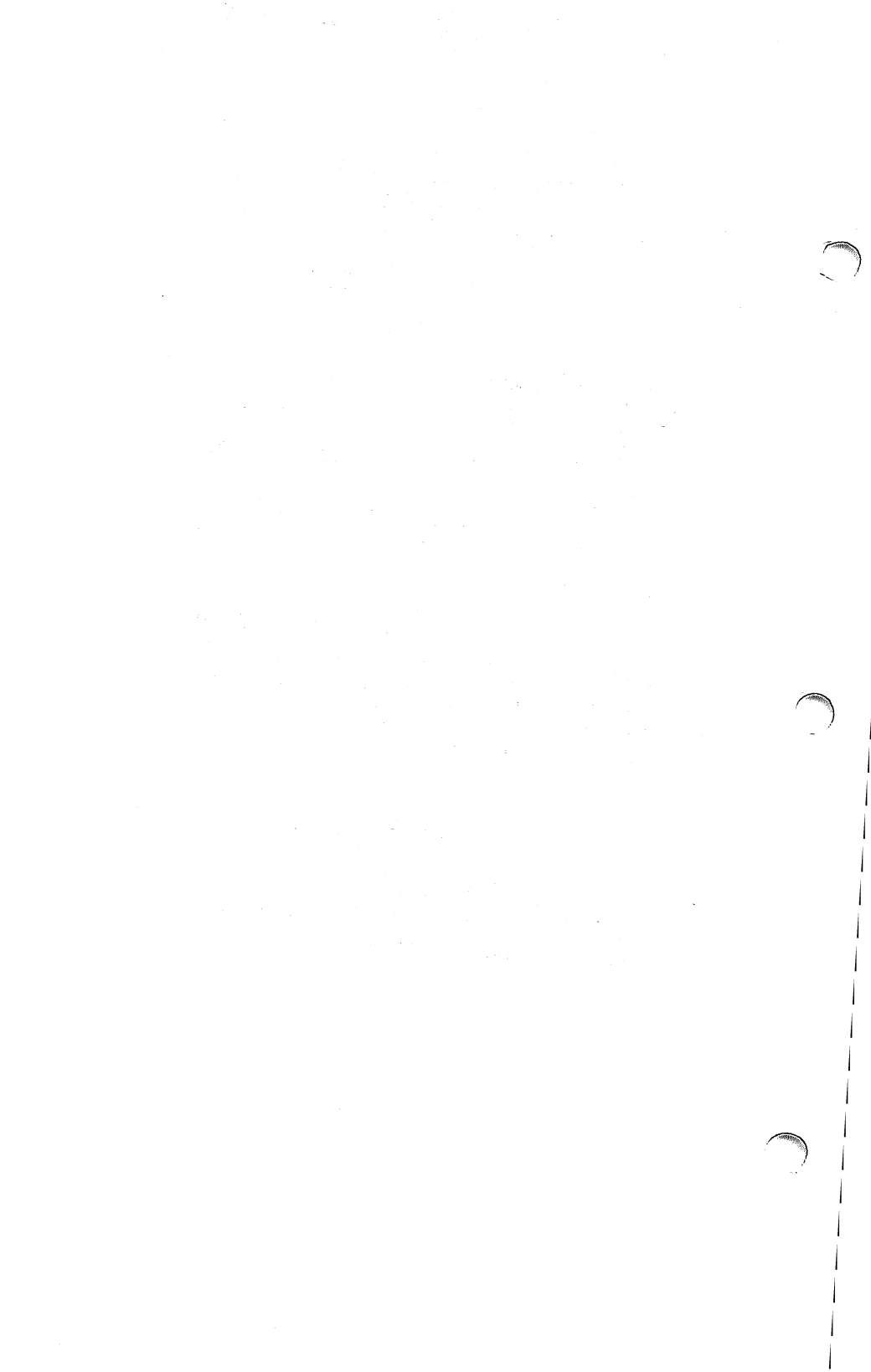
State / Prov : Uses the Data Field "STPROV"
Postal / Zip : Uses the Data Field "POST"
Country : Uses the Data Field "COUNTRY"
Phone : Uses Data Fields "PHONE" & "PHONE2"
FAX : Uses the Data Field "FAX"
Notes : Uses the Data Field "NOTES"

The RECOMMENDED SORTS in the Address Book Application are :

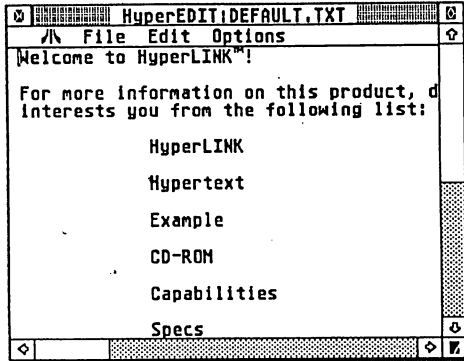
ADDRESS : Tab and Company - Sorted by
Tab 1st then Company
ADD_LNAM : Lastname and then Firstname
ADD_COMP : Company and then Lastname
ADD_POST : Postal or Zip Code

The reason for these specific sorts is that the Telephone Message Application sends a LINK TO the Address Book. When a user calls up the link from within a Message, HyperLINK has to have a matching index to the one requested.

Also note in the Address Book Application that the right mouse button once again calls up the search, save, etc menu selector. Finally of note with HyperLINK's Address Book is that the two Arrow Buttons will advance you to the next or to the previous record.

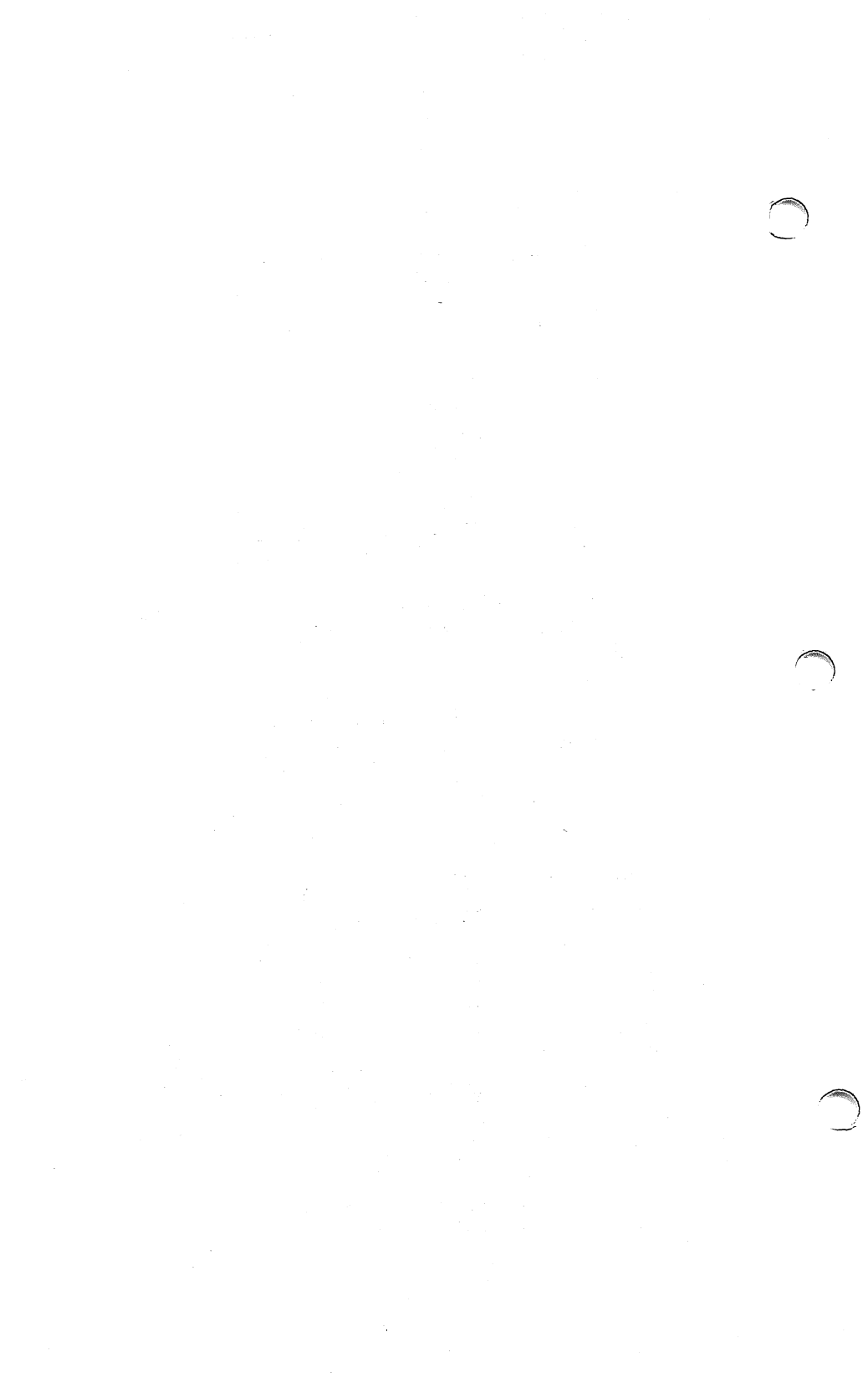


HYPERLINK TEXT EDITOR



For those of you in the ST world that are familiar with the Tele-Communications software "Stalker", HyperLINK's Text Editor should look familiar. It is a modified version of STENO which was the desk accessory text editor that comes with Stalker. We wish at this time to thank Eric Rosenquist for his help and his support in this part of HyperLINK. While the editor is in a working condition, we look forward to many enhancements to this area of HyperLINK.

While the HyperLINK Text Editor does at this time show up in your desk accessory menu, it cannot be accessed as a regular accessory. Instead it can only be loaded as a HyperLINK Application, the same way any other application loads in. However the HyperLINK Text Editor is an exception to many of the other rules of HyperLINK Applications. Some of the exceptions are that you can only have one Text Editor Window open at a time, the window also can not be minimized, and the button that usually sizes downs the windows makes the Text Editor full screen size instead. One of the major upgrades we are looking forward to with regard to the HyperLINK Text Editor, is making it into a true HyperLINK



Application. When this is accomplished it will add the ability to have more than one text window open at a time, thus allowing greater ease for many editing commands.

The HyperLINK Text Editor can operate on different levels. You can use it as a simple word processing application, or you can use it as one of the key ingredients in the LINKING portion of HyperLINK. The linking portion of the Text Editor is referred to as HYPERTEXT.

As a word processor, the HyperLINK Editor has all the standard features of a low-end text editor such as "Cut & Paste", "Tabs", "Find", "Find & Replace" etc. So for many uses this simple text editor can fill many everyday needs.

To use the HYPERTEXT feature in the editor, double click on any word you want to reference. For instance, if you had a sentence such as... "Beethoven was a great composer." By double clicking on "Beethoven", HyperLINK would search the rest of its cross reference or "Link" files and see if there are any other references to Beethoven. And if you have created LINKS to any other HyperLINK Modules, then they too would be searched for reference to Beethoven. Could be that if the sound module is tied in to the editor, HyperLINK could find and begin playing Beethoven's Fifth Symphony.

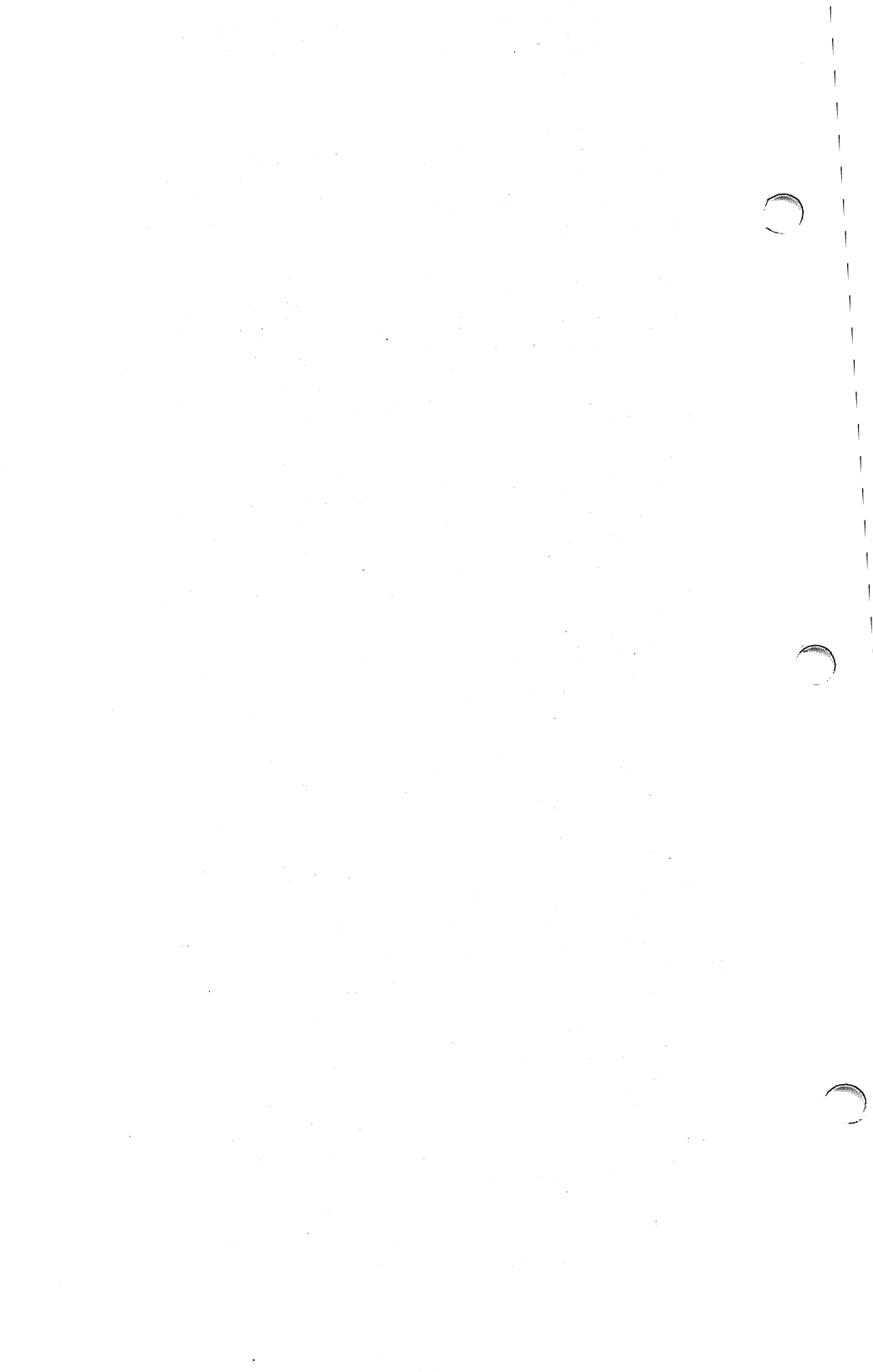
Editor keys and shortcuts:

CTRL-(Mouse Click) : Another way to call a link, equivalent to a Double-Click of the left button.

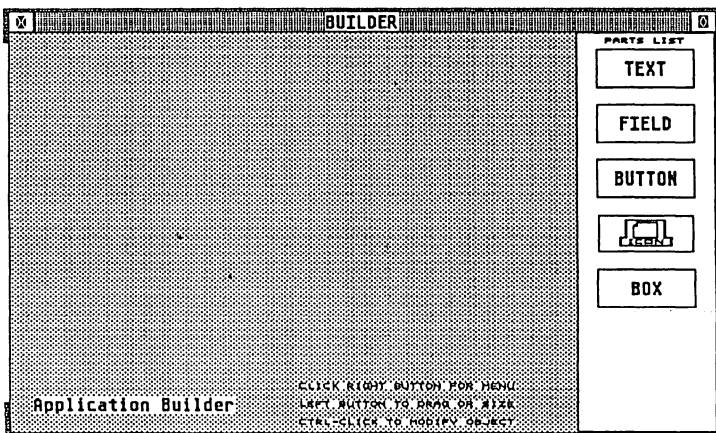
ALT-Double Click : Calls up the TEXTLINK application, allowing you to create new links for the current text file, and for the word you selected.

UNDO : Return to previous text file.

ALT-UNDO : Call up the TEXTLINK HAP without first selecting a word.



THE BUILDER...

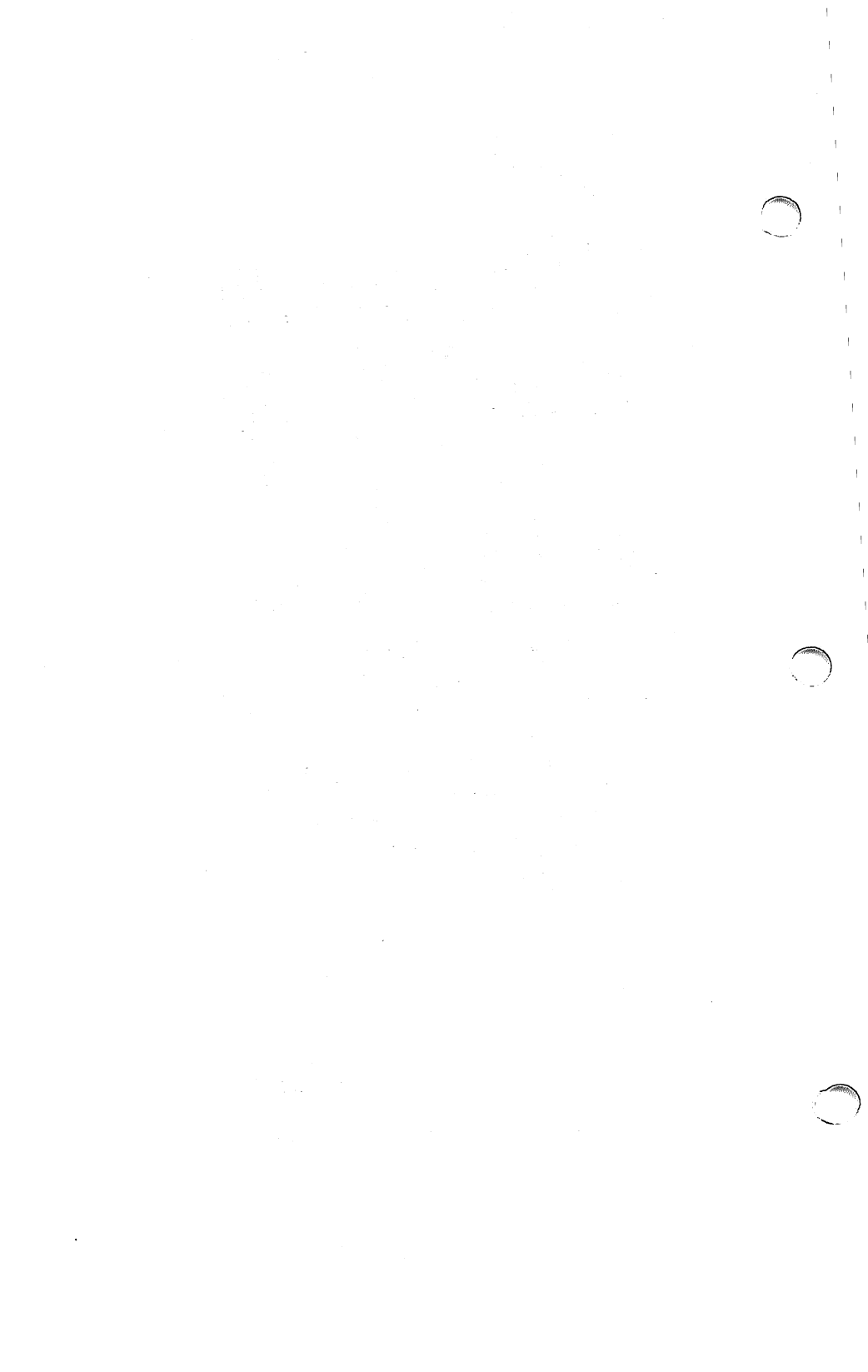


The Builder is the part of HyperLINK that creates all the Forms, and more importantly the LINKS that the applications can use. The Builder is made up essentially of 1) a PARTS LIST screen, where you are actually setting up or building the form itself of an application. 2) the ITEM MODIFYING screen, where you modify the characteristics of every item in an application, and 3) the APPLICATION screen, where you set up the module info for any application.

Let us look at these screens in the order you would use them when creating a new HyperLINK Application :

APPLICATION SCREEN :The Application screen takes the general info on an application you are wanting to build. What you should enter for every application is -

FORM NAME - Name of Application
MAKE - Publisher's Code for creator of form
MODULE NAME - Which HyperLINK module is used



GEN USE FILENAME - Optional extra Data File
ID # - Value GREATER THAN 100 refers to the form
VERSION - Used to indicate the version # of an app

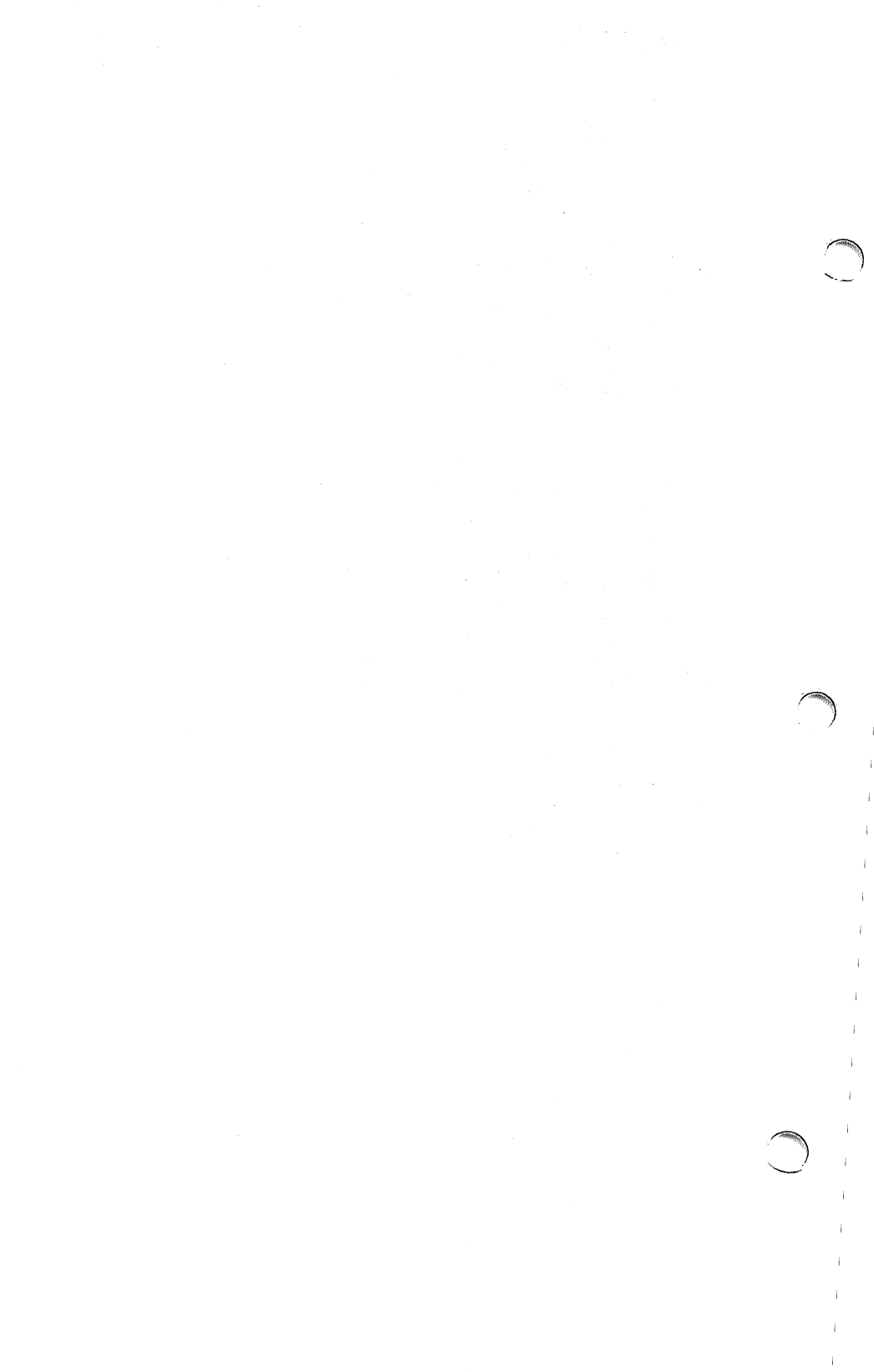
DATABASE TYPE - HyperLINK DB or Other ?
DATABASE NAME - Data File to be used by application
DEF INDEX - Default Index for Application to use
A-LOAD - Whether to auto load first data record

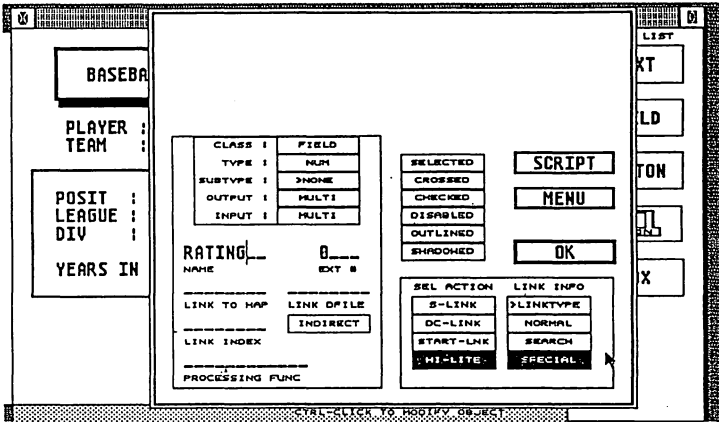
FIRST EDIT FIELD - Name of field to begin edit on
LOG IN ICON - Whether app icon should show or not

WINDOW FEATURES - Use these to set whether an app's window contains the application name, whether it can be closed, can the window be minimized, and can you move the window
APP ICON :What that application's ICON looks like at the moment. You can also load in another existing icon to take its place.

Once you have laid the ground work for your new application, the next step is in the design...

PARTS SCREEN:This is where you place the building blocks for your application. Essentially here you design the look of your application by placing TEXT, FIELDS, BUTTONS, BOXES, and ICONS.





Once the design of an application has been finalized, the third step in the use of the HyperLINK Builder is the individual ITEM Screen.

ITEM SCREEN :The Item Screen assigns many and varied characteristics to every item in the form or application.

BORDER - sizes the borders around items

FILL - sets the fill pattern for each item

ITEM NAME - Name of each individual field, button, etc

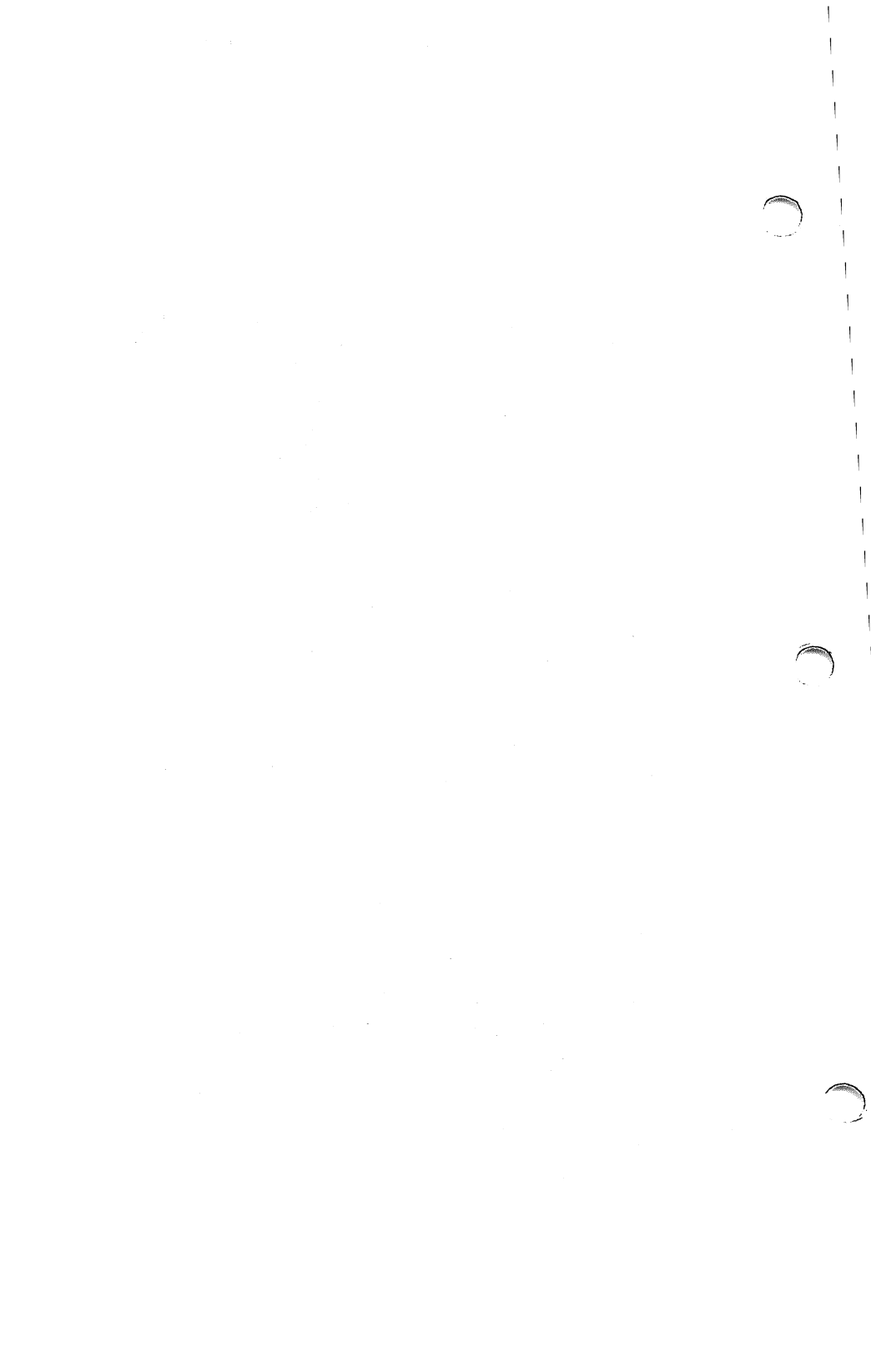
ITEM INFO - For a Database field the Class must be "DBFIELD" and the second class is "DFNAME" and this is used when linking to another file or form.

Other types may be added or created. The rest of the settings under Item Info will be set to defaults by the builder in HyperLINK.

EXT # - Just a form specific number that can be defined by each HyperLINK Module.

LINK TO...HAP - This sets which additional Applications an item or field can link to.

LINK DFILE - This sets which file, in the above named application that HyperLINK will Link to.



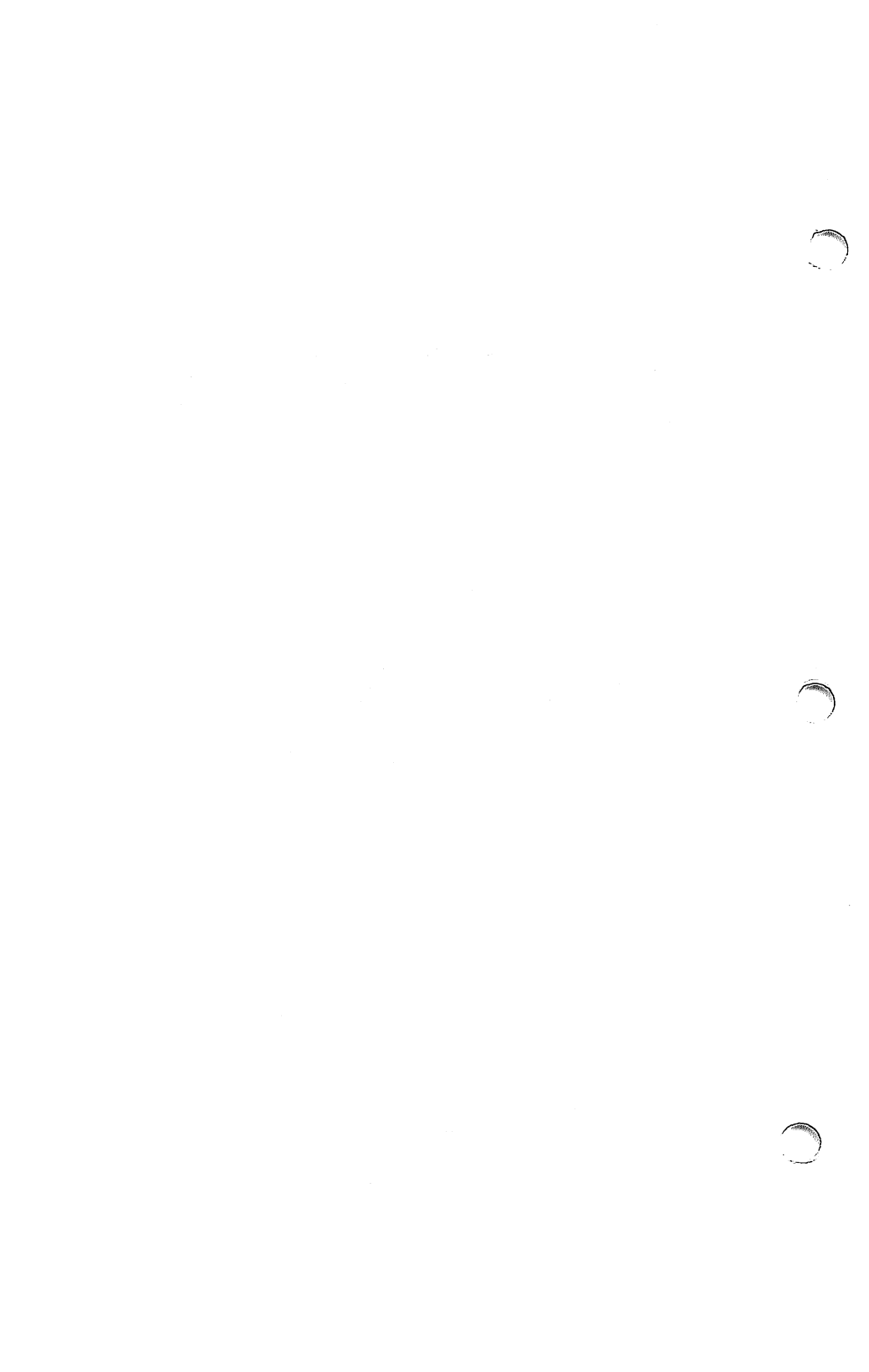
LINK INDEX - Sets which sort will be used by HyperLINK when it is doing Link searches.

PROCESSING FUNC - This is for future use in adding specialized functions to an object or field.

SELECTED/CROSSED/CHECKED/DISABLED/OUTLINED/SHADOWED - These are similar to the items in a programming resource construction set. They select how an item or field is to be displayed.

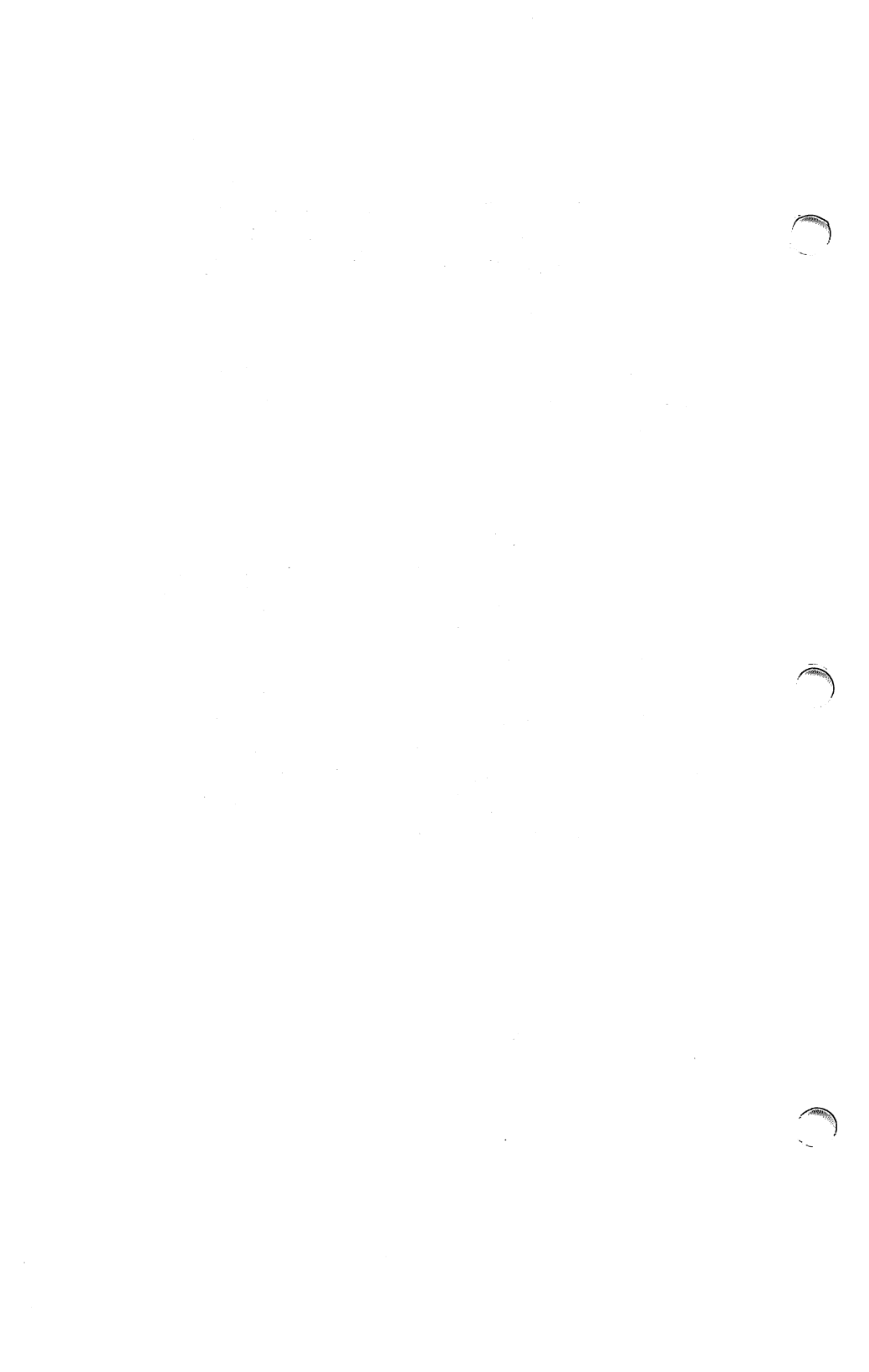
SCRIPT - Calls up a window to let you enter a command script for any given item or field. More info to follow! While scripting is not available in the current version of HyperLINK, it will be available shortly.

MENU - Allows the users a few extra utilities in the creation of your forms : FRONT, BACK, SORT, EDIT, DELETE, and MULTI. Front and Back control the positioning of an entire parent item and all its children, placing for instance a box either on top of or underneath an IMG file. Sort deals only with the children of the parent box selected. Regardless of what order a field may have been entered on a form, Sort will sort the children so that they are editable in the same order as they graphically appear on screen. Edit allows you to go in and changed the contents of an object. Del deletes the currently selected object, and Multi sets all children objects of any type to be multi-selected. That is, if you have three buttons within a parent object, by making them multi-selected on your form, in the application when one of the children buttons is selected, it will de-select any other children button. The first multi button in the graphic layout of your form MUST be given a name for database use.



SELECT ACTION : (S-LINK, DC-LINK, START-LNK, & HI-LITE) These options allow for the linking process to begin at different times. To explain further : S-Link is a single mouse click link so that as soon as an item of field is highlighted, HyperLINK will check for any links. DC-Link is similar, but for HyperLINK to search for links, the item or field must be selected by a mouse double-click. Start-Lnk automatically searches for any link as soon as the form with this given action is called up. Lastly Hi-Lite simply tells any object in a form to Invert its colour when it is selected.

LINK INFO : (LINKTYPE, NORMAL, SEARCH & SPECIAL) Linktype lets you set whether your field or button etc is linking to a text file, graphic file (GEM or IMG), sound file, another data file, speech file, or to a command from the Script Language (look for more info on the Script Language in future updates). The last possibility under Linktype is Nul. Nul can mean all of the above or none of the above or any combination of the above. Sound confusing? Simply put, select NUL for Linktype if none of the other entries fit properly or exclusively.



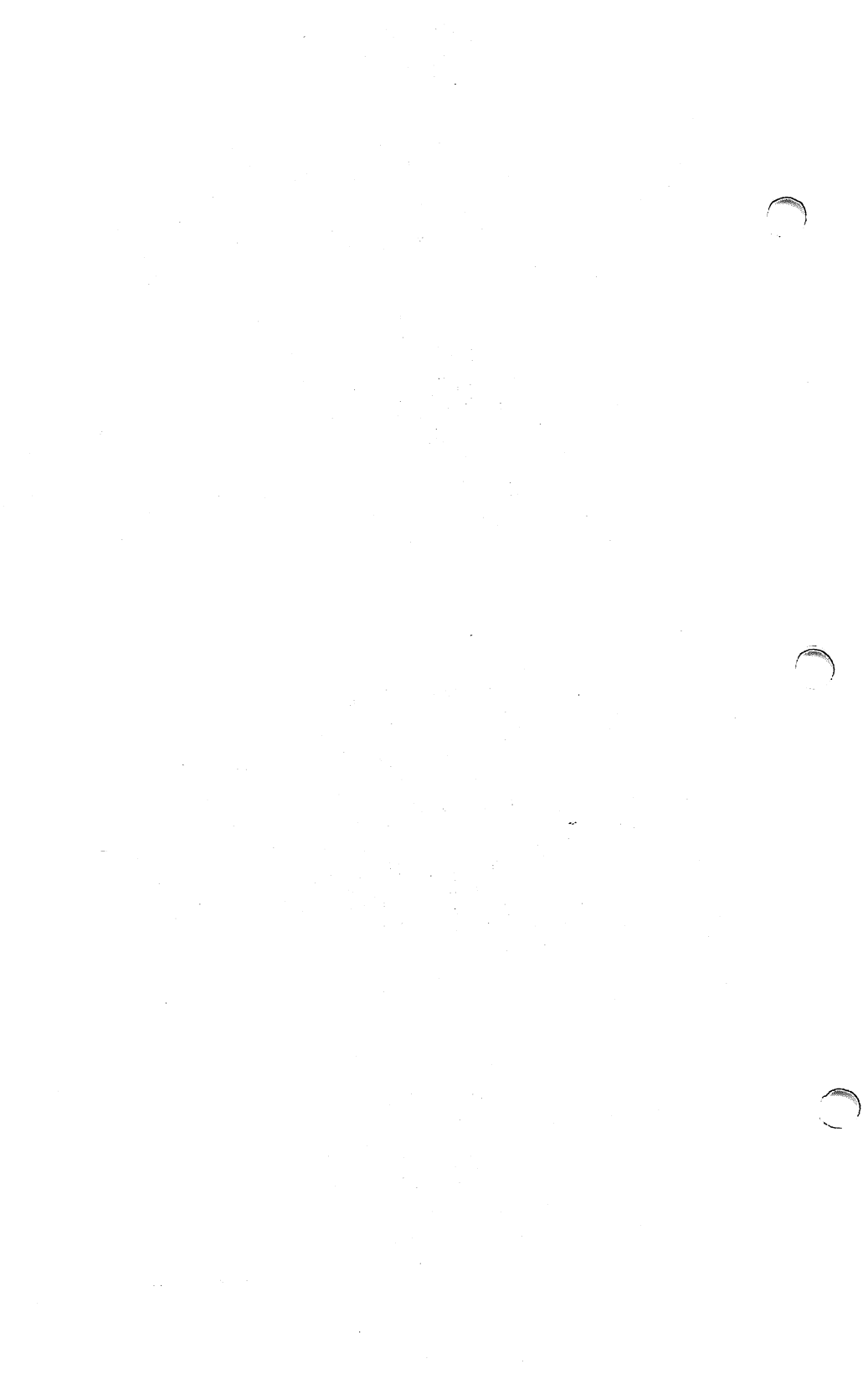
NORMAL/SEARCH/SPECIAL are the settings for how a link is worked. Normal is the default used by HyperLINK. The differences here are as follows : Normal is a link to a file and call up that file. Search is a link to a file, and then a search through that file looking for a corresponding record, and then opening the app at that record. Special is used differently in each receiving application. The only applications that presently support Special are the ones using the Database Module. "Special" is more deeply explained in the LINK COOKBOOK.

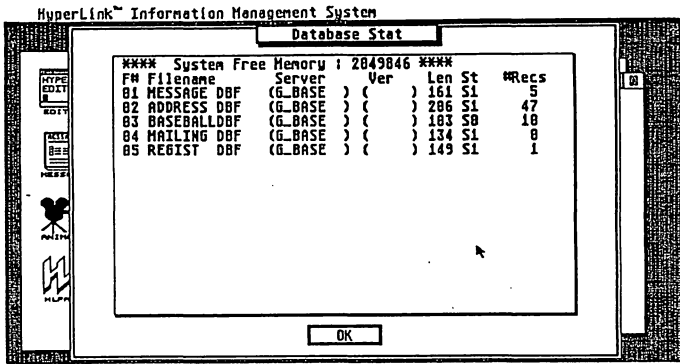
READ THE LINK COOKBOOK!!! It goes into many more specifics on the type of links available, and how to accomplish them.

MENUS

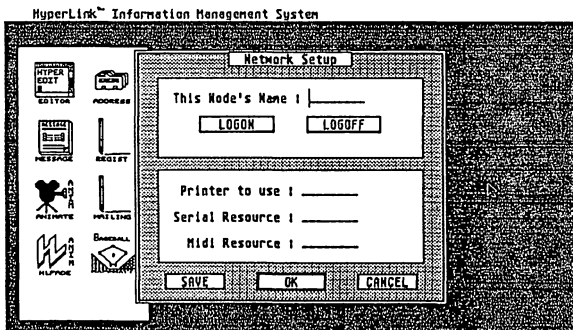
The menu selections within HyperLINK are fairly self explanatory. Under the FILE Menu, procedures such as Loading Modules and Applications, Quitting the Program are available through the menus. Also under the FILE Menu are "Default Apps" and "Default Modules". These entries load in their respective text files which list the default for HyperLINK to load in upon startup. "Run Program" lets you run another ST / TT program while still keeping a tie to HyperLINK. (This is not multi-tasking, merely task switching.) Thus when you quit this other program, you will return to the point within HyperLINK at which you left.

Under the SYSTEM Menu, you can view some useful information pertaining to various aspects of HyperLINK. Module, Application, Task, and Database Status are all available. Mostly these info boxes are just for interest, but they also provide quick summary of what is currently going on in HyperLINK as you can see by the following example.



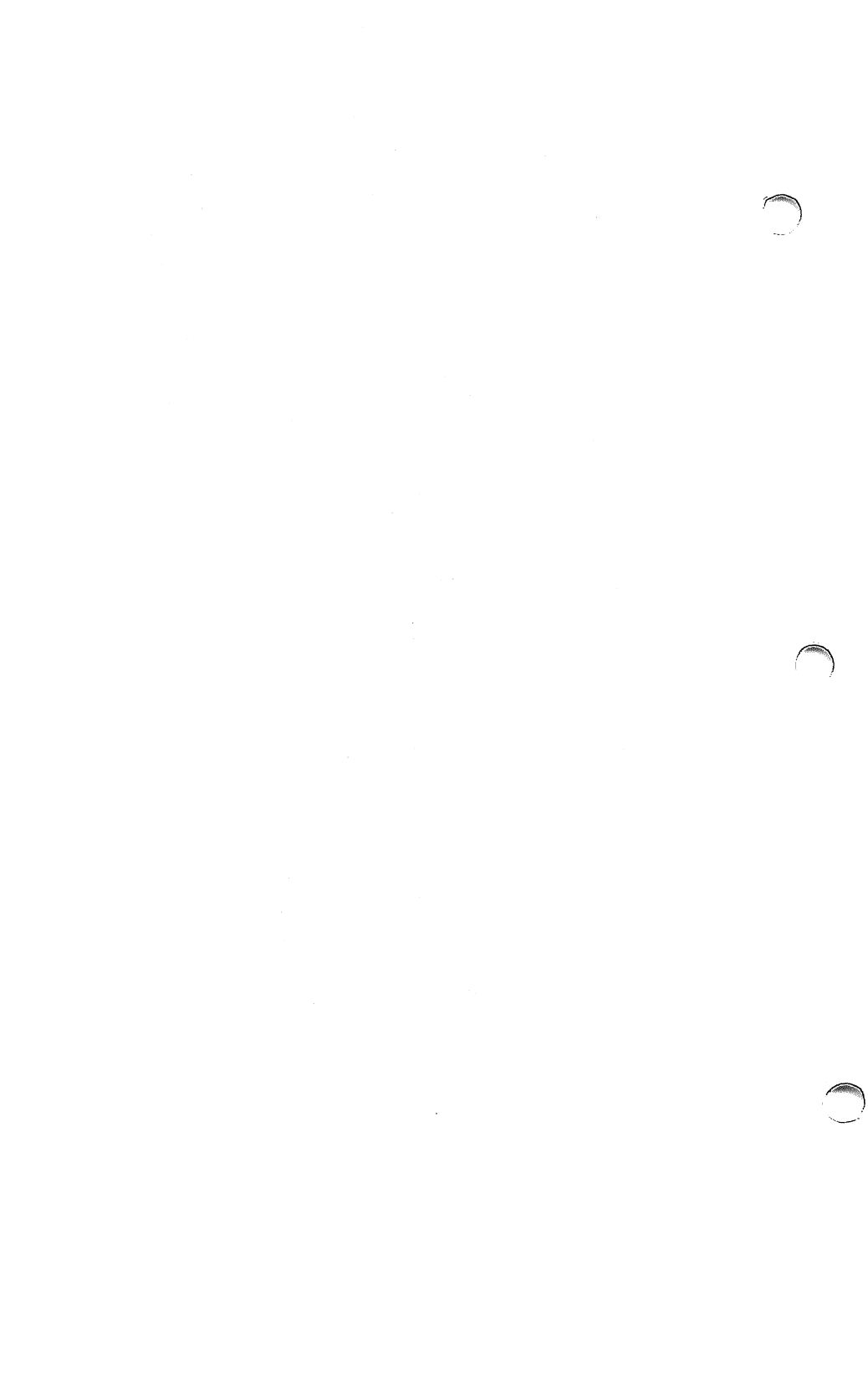


Also under the SYSTEM Menu is the "Preferences" selection which like the "Default Apps" loads in a text file containing a list of your HyperLINK Default Settings. Finally of note in this menu area is "Special". This is for use by modules to tie into. For instance, a module may be developed in future that once loaded in, takes advantage of a menu item, replacing the "Special" menu title with one of its own as well as with a corresponding function.



The NETWORK Menu is only relevant to those who have the hardware and/or software of an ST Network. HyperLINK will support GNET, LANTECH, and will also incorporate any other networks that become available for the ST or TT. Selecting "Network Setup" calls up the basic setup and log-on screen for HyperLINK.

The final menu is that of UTILITY. Under this item we find



"App Builder", which loads the Application Builder up (Sort of a Quick Load option for the Builder) and "Expansion" which essentially is the same as "Special". ie. another menu pot open for use by future modules.

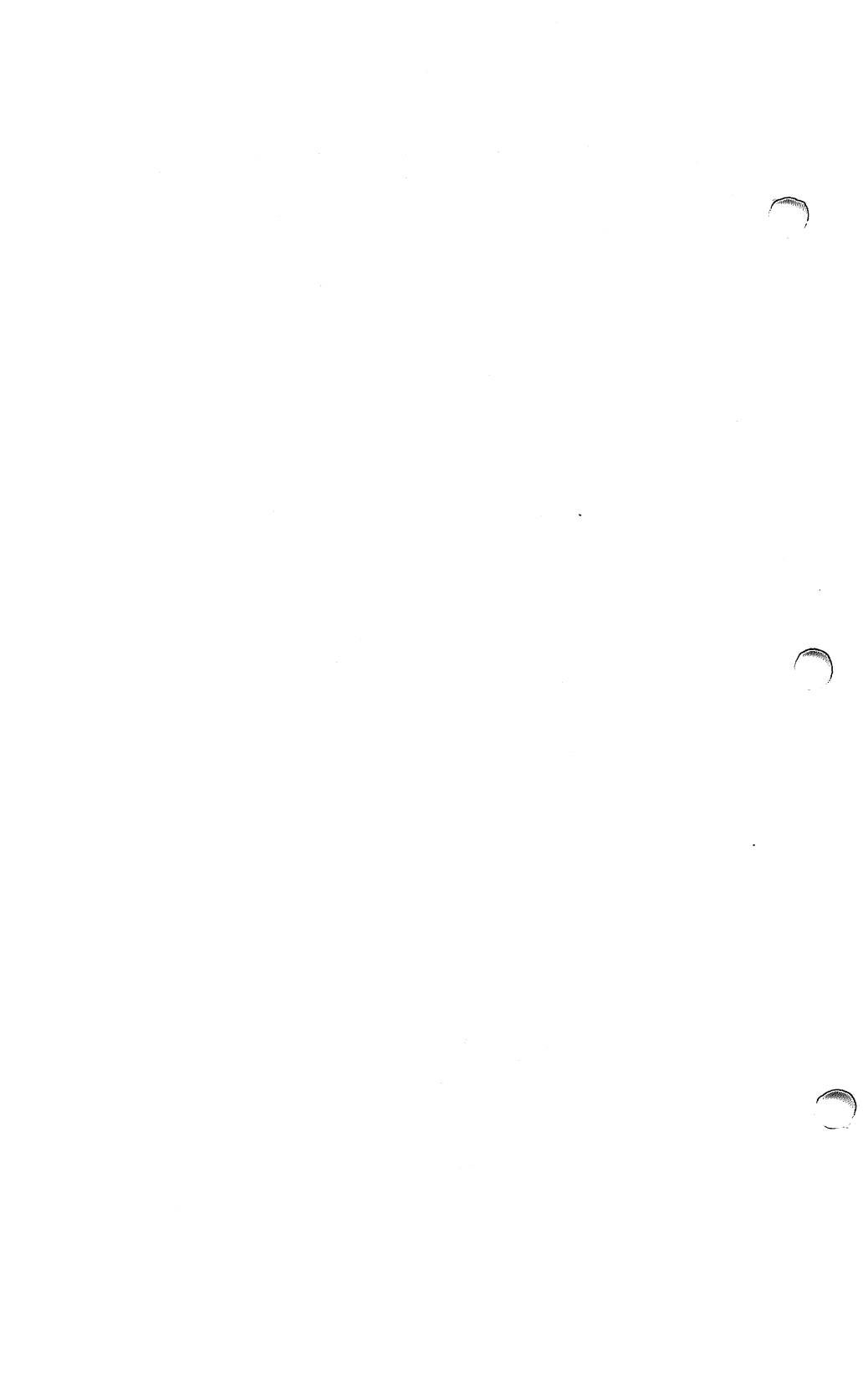
Obviously this is just a preliminary manual for HyperLINK. It is our intent that this brief covering of HyperLINK will give you a basis from which to begin use of this powerful software product.

It is our intent also to keep a very regular upgrade schedule with HyperLINK, adding more modules, applications, and adding features to the software that come as feedback from our registered users.

Please let us know what you think of HyperLINK and how in your mind we can improve the product to make it even more useful to the average computer enthusiast.

Thank you for using HyperLINK, and we hope that it helps make your computing life a little easier!





CREATING A HYPERLINK APPLICATION

The Step-By-Step Guide

Now that you have a basic understanding of what HyperLINK is and what it can do, the best way to learn how to put together a HyperLINK Application (HAP) is to follow along on our step by step assembly of a demo HAP.

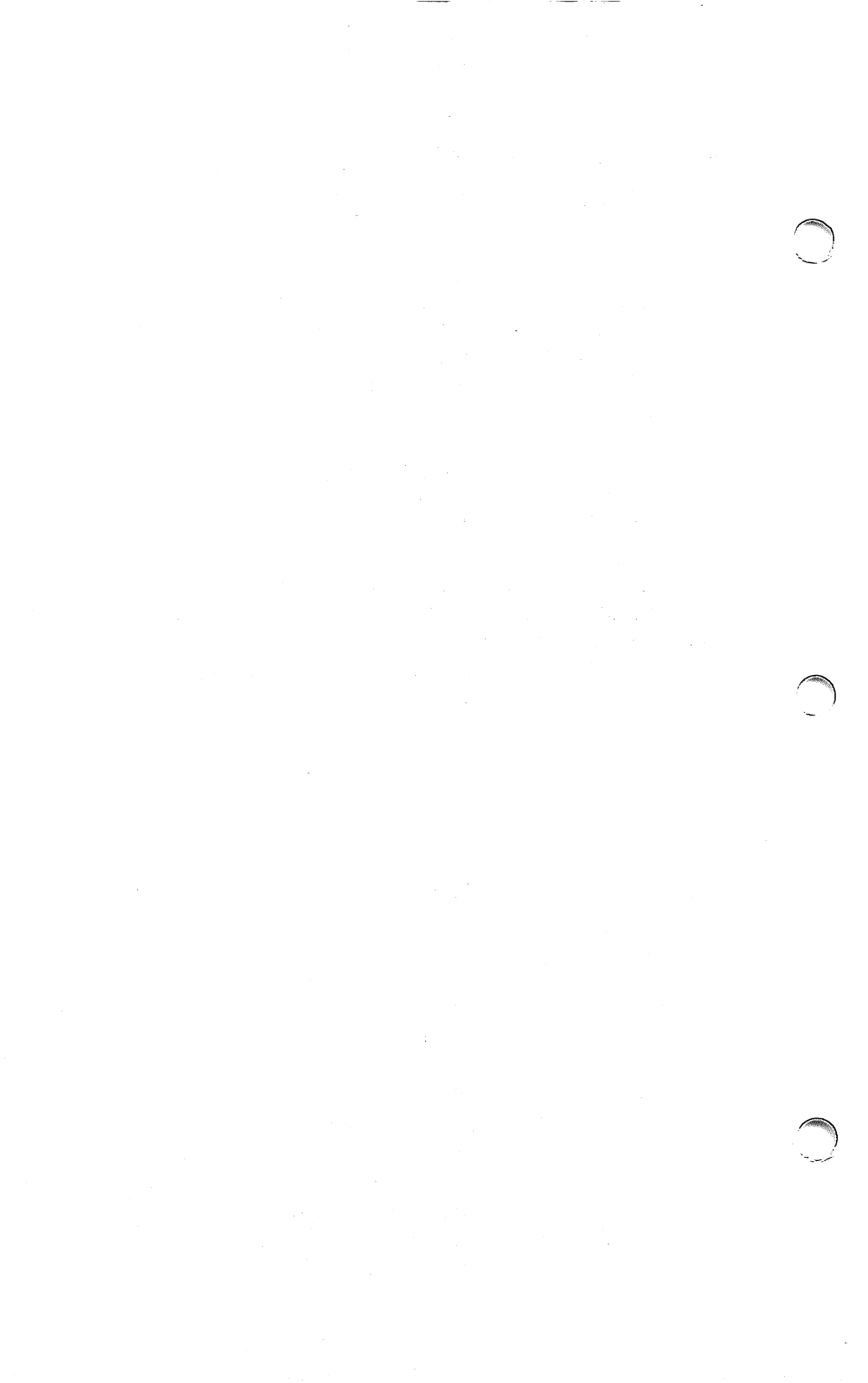
For our purposes, (mostly because one of the principals in our company collects them) we will consider a HAP for a Baseball Card Collector.

The first thing you should do when creating a HAP is figure out what exactly you want to keep track of. For our purposes we will keep track of : Player / Team / Card No. / League / Division / Player Rating / and some elementary stats.

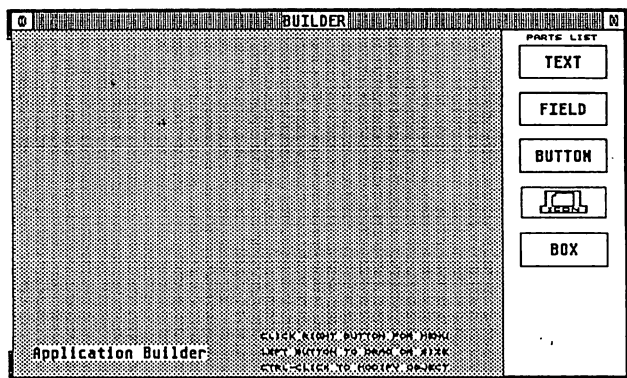
FIRST IDEAS OF APPLICATION FIELDS :

Field	Field Name	Type	Length
Player	Player	Text	40
Team	Team	Text	40
League	League	Text	8
Division	Division	Text	4
Card #	Card	Number	4
Years	Years	Number	2

As you proceed in building a HAP, more ideas will occur to you as you are designing it. With HyperLINK there is no need to worry about this. There are no initial requirements for allocating space. Modify an application to your heart's content! Because of this, only the most basic of concepts need be present when you start your building.



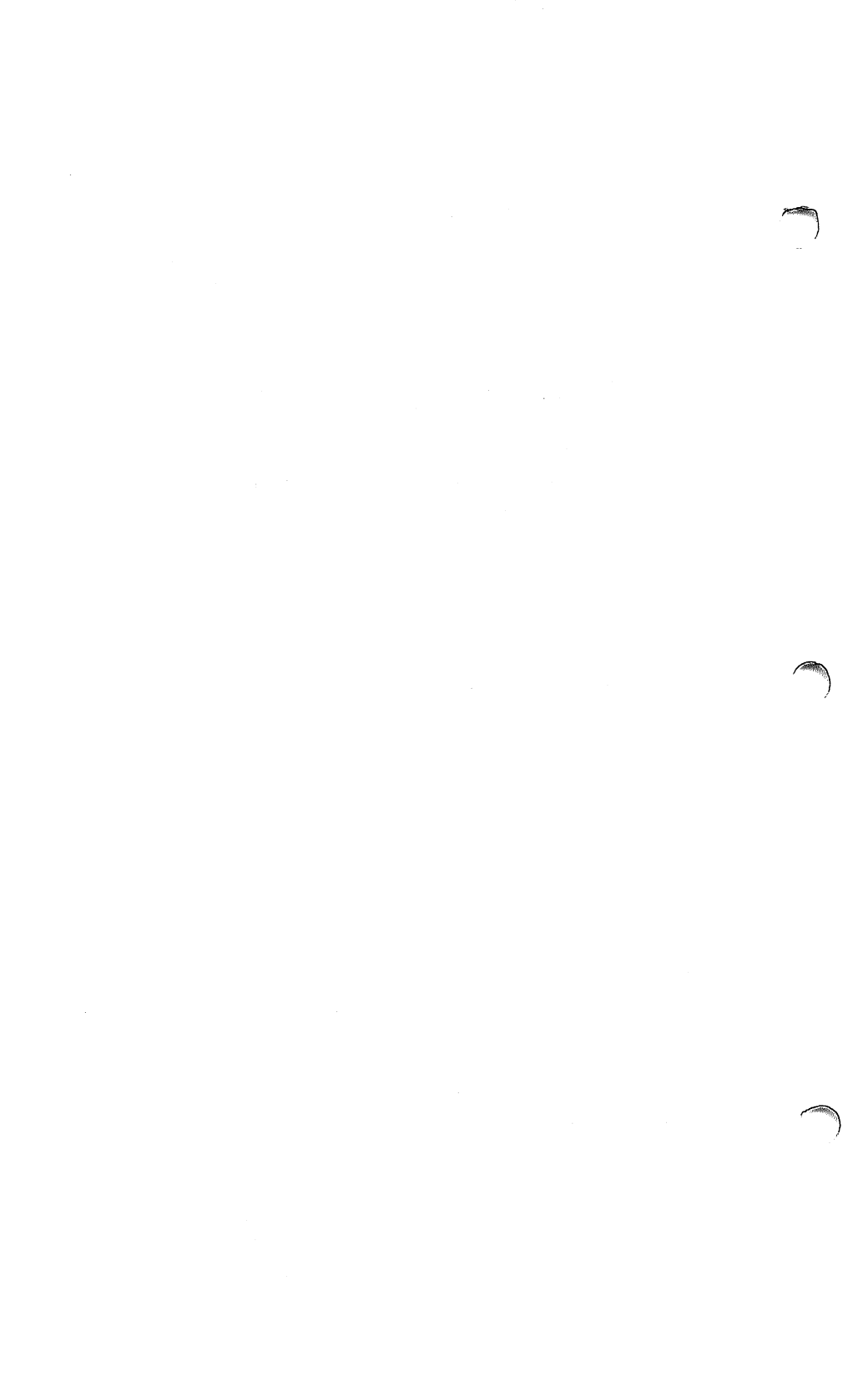
To begin creating a HyperLINK Application, start up the HyperLINK program as normal. If the Builder did not load in as a Default Application, load it in through the pull-down menu. Next, open the Builder Application. What should now appear on screen is the empty Builder Construction Form...



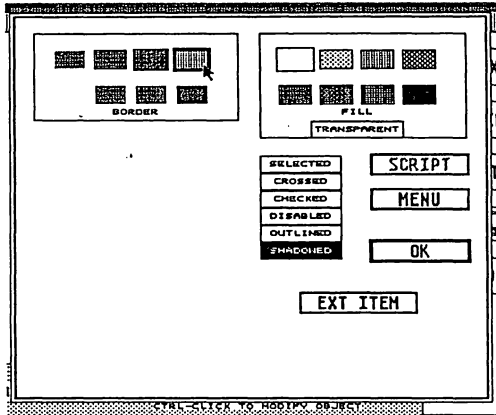
To start editing a new HAP, click on the right hand mouse button and a modified version of the database menu will appear, specifically for use with the builder. Select NEW from this menu and a blank area should then appear in the builder. Size this application screen according to what you think you will need. Don't worry, if you need larger than you first select, you can still drag the box bigger. (At least to the maximum size of the builder screen.)

Now comes the time to first design the basic layout of your HAP, or how it will appear on screen when you call up your application.

Well, let's first create a Shadowed Box and put a title in it. To do this, drag a box to the Builder Work Space. To increase the size of the border on the box and to add the shadow effect, Double Click on the box

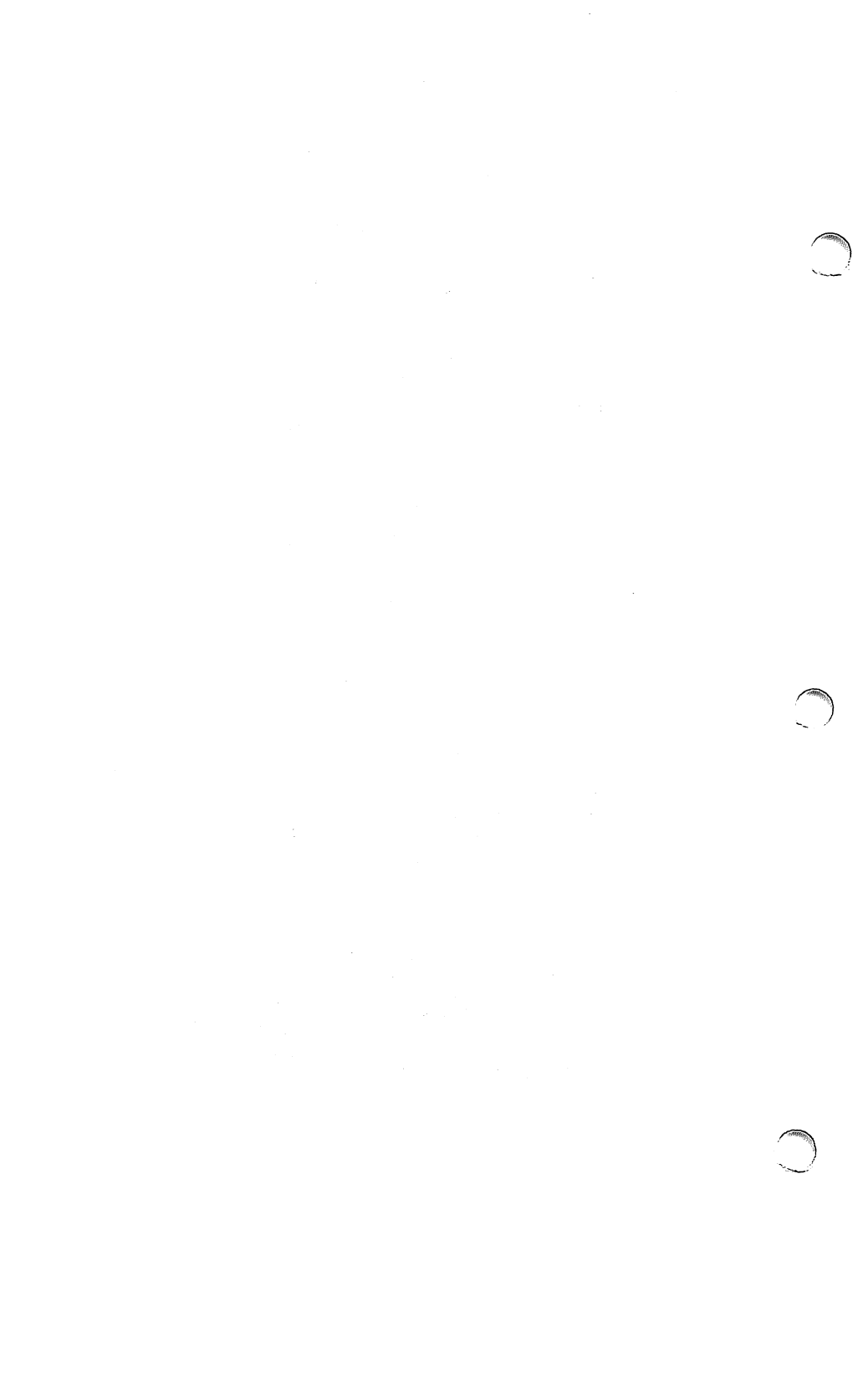


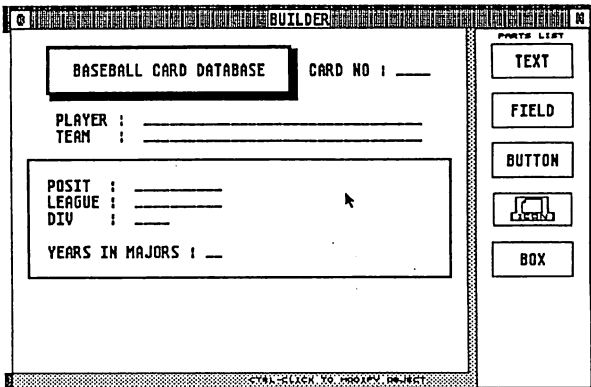
with the left mouse button. (Or just single click on the box while holding down the Control key.) This opens the object editing screen for all the objects in your HAP. The most important knowledge needed for HyperLINK is a full understanding of all that can be accomplished on the Object Editing Screen.



After this first Box is placed and shadowed, the next step is copying a text field on top of it, naming your application. Okay, now you have a title... All you need is the actual application below.

Next step is arranging your data fields on the application. Putting the fields on the HAP is really a two-step operation. First you put a Text entry illustrating what the field is going to be. Secondly you drag a Field onto the HAP and then tell that field what it is going to be. (ie. size, type, name of field etc)

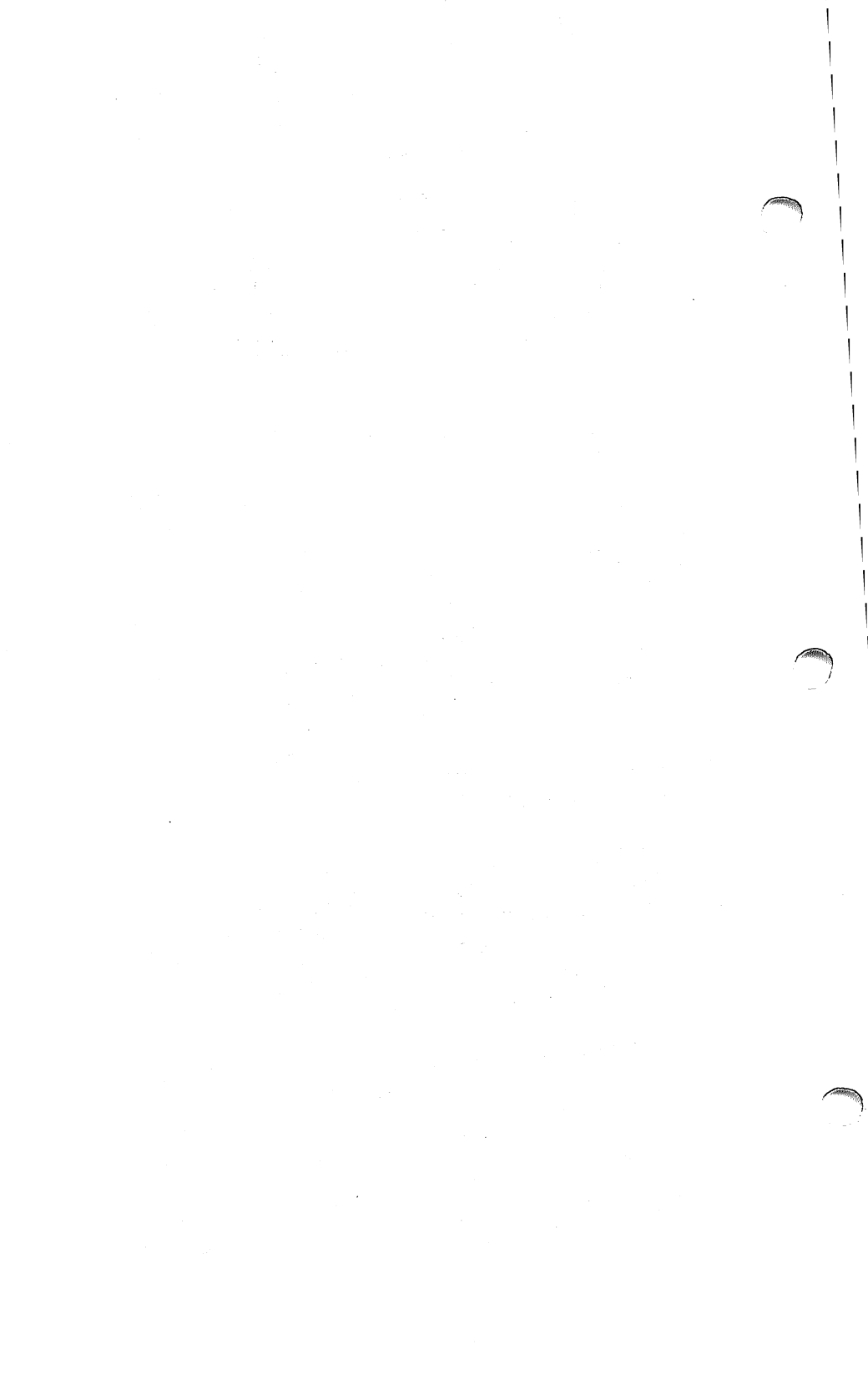


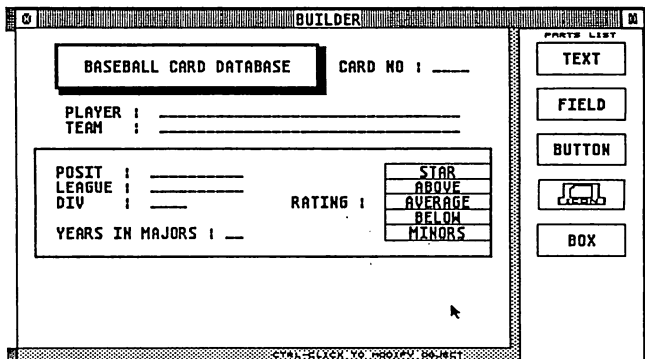


Okay, in the space of about five minutes (probably less) we have done most of the ground work for a basic HyperLINK Application. Now lets fancy it up somewhat. The next thing we decide to do in this Baseball Card Application is rate the players. One way of doing it may be adding Buttons to your HAP representing the different qualities of players you may get. For the purposes of our demo we created buttons for : Stars, Above (Average), Average, Below (Average), Minors (Minor Leaguer). The trick now is to select one, and only one, of these buttons per player. No problem!

HINT ! If you are going to have more than one series of buttons on an application though, you *MUST* make each series Children of different Parent Objects. An easy way of accomplishing this is to put a Box on the Application and then size it to hold all the buttons in a given series. This keeps Multi Buttons organized!

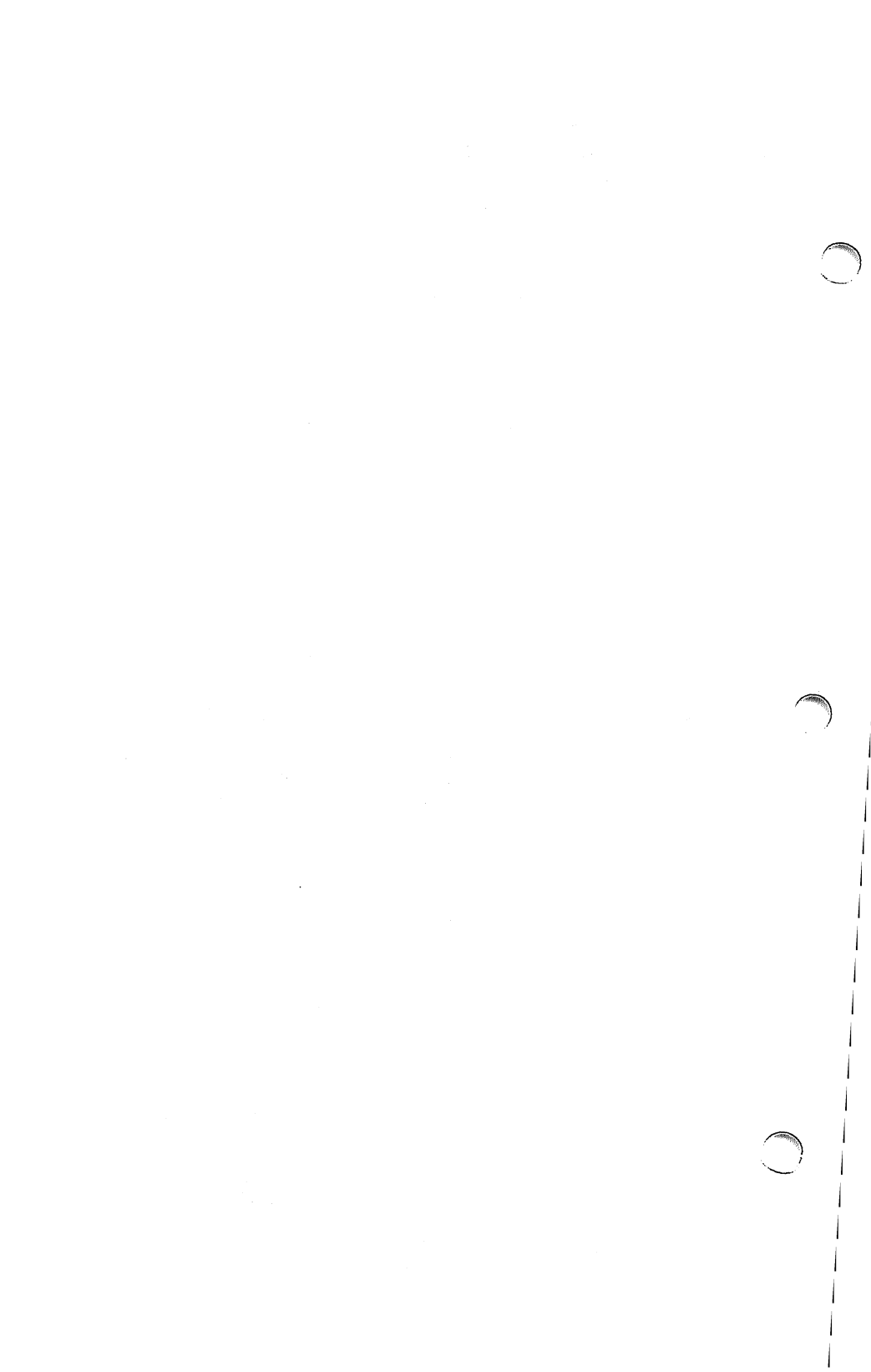
After you have put all your buttons on the application for the Rating, to make the buttons properly selectable, it is necessary to edit some of their characteristics. With Buttons though we don't have to go in and edit each button, we can simply edit the immediate parent of that series. In the case of our baseball card application, it is the box around the buttons.

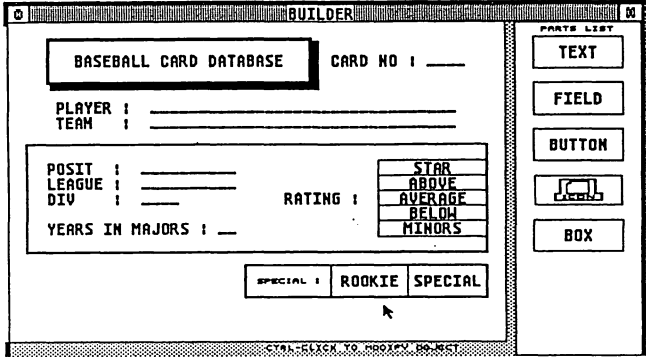




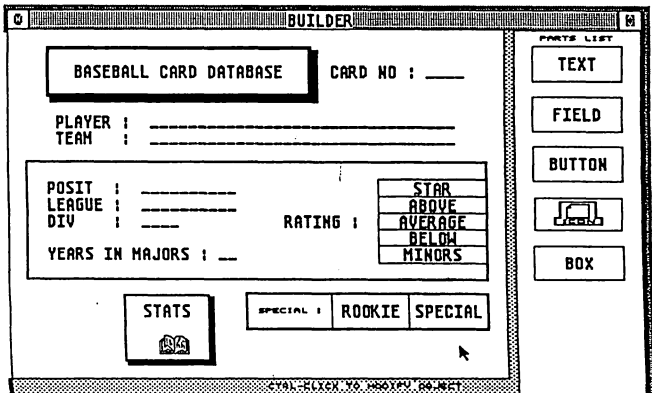
Double Click on the Box around the Buttons. Then select the Menu selection. From here select the MULTI option. This option tells HyperLINK that there are multiple buttons in this parent and that they are related to each other. (In programmer's terms, this sets the buttons up as Radio Buttons.) When you do this, HyperLINK will remind you that "First Buttons Must be given an Item Name". Like naming a field, you have to name the first button in the series when it is part of a Multi. To satisfy this requirement, simply double click on the Star button and give it a name. In our case we used "Star" as the button name as well.

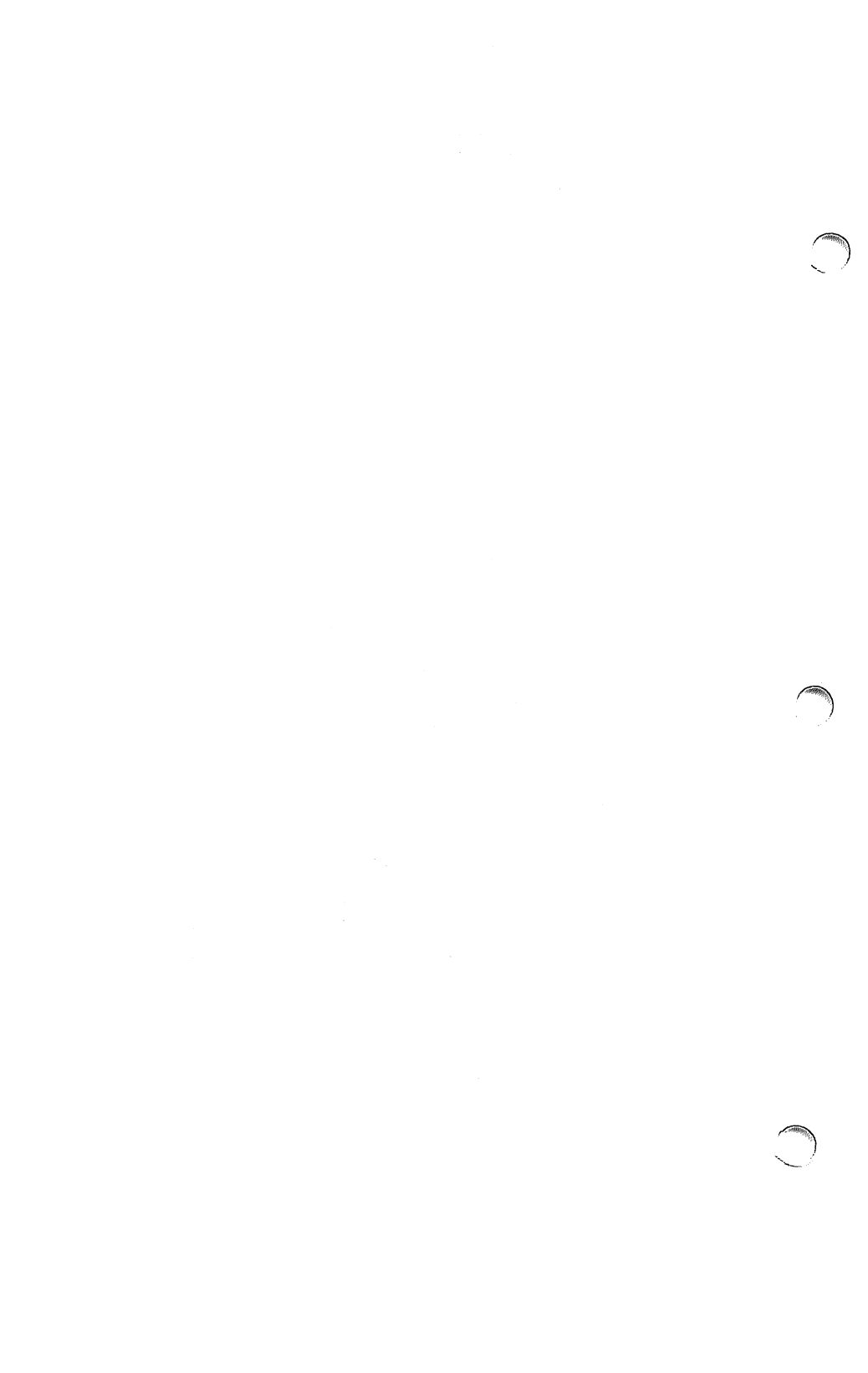
Another example of this "Multi" idea comes next : Suppose you get a Rookie Card. Everybody knows these are the ones that become valuable down the road! Why not set up another Multi button to mark a player card as either a Rookie card or maybe some other type of Special Card. As you can see by the example, this was done by creating first a parent box, then adding a text entry in smaller character size called "Special" and then inserting two buttons, one "Rookie", and one "Special". After they are placed in the parent box, double click on the parent box and again go into the menu selection and set this parent as "MULTI". After you have done this, don't forget you have to name the first button in the series!





What else can we do with our Baseball Card Application? Well, there sure aren't too many stats showing on the screen. We could link this database to a text file that may have more information on the player. Okay, first let's make a Box on the application. (This is so we can do some fancier things, than we could if it was just a button.) Go into the Object Editor Screen (Double Click on the box) and set up its appearance. (ie. Border, Shadow, etc) Now just another box might be a little boring on your application, so why not put an icon on top of the box to spiffy it up a little. To do this, simply drag the icon object to the box. Up will come an option for loading either an ICON or an IMG file.



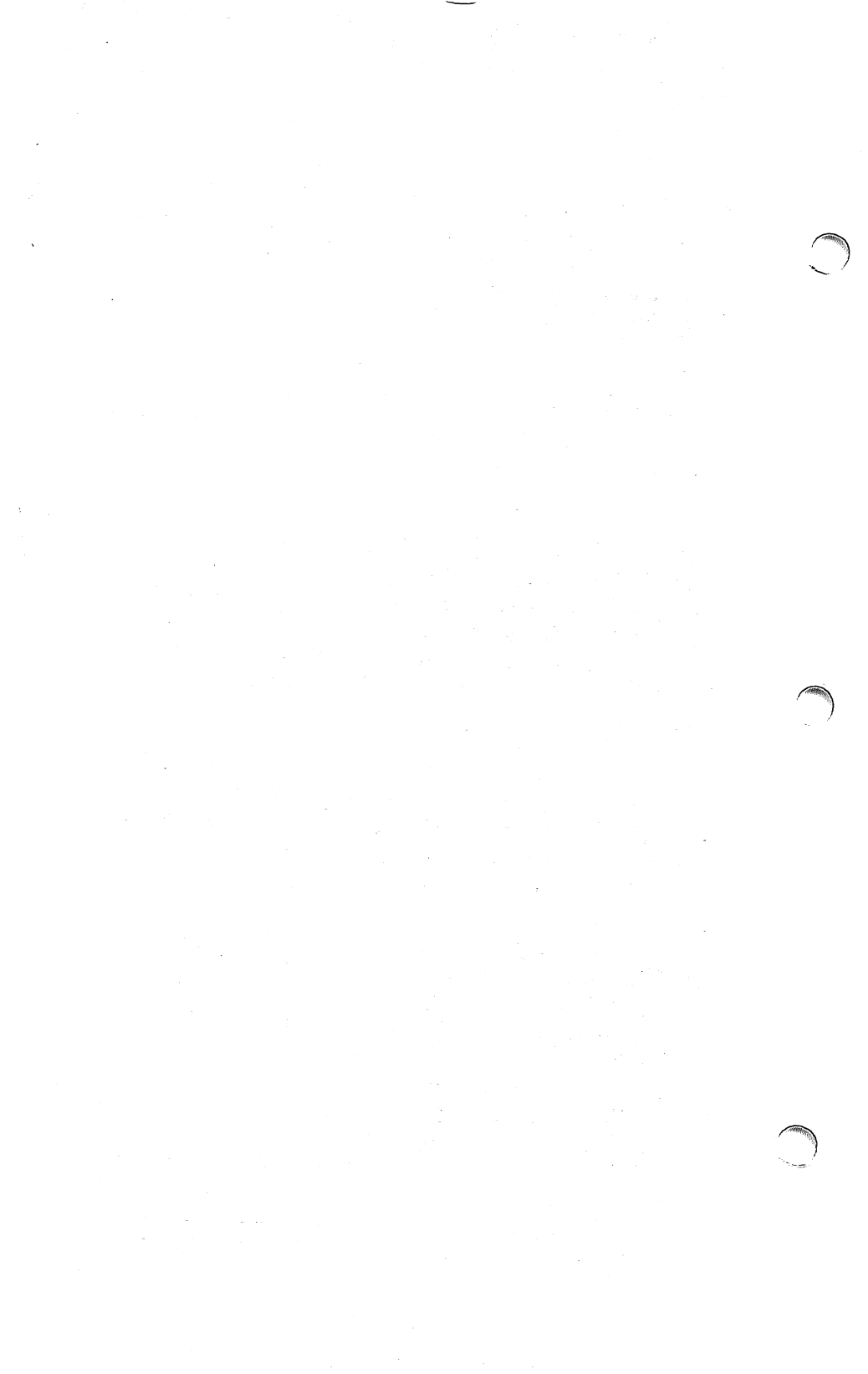


ASIDE ! Thanks to the friendly folks at JMG, quite a few 32 x 32 icons, and a few 48 x 48 icons, have been included with your copy of HyperLINK. Considering the pathetic nature of most icon editors, you should really appreciate this.

NOTE ! HyperLINK will of course have its own icon editor in the near future, but rather than hold up the release of the product, we will be sending it out as soon as it is available to all registered owners.

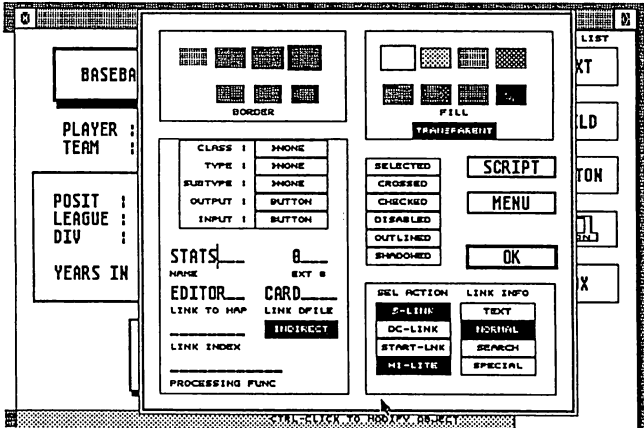
In our Baseball Card Application, we've used the 32x32 icon "Book2". After loading this on top of the box, and also dragging a Text object onto the box as well for labelling purposes, we discover a problem. This Box has a couple of children, the Icon, and the Text label. When a user is using this application, the way it stands anyway, the Text object will be one item, the icon will be another, and the Box itself will be a third. This could really make any linking procedure difficult. So to make it easier, on top of the entire box, why not put an invisible box on top of the other box. This will keep everything in the original box still visible, and make it necessary for only one link. Only Boxes can be made invisible, or transparent. Drag a box over the other box, then double click so we can get into the Object Editor. To accomplish our goals we need to set this box as Transparent. (You will find the Transparent selection in the Fill Type Selector.)

Now what??? Well we also want this box has to be selectable, and the whole point is that we want it to link to a text file. For this, we need to go into the EXT ITEM Selector. Here we find the answers to our needs. To make the Box selectable you have to do two things. First click on the Output selector. Here select Button. Next click on the Input selector. Once again

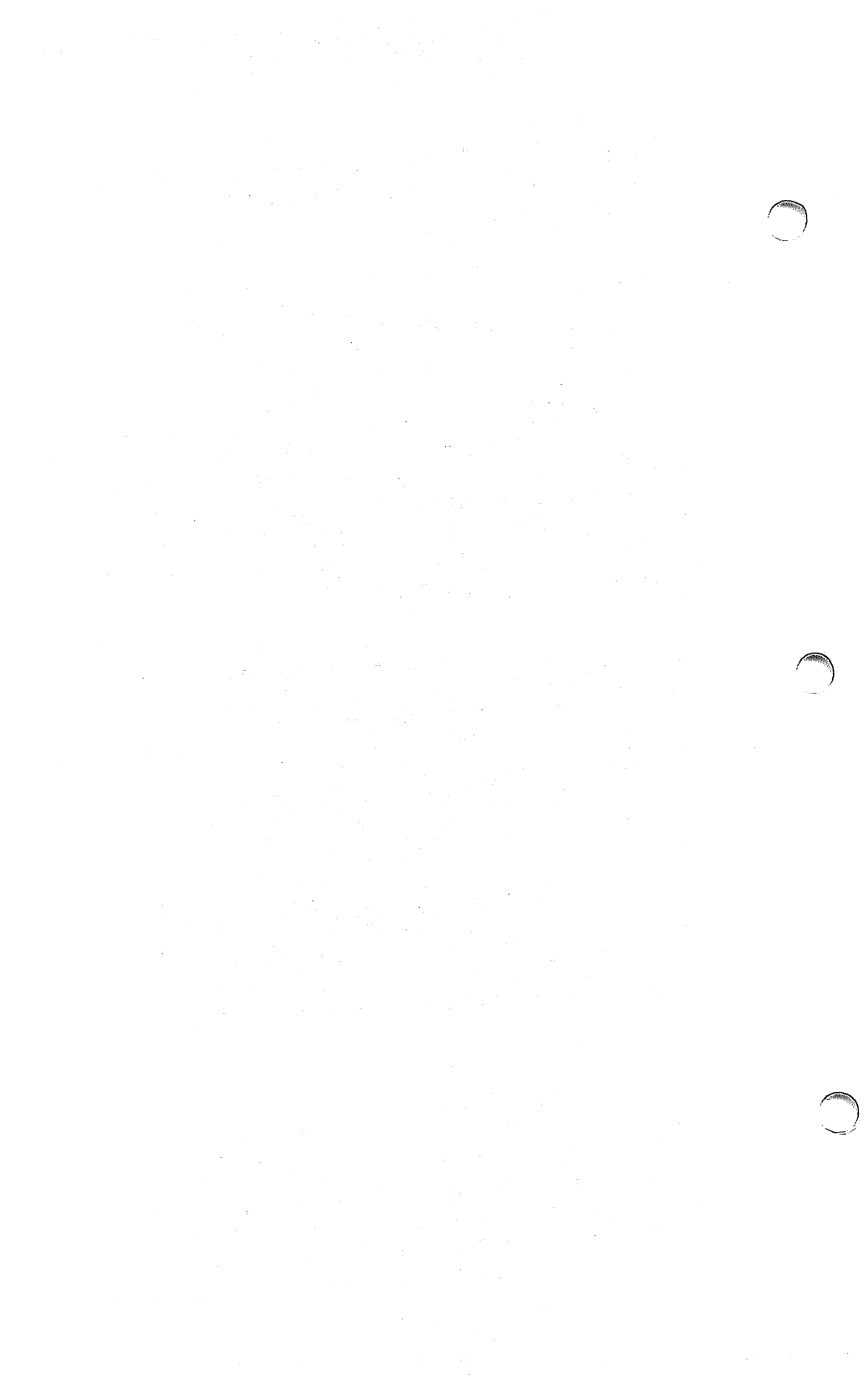


we select Button. Simple right! Now we have made our Box really an invisible Button on top of the Icon, Text and Original Box.

Now comes the fun part... Linking... The whole idea behind HyperLINK! First, how do you want the link called up? by Double Clicking the mouse? by Single Clicking the mouse? In our case we have selected Single Clicking : S-LINK. This option is located under the SEL ACTION menu on the object editing screen. Next let's figure out how we want the application to link. Well if we click on this button, HyperLINK should look up the Card Number for the record we are in, then link it to any corresponding text file. Therefore, you must have text files which use as their file name, the matching number for the card you are linking to.



To accomplish this link : First name the invisible button. For our purposes we will call it STATS. Next, since we are linking to a text file, the LINK TO HAP must be set to EDITOR since that is the HyperLINK application we are using. Then we have to tell our application that this is an INDIRECT LINK. (Meaning that it needs information from another field in the

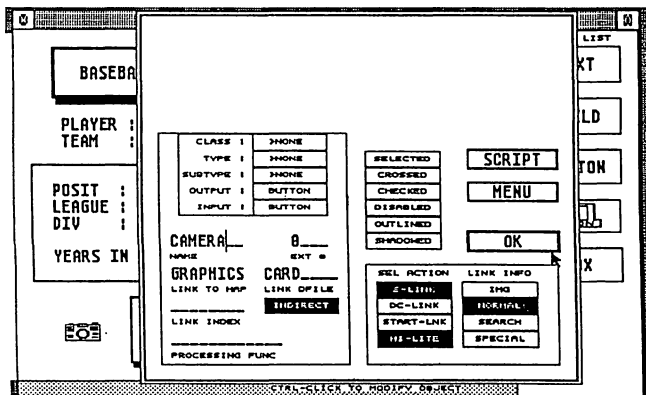


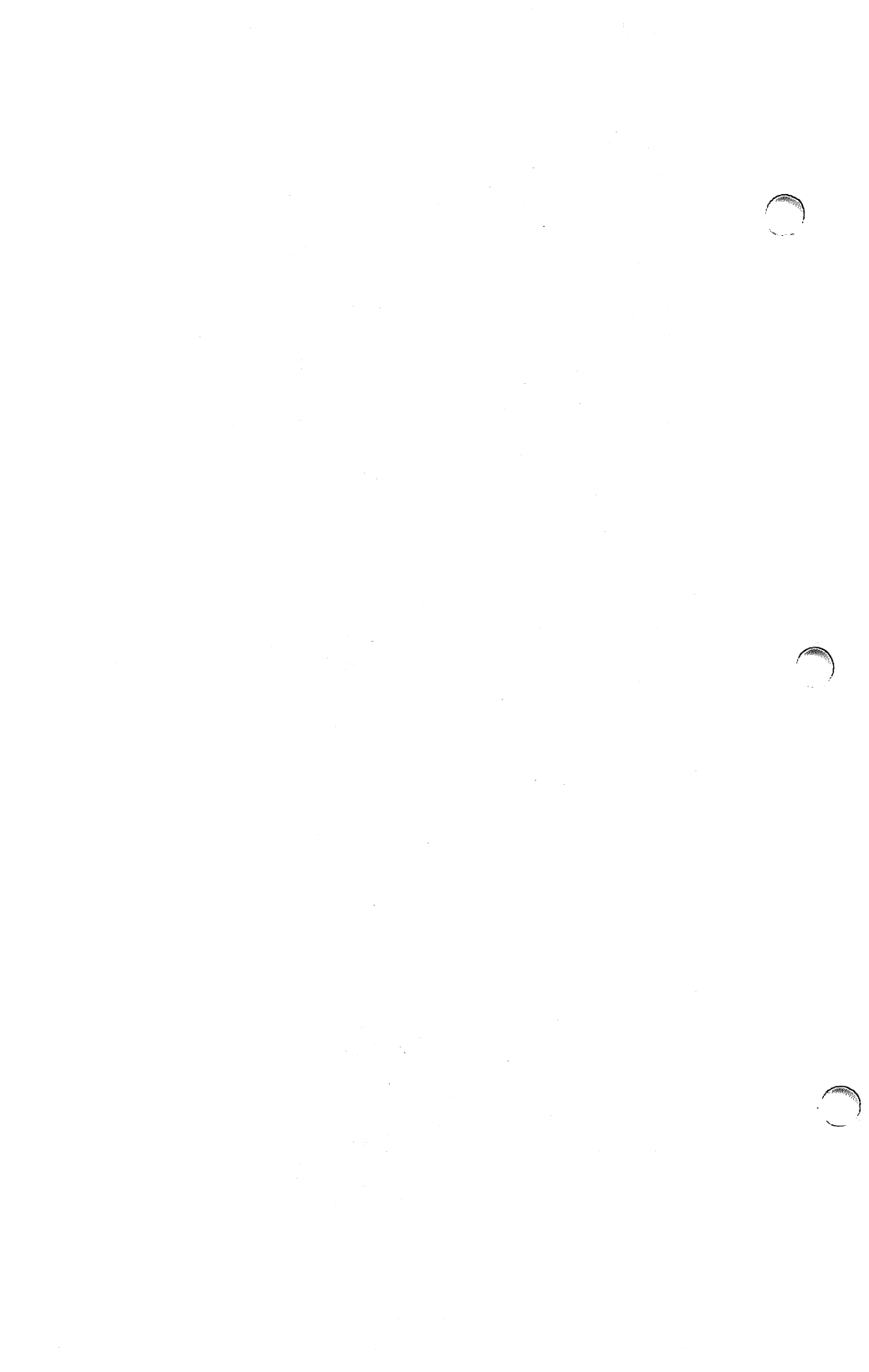
application from which to base its link search.) So we select the INDIRECT Button on the editor screen. Lastly we have to tell which field to get this information from... in this case, the CARD field in our database.

To explain in brief : when this button is selected while running the Baseball Card HAP, HyperLINK will take the information from the CARD field (ex Card # 144), then look up using the EDITOR HAP a corresponding text file. (ex. 144.TXT) HyperLINK will then open the Editor Application and load in any text in this file.

What next for our Baseball Card HAP? Since HyperLINK also comes with an IMG Graphics Module and Application, why not have scanned images of the cards load in? WOW... Another linking function!

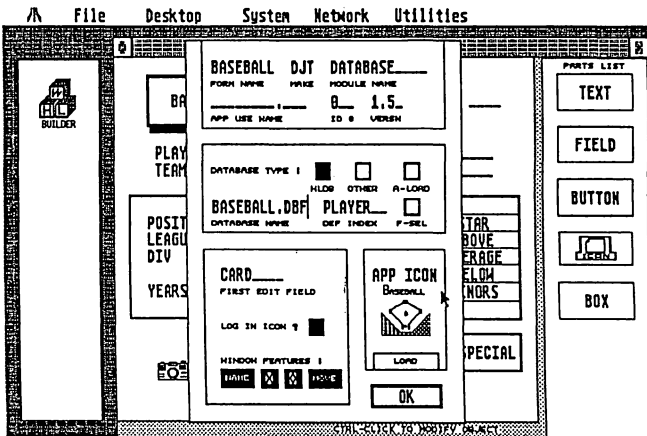
Let's drag another icon onto our application... Seeing that we're going to showing pictures, how about using the camera as our icon!

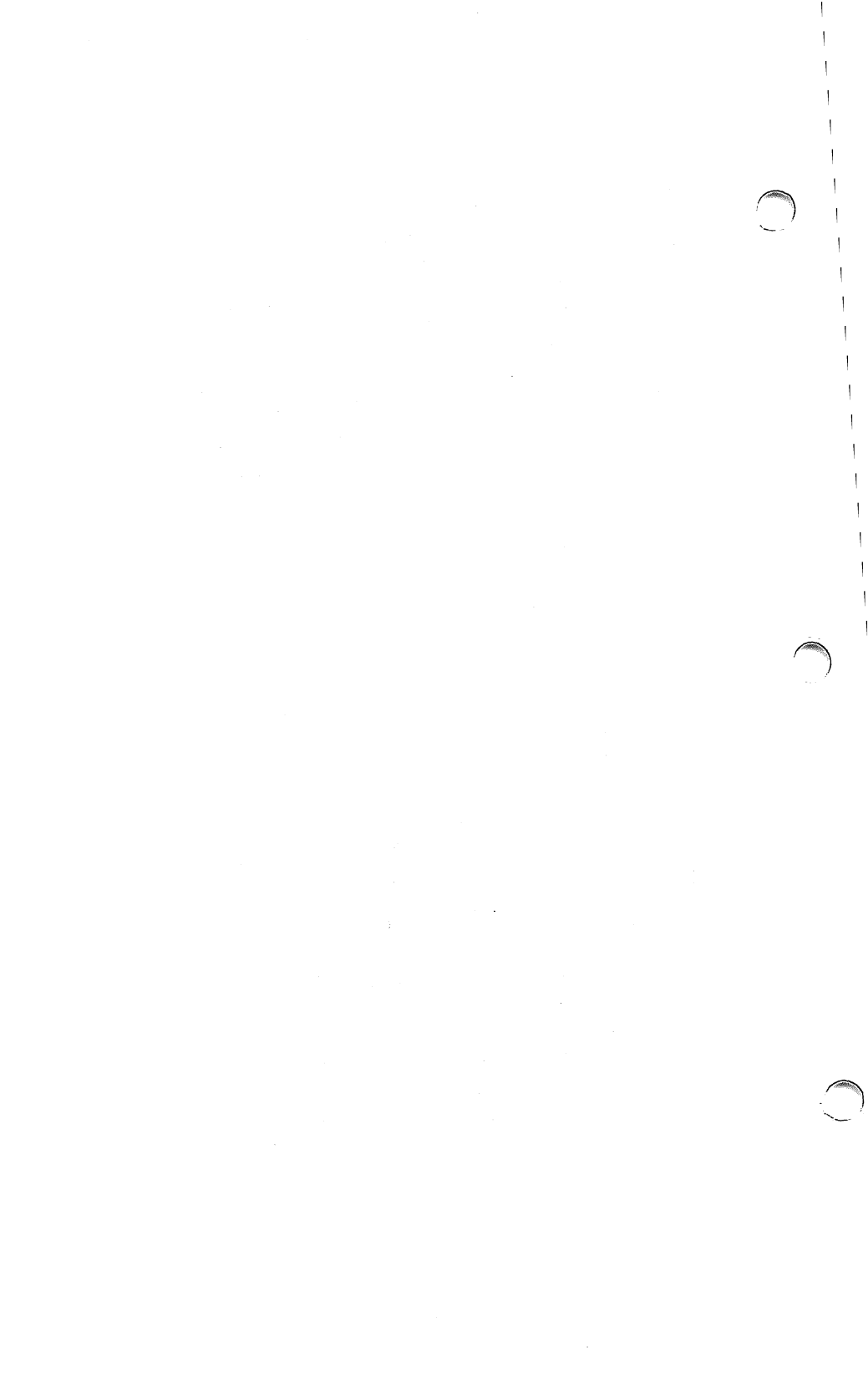




Once the camera is placed on our form, Double Click on it to call up the object editor screen. We have to make it a button first of all, so click on Output and make that a button, then click on Input and make that a button as well. Let's give the icon a name, how about CAMERA! If we are going to be importing IMG files, we have to use the GRAPHICS application, so enter that in the LINK TO HAP. Next, once again this link is going to be INDIRECT since it has to take data from another field in the application before it can do the link. And also again, we'll use the CARD number as the linking search. Therefore, as the text file had to be named 144.TXT, the IMG file has to be named 144.IMG. The final decision for this link is whether it is called by a Single Mouse Click or Double. Single Click seemed more appropriate for this button, so that's what it is!

Essentially all that remains to be done within the Builder is the General Application Information. To get into this, click the right hand mouse button and select APP INFO.





Here we have to fill out...

· FORM NAME : which is simply whatever you the user want to call your application. BASEBALL.HAP seemed appropriate for our use!

MAKE is an optional part of the name that can be used as an identifier of who made the form. "DJT" are the initials of the wonderful creator of the Baseball Card HAP.

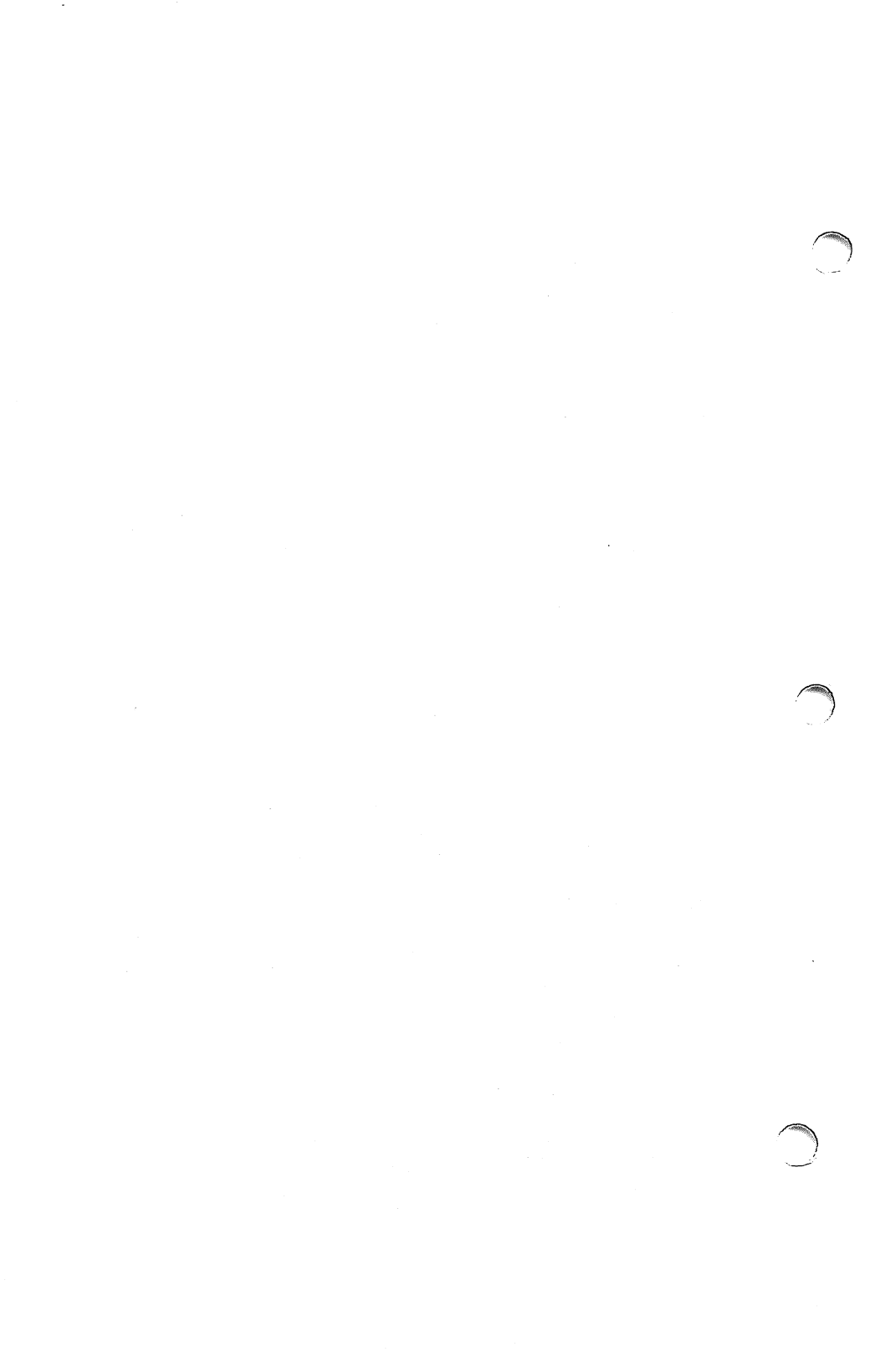
MODULE NAME is the HyperLINK Module which this form uses. The Baseball Card HAP being a database, it uses the DATABASE MODULE from HyperLINK.

APP USE NAME is mostly there for future use. It would be used if a HyperLINK application used two files. This would be where the second file name would go. No need for this is our demo!

ID# is a unique number for each application that can either be used or ignored. Future developments in HyperLINK may use this to a greater extent. Didn't need it, so we ignored it!

VERS simply refers to the version number of your HAP. As you add more features and changes, you may want to change your version number so that users may reference the version of your HAP that they have. We modified the Baseball Card HAP a few times, so it ended up with a version number of 1.5.

DATABASE TYPE : HLDB is a standard DBase type format; OTHER is used if it is not not an HLDB. BASEBALL.HAP uses standard HLDB format.



A-LOAD is selected if you want your HyperLINK Application to automatically load the first record when it is called up. The Baseball HAP does not load the first record automatically.

F-SEL brings up a file selector to select the data file your application is using when that application is called up. This is not currently usable with Database Applications, so we won't be using it with the Baseball HAP.

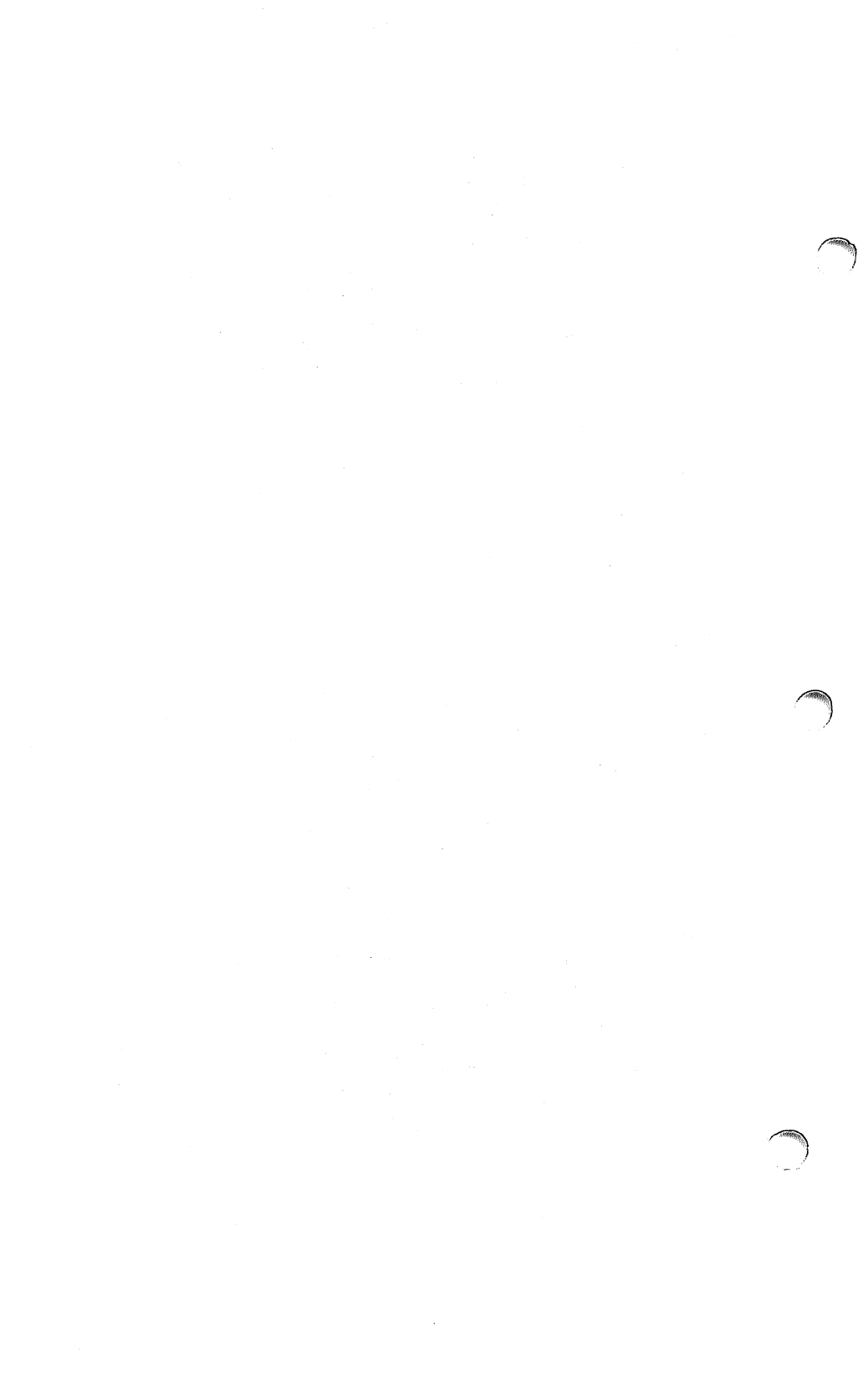
DATABASE NAME is the name of the data file that an application uses by default. When your application is first started, if it does not find a file with this name, it will ask if you want to create a new one. Users should use a DBF extension on their file names, but this is not required. For the use of our demo, we look for a file called BASEBALL.DBF.

DEF INDEX is the default index (or sorting criteria) by which your database will be sorted when it is first called up. For the Baseball card HAP we use the NAME field for default indexing. Now as we go through the records of players we see they have been arranged (or sorted) in alphabetical order.

FIRST EDIT FIELD indicates the field in your database that HyperLINK will default to when you begin an edit. As a general rule, for esthetics anyways, you should use the first field on your form. Therefore the CARD field is the first edit field.

LOG IN ICON ? When this is selected, an icon will appear in the HAP Applications Selector on the left side of the screen. If this is de-selected, then no icon will load in. We have it selected in our Baseball App.

WINDOW FEATURES By selecting or de-selecting, you change the characteristics of your application's window, ie. whether the window can be closed,



minimized, moved, or whether the name appears on the window. In the Baseball HAP, all of these features are selected so that the window can be moved around, closed, minimized, etc.

APP ICON is the icon that loads up in your HAP menu (use a 48 x 48 icon for this purpose). You can select LOAD ICON to change this away from the default. We designed a new icon just for our baseball application. Isn't it great!

You have just gone through the making of a HyperLINK Application! Not too tough is it? Now all you have to do is click the right hand mouse button and save your new application.

The last thing to do before using your HyperLINK Application comes up after you have loaded it into your HAP menu. Holding down the ALT key, click on your new application. This will call up the APPLICATION DEFAULTS screen. We have already covered these earlier in the manual. With reference though to our Baseball Card HAP, maybe we should just mention again that this is the screen from which to create indexes for your database. If for instance you wanted to create an index by TEAM, this is the place!

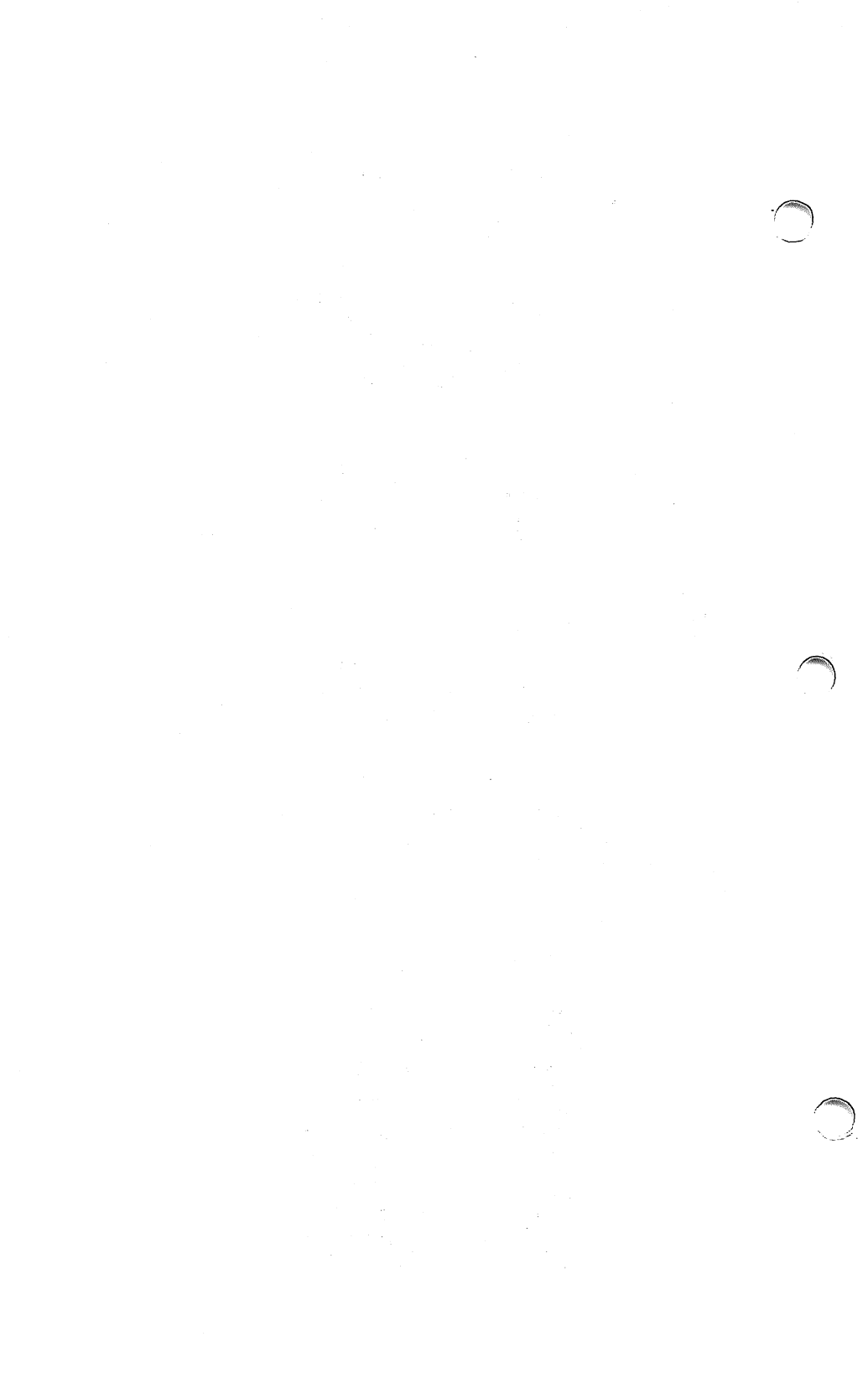
DESKTOP SYSTEM NETWORK UTILITIES

APPLICATION DEFAULTS

App Name : BASEBALL
Module Name : DATABASE
DataBase File : BASEBALL.DBF
Default Index : PLAYER LIST
Gen Filename :
DB Type : HL-DB App Optn : 8

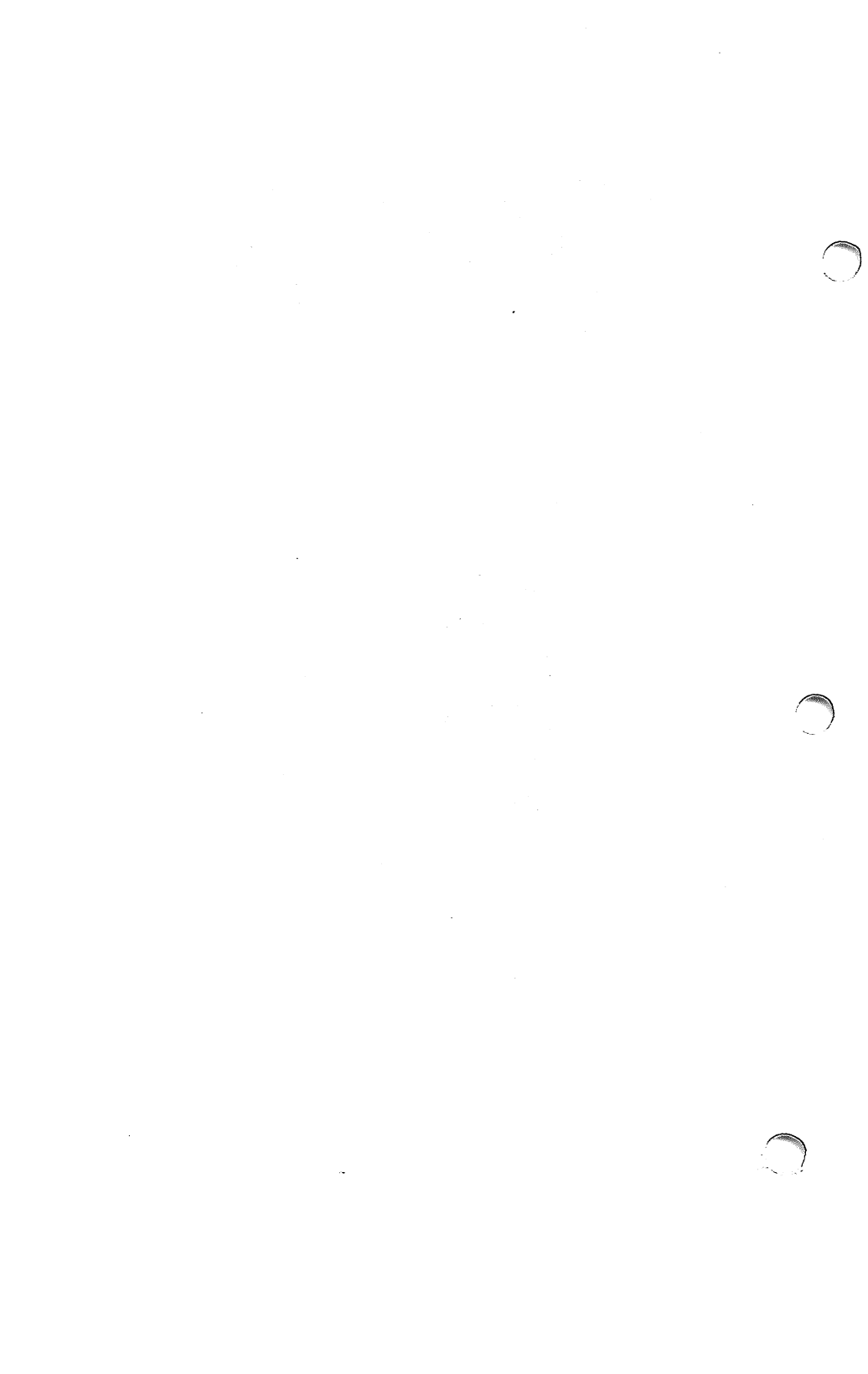
1st Rec Load Module Defs

Create INDEX PACK
RE-INDEX CHECK Data
Delete INDEX Backup Data
OK CANCEL



Hopefully this step-by-step building of a HyperLINK Application has helped you understand the way in which HyperLINK operates. Further demos will be available in future, either posted on GENIE, JMG's own BBS (416) 389-9064 [BBS Available AFTER June 1, 1991], which may help you grasp the potential of HyperLINK even more.





The LINK COOKBOOK

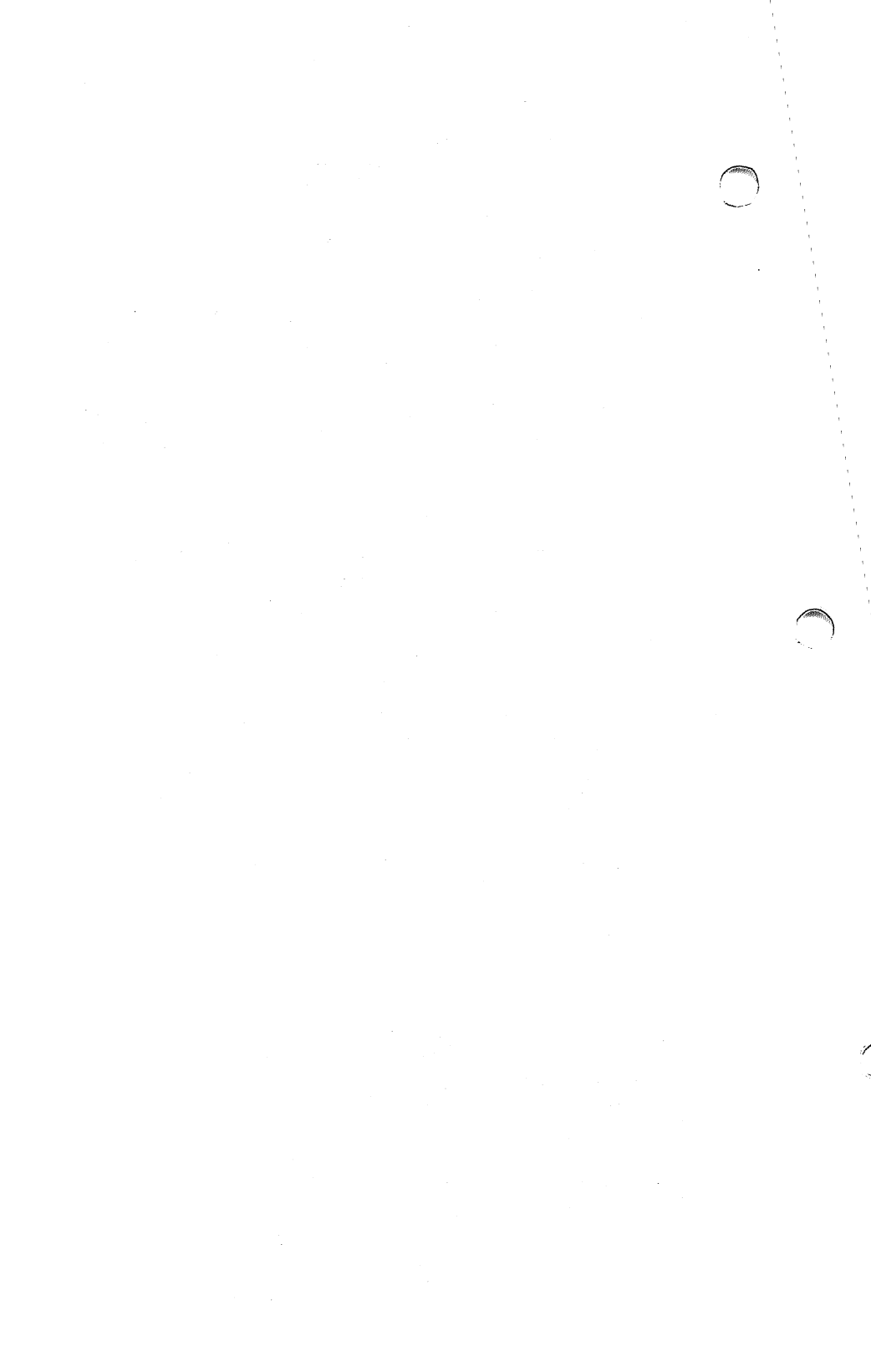
A Basic Description of Link Types and Linking.

The concept of "Linking" is at the very heart of the HyperLINK program and its modules. Because of the versatility HyperLINK provides in linking dissimilar items and objects, this entire area of the program can become confusing and difficult to grasp. Hopefully this section of the manual will help clear up any such confusion, and shed some light on this very powerful feature of the HyperLINK system.

Before we begin explaining what linking really is about, we should make clear two points: first, each link is controlled by information that you provide and manipulate. In general, the two places where this happens are in the Item Info editing form of the BUILDER, and in the TextLink editing form. This chapter will help you put correct information into these forms to create the links you want. The second point is that Linking technology is being developed and improved, and as HyperLINK evolves in the future, Linking will grow to be more "intelligent", to the point where HyperLINK will try to figure out what you want to do and fill in the details automatically. For now, however, the details must be provided by us mere humans.

A HyperLINK Analogy

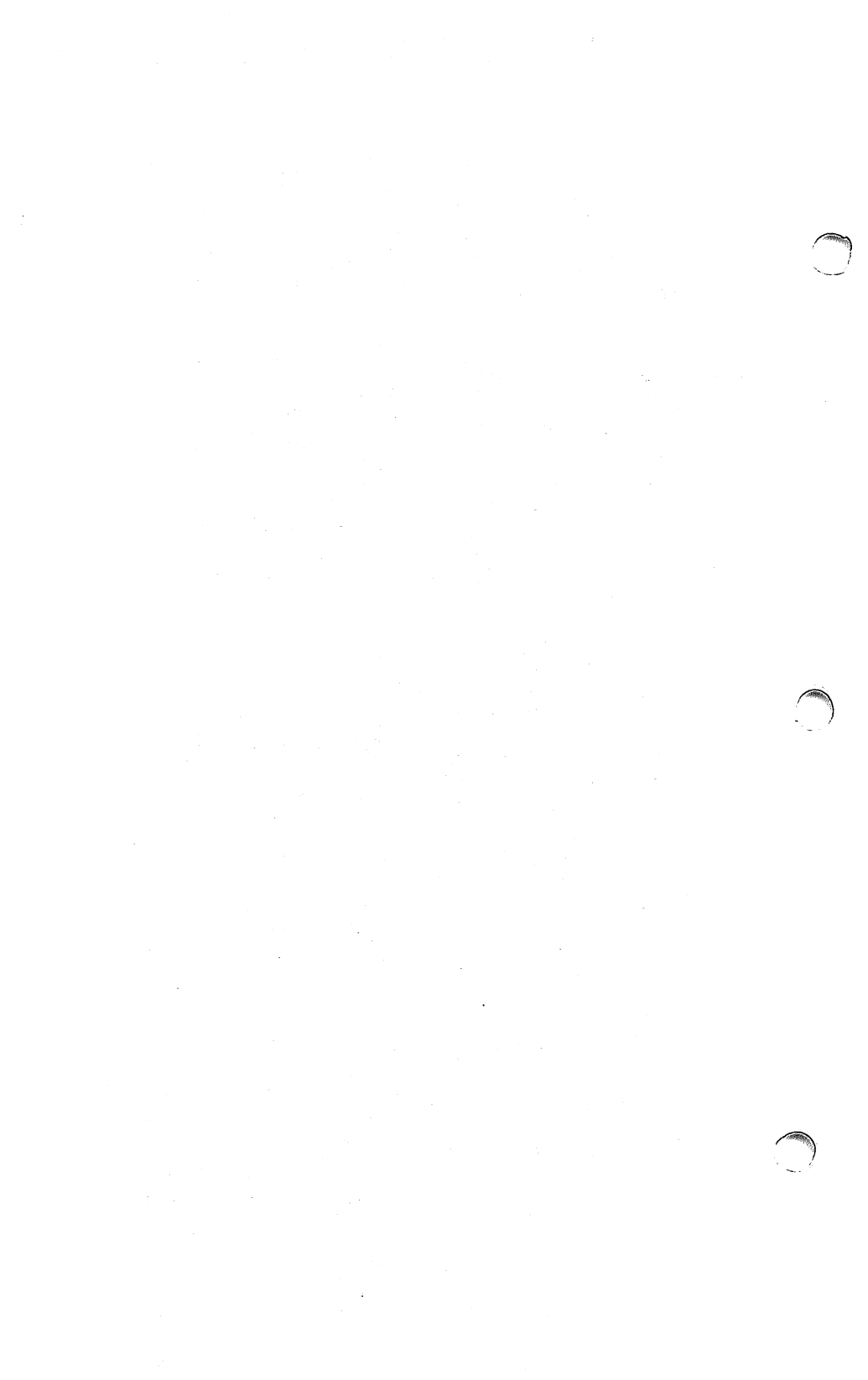
Since the ways in which HyperLINK handles "linking" can be complex to describe from scratch, we decided that an analogy might be in order here. The one we chose is that of a little company operation, with a shipping desk and a receiving desk. Inside each company are a bunch of little managers, each in their own offices. The company represents a HyperLINK MODULE. Each little manager is a HyperLINK APPLICATION. There may only be one, or there may be many of these guys in each company.



The shipping and receiving departments handle shipping (sending) and receiving LINKING PACKETS, which you can think of as little forms that each company sends and receives by courier. The courier is, of course, HyperLINK itself, directing these packets from company to company (module to module).

So, let's take a trip down to the DATABASE company. If we look in, we'll see that there are a few managers in residence - reading the signs on the office doors, we can see ADDRESS department, MESSAGE dept, and BASEBALL dept (three of the sample database applications that come with HyperLINK). As we walk into the BASEBALL office, we see the little manager at his desk, doing what the user is asking of him. Since this is the DATABASE company, he has no trouble calling up a data record, showing it on the screen, taking input as it's typed, and saving it back. Why, this little guy can even do searches, delete records, and other fun database things. But then the user clicks on the button on the BASEBALL form that is marked STATS. The manager looks at the button and sees that this isn't a database item - the item info says that this item is LINKED if selected. At this point the manager takes out a memo form and fills in a series of blanks. The information he puts in these blanks is taken right from the item info list of the STATS button, info that was recorded when the BASEBALL application was created. Our little manager then rushes out and hands this form to the Shipping desk, saying "Send this out right away!"

The shipping department takes the memo; depending upon what the manager wrote, the shippers might have to add some information to the form, but usually they'll just call a courier and have it sent off right away. When the courier comes, they'll take the packet and look up the full address. (All the managers are a bit lazy, you see. They'll only write the destination office name on the memo, not the company that the office is in. Luckily the courier has no problem

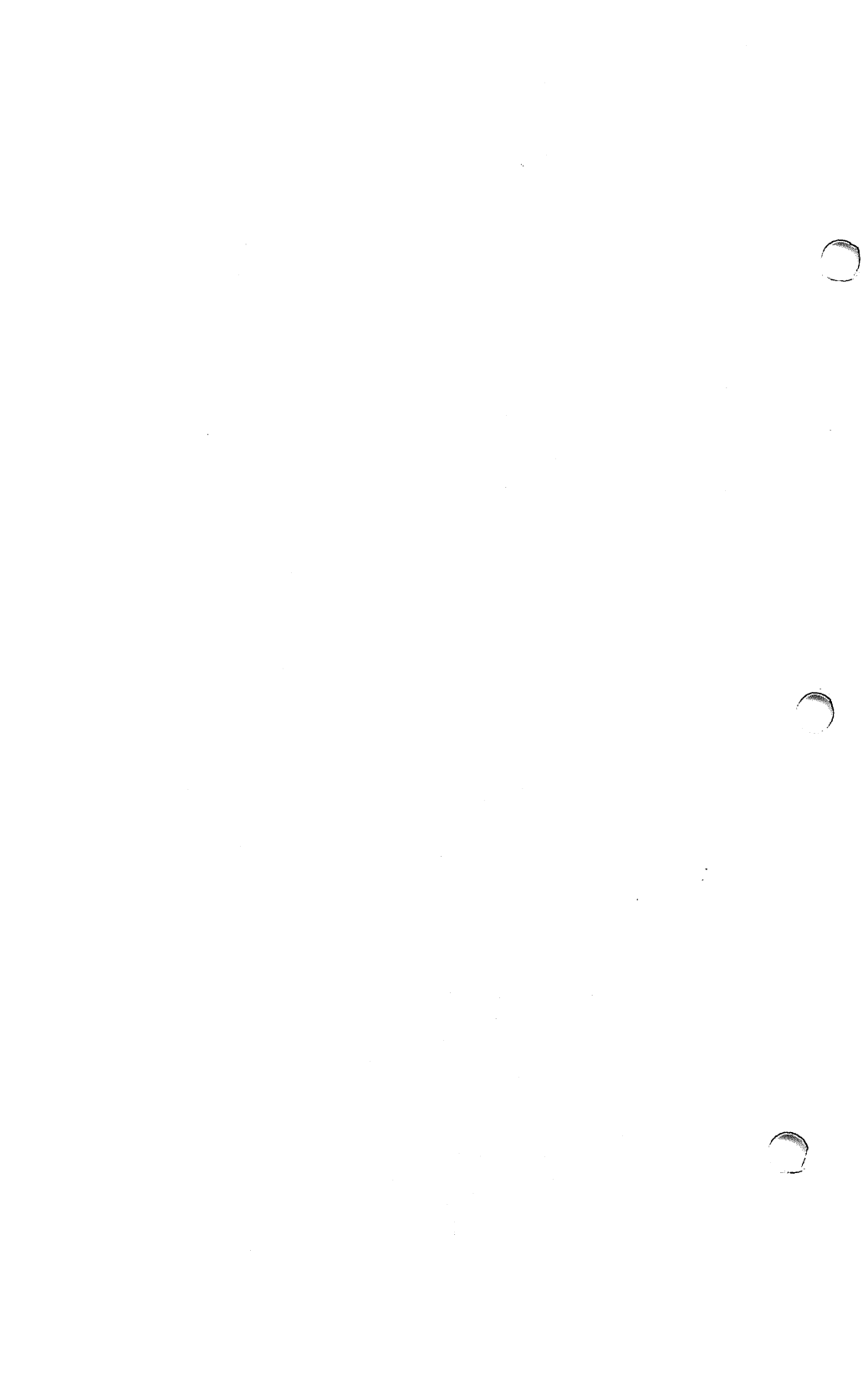


looking up which company the office is in, and of course no two offices can ever have the same name.)

In the case of this memo, from the STATS button, the destination office is EDITOR, and the company is HL_EDIT (that's the module name). So the courier rushes out to the HL_EDIT company and hands the memo to their receiving desk. They look at it, rush it up the EDITOR office and the manager there follows the instructions on the memo. The memo will tell him to open up the EDITOR window, and then load a specific file into it. After he does this, the memo is shredded, having done its job.

Now supposing the user clicks on the camera icon, the BASEBALL manager rolls into action again... same procedure, but the CAMERA button lists GRAPHICS as the destination application (office). Once again, the courier looks up the address and sends the memo off to the correct company (HL_IMG in this case). Once the GRAPHICS manager gets a hold of the memo, he follows the instructions, loading up the file name he was given and showing it in a window.

Before discussing the details of these memo forms (the Links in HyperLINK), we should detour a bit and look at the types of companies involved... Some, like DATABASE, do all sorts of things. They have a shipping department and a receiving department (sending and receiving links), so they are called ACTIVE. Others, like HL_IMG, only have a receiving department - they never actually send anything out, they only receive memos from others. These are PASSIVE companies (Applications). Another difference between some companies is the offices themselves - DATABASE has many, such as the ADDRESS database, the MESSAGE database, the BASEBALL database, and you can add more offices very easily by loading in new database HAPs. These are called GENERIC. Other companies, like HL_EDIT and HL_IMG, are a bit different. They only have one office, never more. Unlike a database, which can be

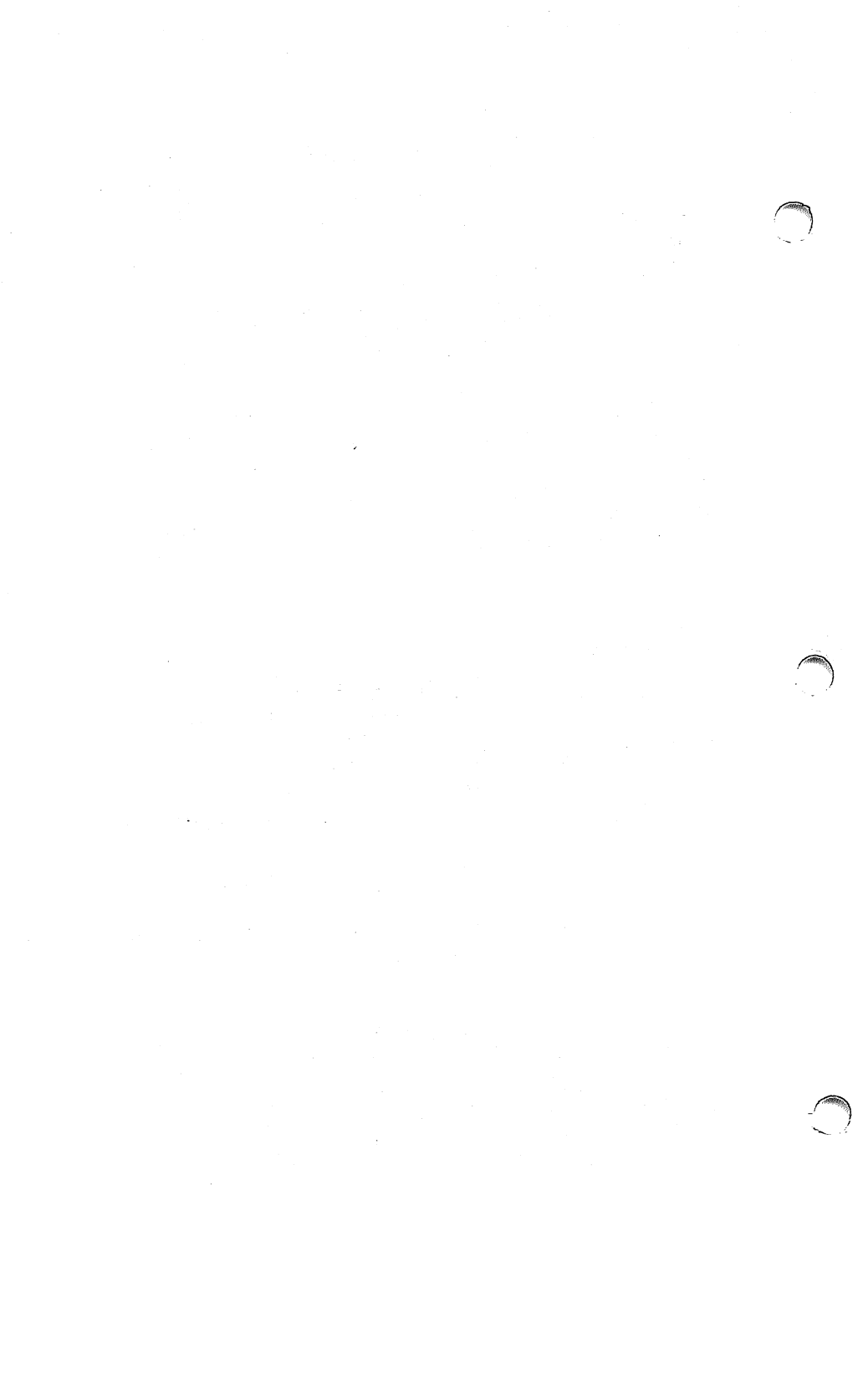


modified and built in hundreds of different ways, or an Animation player, which can have a different animation film in each office, companies like HL_IMG can only do one, specific thing. HL_IMG gets a picture file from disk and shows it in a window. HL_SND gets a sound file from disk and plays it through the speaker. They each do only one specific thing, and so they only need one office and one manager. These are called SPECIFIC applications.

Now back to our memo system. At present, only two companies actually send out memos ("active modules"), these being DATABASE and HL_EDIT. We'll run through some examples with these. First, let's take that STATS button memo that was sent out earlier. Just what was on that memo form? Well, each memo form has these lines to be filled in:

- Link Type :** An indication of what type of object is being linked to.
- Link To :** Which office (application) to deliver the link to.
- Link Dfile :** The name of the file for the receiving office to load.
- Link Idx :** Usually the name of a sort index to use.
- Link Search :** Some extra information, such as a name to look up.
- Link Appdef :** This is just a number passed on to the destination.

So, where did our little BASEBALL manager get the information to fill out this form? Simply he took it from the Item Information for the STATS button, and this info was entered using the Builder when the BASEBALL application was first created. Lets take a look at the Item Info form for the STATS button:



 BORDER		 FILL																							
<table border="1"> <tr><td>CLASS :</td><td>>NONE</td></tr> <tr><td>TYPE :</td><td>>NONE</td></tr> <tr><td>SUBTYPE :</td><td>>NONE</td></tr> <tr><td>OUTPUT :</td><td>BUTTON</td></tr> <tr><td>INPUT :</td><td>BUTTON</td></tr> </table>		CLASS :	>NONE	TYPE :	>NONE	SUBTYPE :	>NONE	OUTPUT :	BUTTON	INPUT :	BUTTON	<table border="1"> <tr><td>SELECTED</td><td>SCRIPT</td></tr> <tr><td>CROSSED</td><td></td></tr> <tr><td>CHECKED</td><td></td></tr> <tr><td>DISABLED</td><td></td></tr> <tr><td>OUTLINED</td><td></td></tr> <tr><td>SHADOWED</td><td></td></tr> </table>		SELECTED	SCRIPT	CROSSED		CHECKED		DISABLED		OUTLINED		SHADOWED	
CLASS :	>NONE																								
TYPE :	>NONE																								
SUBTYPE :	>NONE																								
OUTPUT :	BUTTON																								
INPUT :	BUTTON																								
SELECTED	SCRIPT																								
CROSSED																									
CHECKED																									
DISABLED																									
OUTLINED																									
SHADOWED																									
STATS ----- 0 ----- NAME ----- ----- EDITOR ----- CARD ----- LINK TO MAP ----- LINK DFILE ----- ----- ----- LINK INDEX ----- ----- PROCESSING FUNC -----		<table border="1"> <tr><td>SEL ACTION</td><td>LINK INFO</td></tr> <tr><td>E-LINK</td><td>TEXT</td></tr> <tr><td>DC-LINK</td><td>NORMAL</td></tr> <tr><td>START-LNK</td><td>SEARCH</td></tr> <tr><td>HI-LINK</td><td>SPECIAL</td></tr> </table>		SEL ACTION	LINK INFO	E-LINK	TEXT	DC-LINK	NORMAL	START-LNK	SEARCH	HI-LINK	SPECIAL												
SEL ACTION	LINK INFO																								
E-LINK	TEXT																								
DC-LINK	NORMAL																								
START-LNK	SEARCH																								
HI-LINK	SPECIAL																								
<table border="1"> <tr><td colspan="2">TRANSPARENT</td></tr> </table>		TRANSPARENT		<table border="1"> <tr><td>MENU</td></tr> <tr><td>OK</td></tr> </table>		MENU	OK																		
TRANSPARENT																									
MENU																									
OK																									

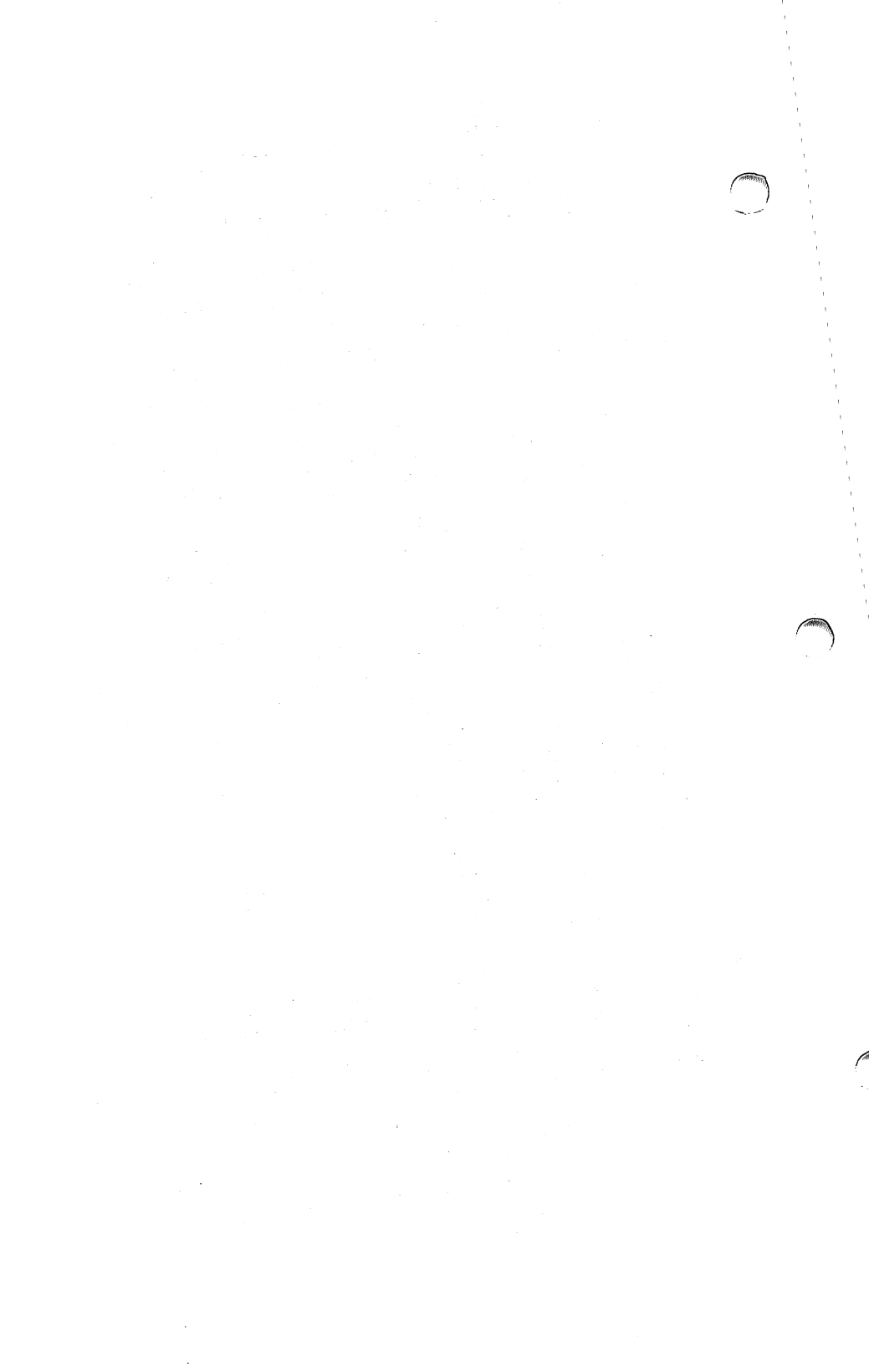
The LINK TYPE was taken from the box at the right side... when the box is clicked on, a choice of link-types pops up. What does the link-type indicate? Well, for the most part it shows what type of "object" you are linking to. Is it a text file? A graphic IMG file? A sound file? A Command file? Now it should

>LINKTYPE
NUL
CMD
DATA
TEXT
GEM
IMG
SND
SPCH

be noted that this part can actually be left blank at most times. If you are linking to the EDITOR office, it will assume that you are asking for a text file, and go from there. (So if you leave link-type unset, or set to NUL, these assumptions will be made.) So what good is link type? Some applications may let you ask for either graphics OR text, and setting the link-type will let you show what you want. And if you set link-type to IMG and send the memo off to the EDITOR office, it will very efficiently be ignored. (No, the mail is not forwarded in HyperLINK!)

The LINK TO, LINK DFILE, and LINK IDX lines are all filled in from the left hand side, at first just taken directly from the Item Information. But wait, it is not always this simple. Looking at the Item information, you can see that it has an INDIRECT setting. This means that the BASEBALL manager has to do a little cross-referencing before he sends the memo out.

What does the INDIRECT setting do? Well, ordinarily the file name to load after a link is taken directly from the link-dfile entry. So if indirect were not set, every

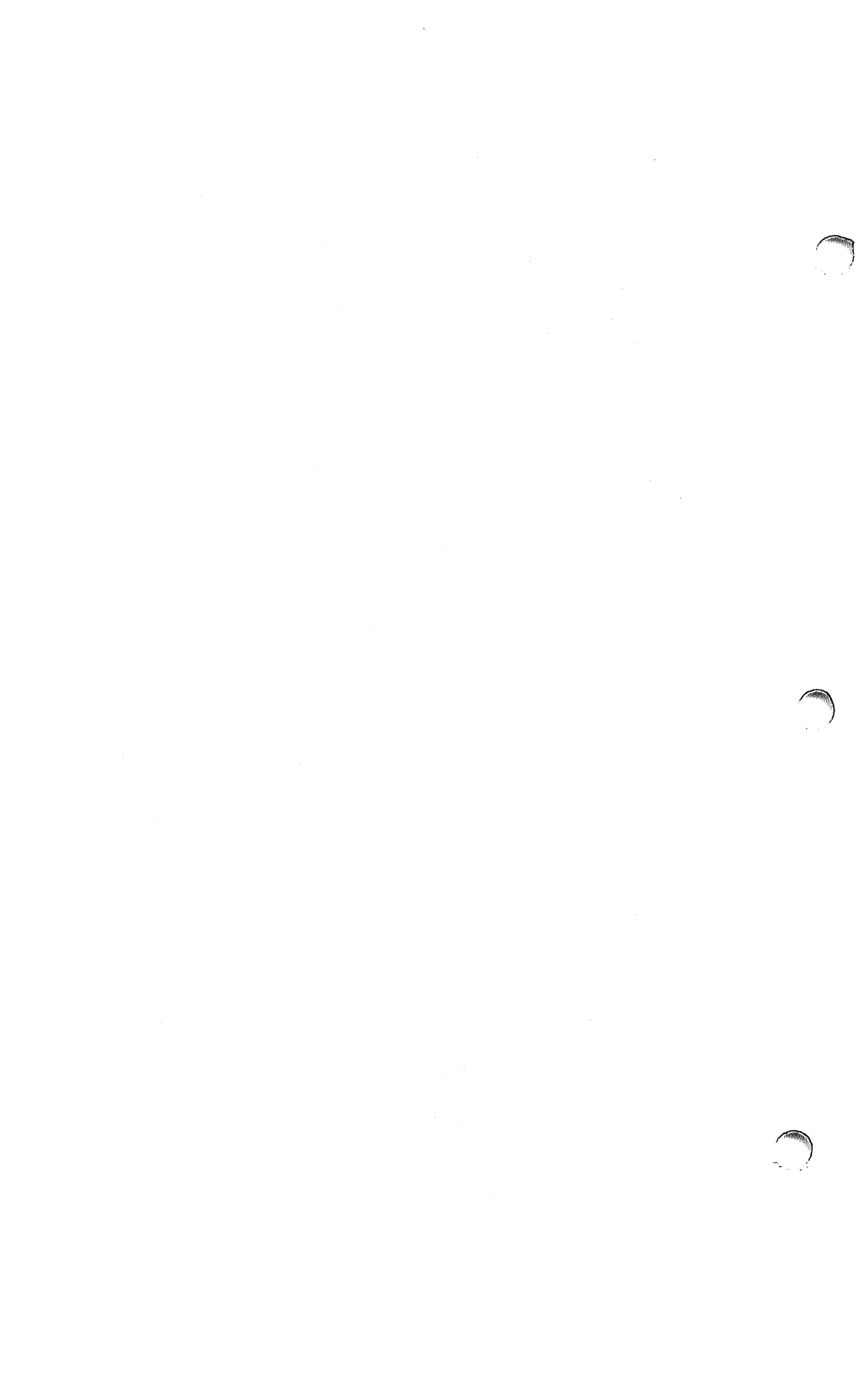


time you clicked on STATS it would load up a file called CARD.TXT (the name in link-dfile). But no, what you want is to load up a different text file for each card, and the filename for the card you want to load changes with each record of the database. So make it INDIRECT, and then use link-dfile to tell it where to find the filename. The "where" is the name of the item (a field) that the user enters into as he puts data into the database. (The name you put into link-dfile is the name that was given to the referenced field in the Builder.)

So look what happens now, with the INDIRECT set. Our little manager takes the name that was in link-dfile and uses it to look up another Item Info form - that for CARD. Then he looks into the contents of the CARD field itself. As an example, Wade Boggs is "144". So then the manager fills in "144" into the link-dfile line that he writes onto the memo form.

The LINK IDX and LINK APPDEF are set directly from the Item Info and passed on in the memo. And what about LINK SEARCH? Well, this is a different matter. It only applies when the Link Type information is changed from the usual "NORMAL" state to either SEARCH or SPECIAL. If one of these is set, the manager has a little extra work to do before sending the memo out: if the current item (i.e. the item that was clicked on) is a text field, then the manager fills into the memo form the actual contents of this field. We'll give an example of that in a moment.

First, let's take a look at what happens when EDITOR gets this memo. He'll check the link-type; if it's not NUL (not set) or TEXT, he'll ignore the memo. Assuming it's accepted, he'll then look at link-dfile. By the time EDITOR has the memo, any indirect stuff has already been looked up and changed, so he'll just use link-dfile as a file name. IF, however, link-dfile is blank, then EDITOR will simply assume that you mean to use the currently loaded file (all applications do this). EDITOR even does an extra bit of work - if



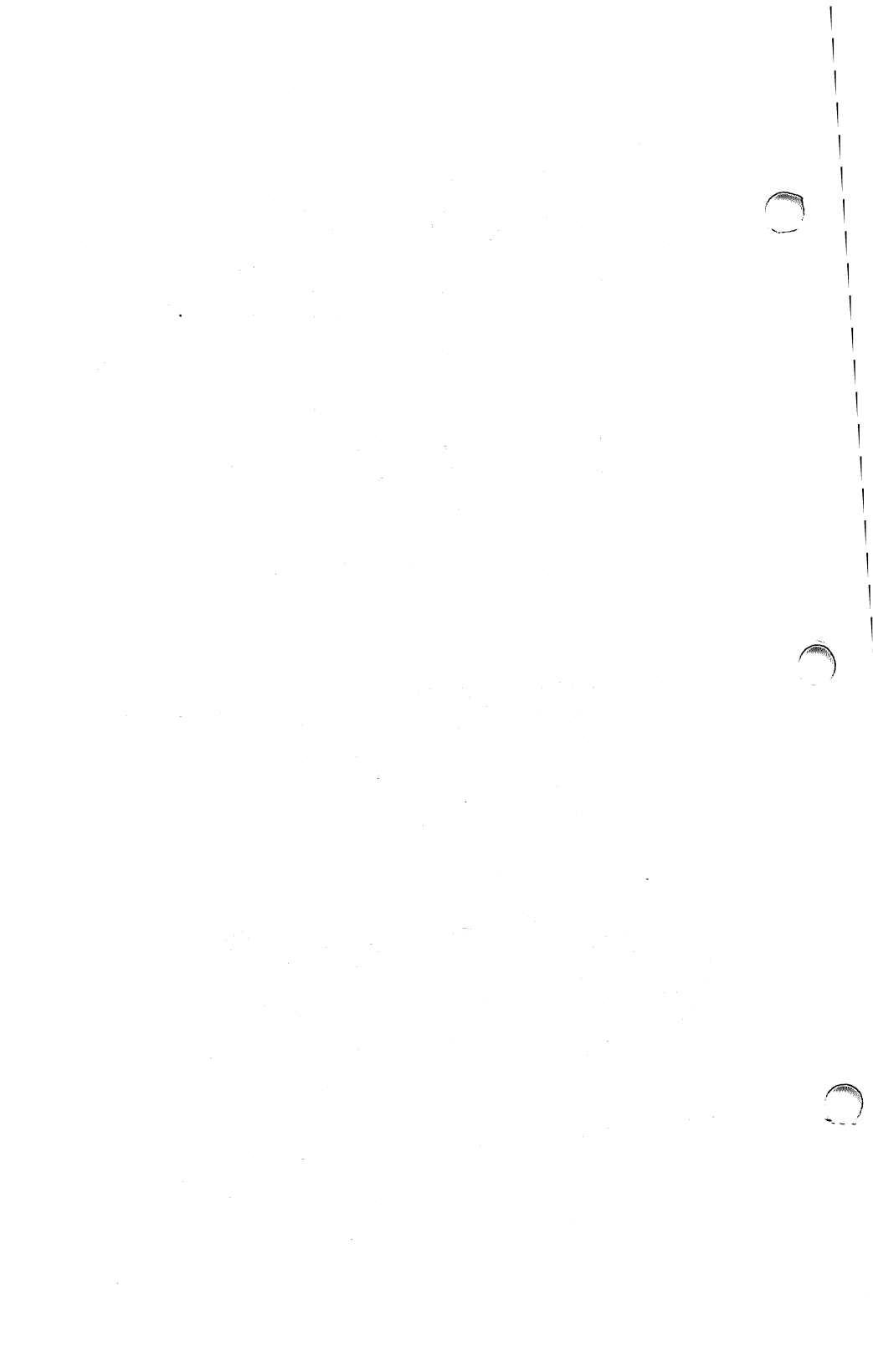
the LINK APPDEF number is not zero, he will use it as a line number to automatically move the cursor to. (This way, different buttons could refer to different parts of a text file.) Then he'll do one last thing - IF the SEARCH button was set on the original Item info, EDITOR will have received a name (or some such string) to look for, and he'll trigger HyperEDIT's text find feature to line up the cursor to it. If NORMAL was set, he'll just leave things like they are. (I can see some of you asking "And what if SPECIAL is set in the original Item Info?" Well, the EDITOR, like most applications; will simply ignore a SPECIAL link. The only exception is DATABASE itself, which uses it to do, well, special things...)

After EDITOR has decoded the link memo, he puts up the HyperEDIT window with the text file as requested.

For another example, let's look at the MESSAGE office. This is also a DATABASE office, but the way it was made it only has links to another DATABASE application - the ADDRESS office. The links in the MESSAGE form are the FROM and OF fields - these link to the ADDRESS form, doing a lookup on either the NAME or COMPANY fields respectively. Take a look at the Item Information for the FROM field in the

The screenshot shows a window titled "MESS" with a menu bar (File, Desktop, System, Network, Utilities) and a toolbar. The window contains several panels:

- Form Fields:**
 - To _____
 - Date ____/____/____
 - From _____
 - of _____
 - Phone _____
 - Received by _____
 - Message _____
- Checkboxes:**
 - Telephoned R
 - Wants Visit W
 - Please Call WI
- Buttons:**
 - SCRIPT
 - MENU
 - OK
- Form Section:**
 - CLASS I FIELD
 - TYPE I CHAR
 - SUBTYPE I NONE
 - OUTPUT I TEXT
 - INPUT I TEXT
 - MFROM _____
 - NAME _____
 - ADDRESS _____
 - LINK TO MAP _____
 - ADD_LNAM _____
 - LINK INDEX _____
 - PROCESSING FUNC _____
- Buttons:**
 - SELECTED
 - CROSSED
 - CHECKED
 - DISABLED
 - OUTLINED
 - SHADOWED
 - LINK INFO
 - S-LINK
 - DC-LINK
 - START-LINK
 - HI-LITE
 - DATA
 - NORMAL
 - SEARCH
 - SPECIAL

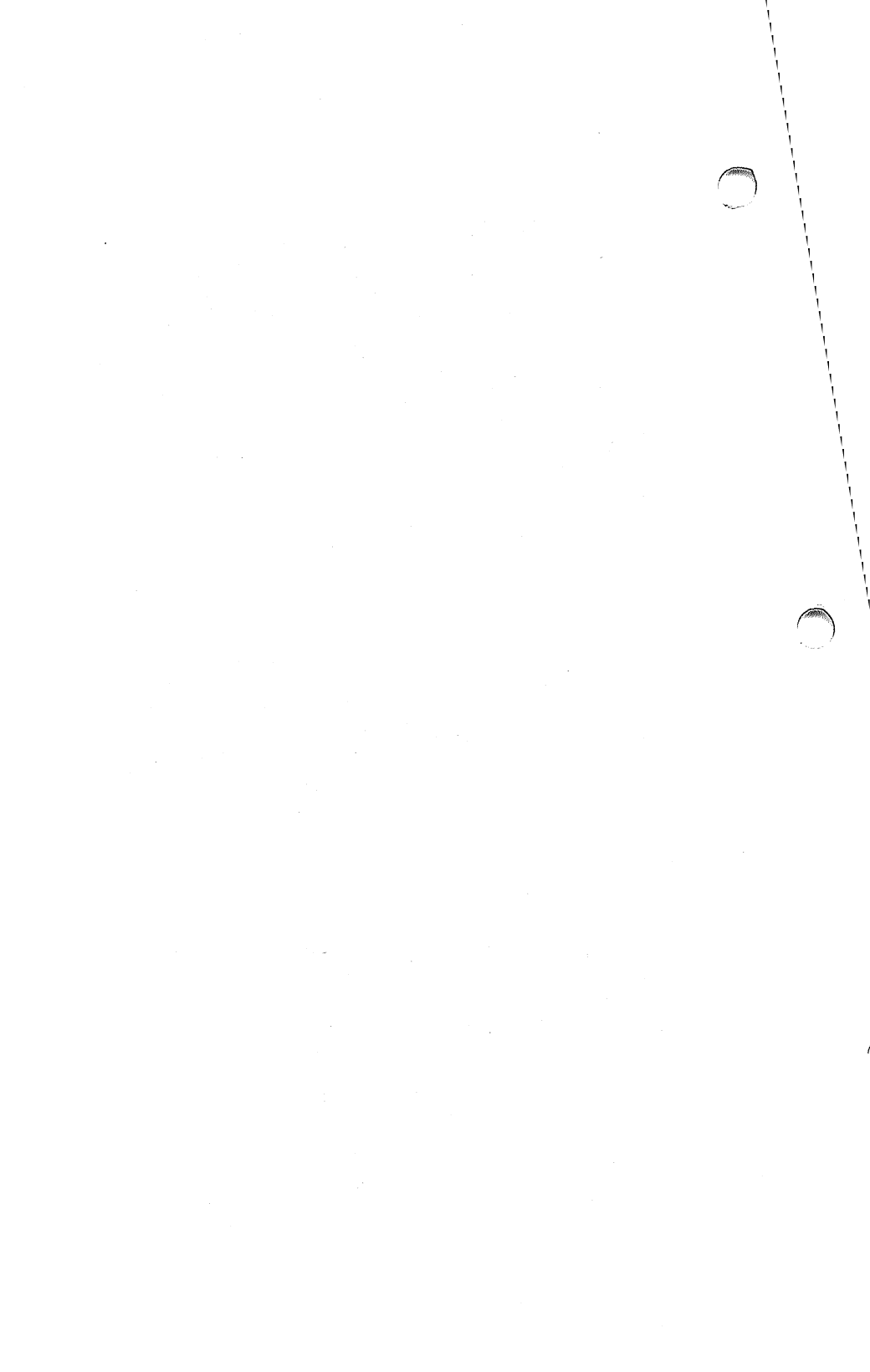


MESSAGE application:

First, the easy stuff. It lists ADDRESS in the LINK TO field, and that's obvious enough. Notice, however, that LINK DFILE is blank. As just mentioned, this means that the database file that the ADDRESS application will use will be the "default", or whatever was used last. If you wanted to use only a specific database to link to, (i.e. CUSTOMER.DBF), you would enter the name here (no extension). The LINK IDX is new here - it is used to indicate which index (sort) should be set to active after the link. This is especially important in database searches; to properly look up a name, we ask for the index called ADD_LNAM. When asking for the company, we ask for the ADD_COMP index. Assuming that these indexes exist (and are properly sorted, i.e. by the name field for ADD_LNAM), then the database search function will have no problem working. This link IS a search link, as you can see by the SEARCH button being set on the right side. So what happens when the manager gets a hold of this? Seeing the SEARCH setting, he'll look into the field itself and write the contents onto the link memo. So if the message form has "BUSH, GEORGE" entered in the name field, this is exactly what will be entered into Search section of the link memo.

At this point, you would think that MESSAGE would walk next door to the ADDRESS office and hand the memo over. Well, not quite. MESSAGE still walks to the Shipping desk, the courier still picks up the packet and checks the address (to see which company ADDRESS is really in), and then it hands it to the Receiving desk at DATABASE. They take it up to the ADDRESS office, and the manager there gets to work.

First, ADDRESS loads up the database file named in link-dfile. If link-dfile is empty, ADDRESS just uses whatever database was used last (or the default). Next, the index is changed to whatever is named in link-idx. Once again, if the name is blank, then no change. Then, the ADDRESS manager takes a look at



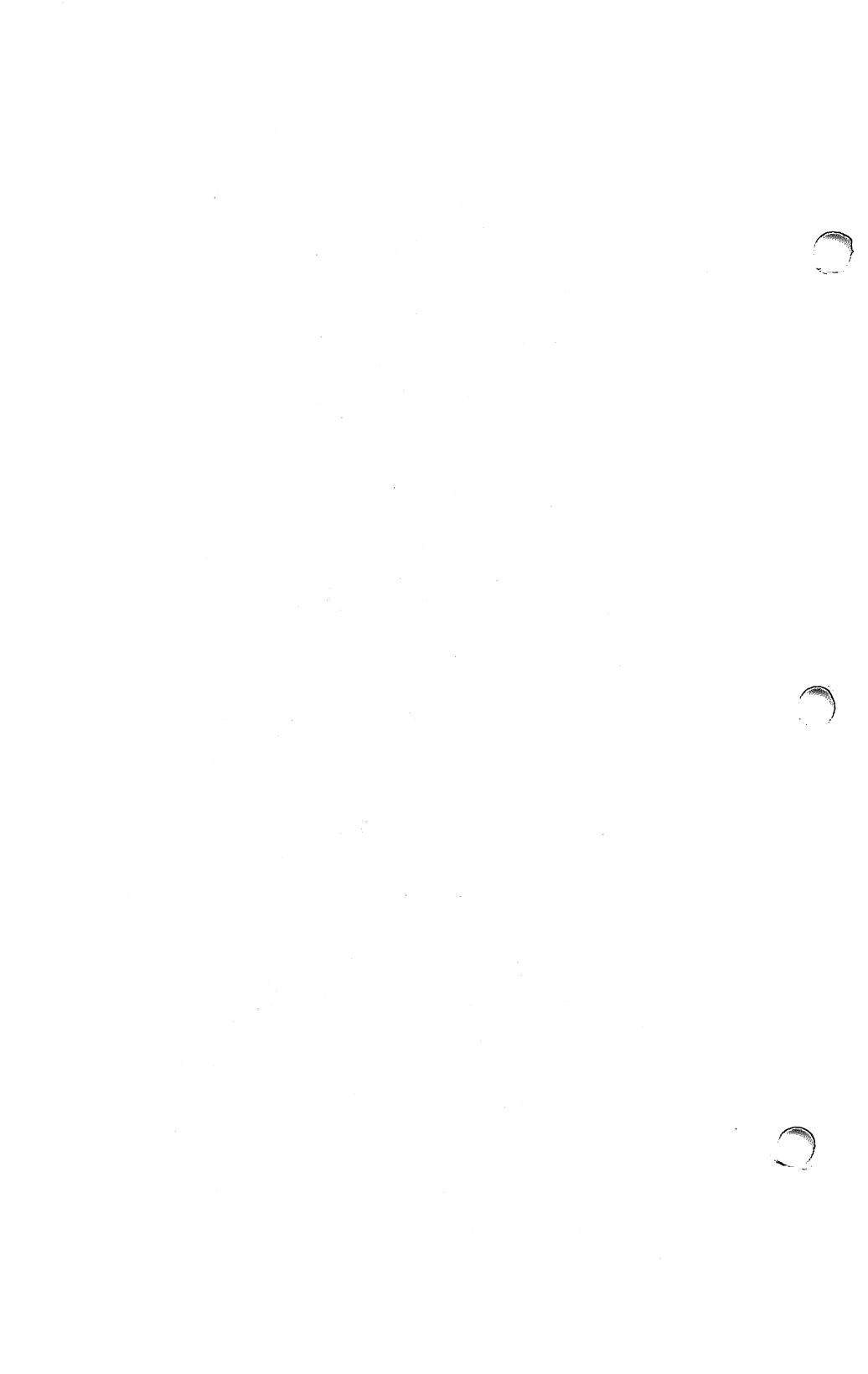
the search flag - if a search is called for, he takes the link-search information from the memo form and then calls a "FIND" command using it. The result is that once the ADDRESS window is opened, it automatically searches for the closest match to the data sent over in the link.

So far we've discussed links from DATABASE applications, but links from other sources still follow the same basic rules. As we've mentioned at the start, the only other link creator ("Active" module) is the HyperEditor. When you call up a link by double-clicking on a word in the editor, a very similar process happens as with the database.

The major difference between the two is the source of the information to fill in the link memo with. Under DATABASE, that information was entered when somebody first created the application in the Builder. Using the Item Information form, they entered all the link details at that time. With the EDITOR, all the same information is entered by using the TEXTLINK form; when you ALT-Click on a word in the HyperEditor, this form comes up and allows you to give all the details:

The screenshot shows a window titled "TEXTLINK" containing a "TEXT LINK BUILDER" form. The form includes a "LINK TYPE" dropdown menu with options: CMD, FORM, TEXT, .IMG, SND, and SPCH. Below this are fields for "LINK TO APP", "LINE #", "DFILE", and "INDEX". A "LOOKUP" dropdown menu shows options: LOOKUP, SPECIAL, and NONE. The "LOOKUP" field is currently set to "Hypertext".

The layout is a bit different, but the result is mostly the same. When you select a word, the EDITOR office looks to see if a TextLink entry exists for the word. If



it does, he goes and fills in a link memo form just the same as BASEBALL or ADDRESS did, and takes the details from the TextLink fields. Note that there is no "indirect" setting here; Searches are still allowed, however. When the link memo is filled in, it is sent to the shipping desk and out, exactly like before.

Another difference people may notice between a DATABASE link setup and the TEXTLINK form is that only the former offers you the "SEL ACTION" settings. Even though these don't directly affect how a link packet works, they deserve some explanation. The reason that the EDITOR doesn't give you this option is that the editor only has one way to "Select a Link Action", that is a link is started when you double-click (or CTRL-click) on a word. In a DATABASE application, you can be a little more versatile. By setting S-LINK, you have told HyperLINK to start a link if that object is Selected (i.e. single-clicked). DC-LINK, on the other hand, will only do a link if you Double Click on the object. START-LINK is a bit more unusual; it asks for the link to be done every time you open that form (this is rarely used). And "HIGHLITE" is a modifier of S-LINK and DC-LINK; if set, the button or object will "highlight" (reverse colour) when it is selected with the mouse. This is usually the way things work. Also note that you can have more than one Select Action activated at the same time.

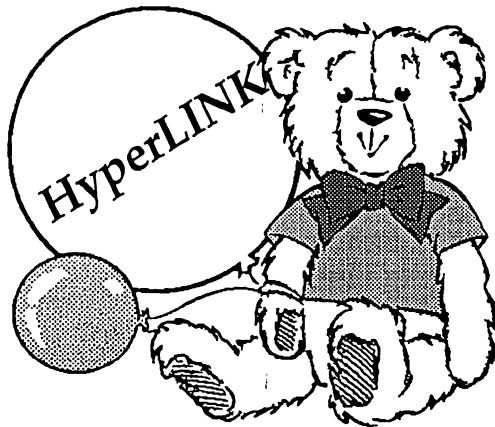
By the way, we should mention one other "company" that we have so far ignored. HyperLINK provides a built-in module called NUL, and this module functions almost exactly like the DATABASE module. The difference is that it does NOT create or manipulate any database files. The point of this is to allow creating a "control" type of application, one where there is not data input or output but other features, primarily linking, are still active. Consider a European map application: the main map screen has no data input or output, it only contains the map image and a number of buttons that link to country



information (another app). This type of form is a "NUL" based application, though it is still built with the Builder and operates like a database form does.

I hope this analogy has shed a little bit of light onto the subject of Linking within HyperLINK. It is one of the most powerful elements of the program, and linking can be as simple or as complicated as you choose to make it. Understanding the basic principles as described above can go a long way towards allowing you to create complex links between different applications and different objects.

For even further information regarding links, watch for "George's Favourite Recipes". Coming soon to a HyperLINK Addendum near you.



1. The first part of the document discusses the importance of maintaining accurate records of all transactions. It emphasizes that this is crucial for the company's financial health and for providing reliable information to stakeholders.

2. The second part of the document outlines the specific procedures for recording transactions. It details the steps from initial entry to final review, ensuring that all necessary information is captured and verified.

3. The third part of the document addresses the challenges associated with record-keeping. It identifies common pitfalls and provides strategies to avoid them, such as regular audits and clear communication between departments.

4. The fourth part of the document discusses the role of technology in modern record-keeping. It explores how digital tools can streamline the process and reduce the risk of errors, while also highlighting the importance of data security.

5. The final part of the document provides a summary of the key points and offers recommendations for future improvements. It encourages a continuous approach to refining the record-keeping process to meet the evolving needs of the organization.

HYPERLINK QUICK KEYS REFERENCE GUIDE

WITH NO HAP'S OPEN

- F1-F10 : These keys will open a HyperLINK Application in the corresponding position on the main menu.
- HELP : Calls up the Text Editor and loads in the default Help text.
- <CTRL> <Q> Quits program

WITH A HAP OPEN

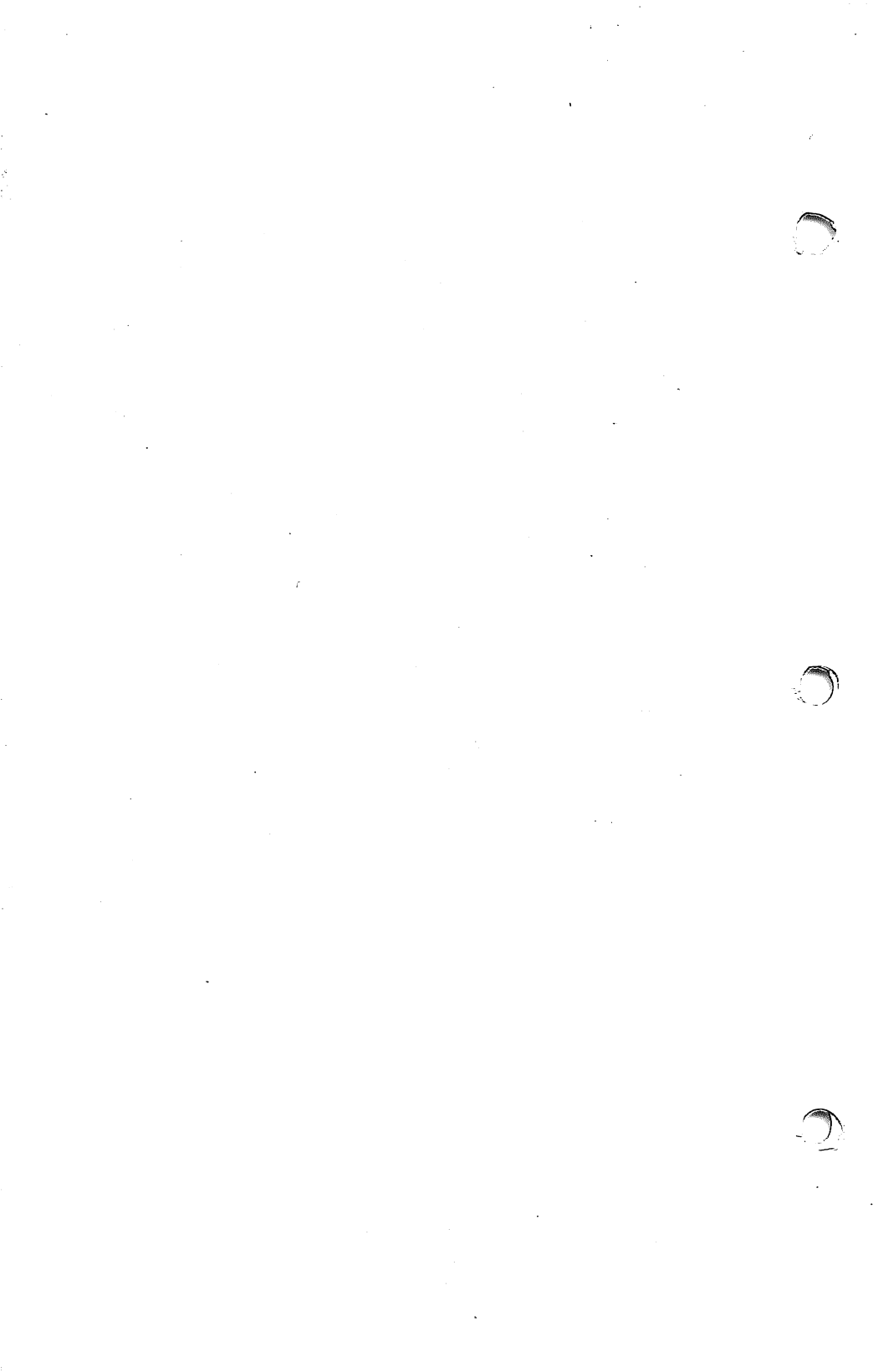
- F9 : Minimizes Active Window
- F10 : Closes Active Window

WITH A DATABASE HAP OPEN

- <ALT> <S> Saves current record
- <ALT> <E> Produces Empty record for new entry
- <ALT> <F> Find a Record Search Screen
- <ALT> First record (Beginning) of database
- <ALT> <L> Last record of database
- <ALT> <N> Next record in database
- <ALT> <P> Previous record
- <ALT> <I> Select Index (Sort File Name) to use
- <ALT> <D> Delete record
- <ALT> <C> Send record to clip board
- <ALT> <O> Send record to printer (Output)

WITH HYPER EDIT OPEN

- <ALT> <Click> Text Linker with word loaded
- <ALT> <Undo> Text Linker no word loaded
- <SH> <Undo> Loads Default Text File
- <Ctrl> <Click> Same as Double Click - Calls up Link
- <Undo> Loads in previous Text

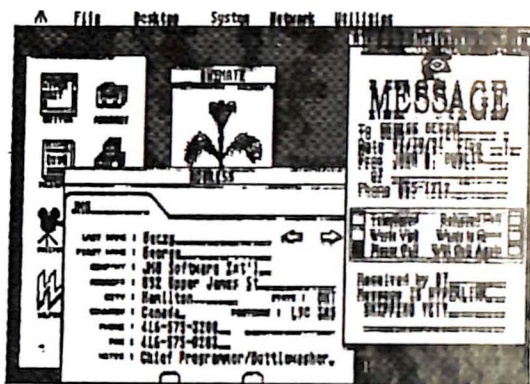




NOW SHIPPING!!!

HyperLINK

By Turning Point Solutions



With multi windowing capabilities, HyperLINK fulfills the "Cluttered Desktop" metaphor!

SYSTEM REQUIREMENTS :

- * ST/STE/MEGA STE/TT
- * Requires 1 Meg RAM
- * Color or Mono Monitor
- * Hard Drive Recommended
- * Comes on 2 DS Diskettes

\$149.00

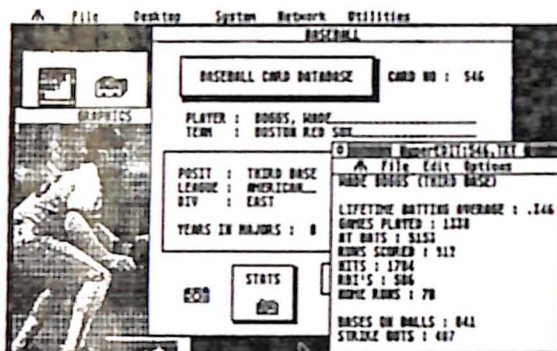
Price in U.S. Dollars

Limited Time Pricing!!! HyperLINK will be going up to \$199 US August 30th.

* Shipping Charges : Add \$5 for US and Canada, \$25 for Europe, Australia, Japan and elsewhere.

Dealer / Distributor Enquiries Welcome!

- * Hypertext Environment
- * Multi-Media Support
- * Networking Support
- * Customizable Linking
- * Custom Application Builder
- * Multiple HyperLINK Applications can be run at once!
- * Text / Graphics / Sound / Speech / Animation Modules all included
- * DBase Compatible File Structure
- * Includes ICON Library
- * Customizable Report Generator
- * Supports Monitorm Monitors
- * Custom Windowing (Allowing 7 Active Windows)



Create your own Applications quickly and simply!

JMG Software International, Inc.

892 Upper James Street
Hamilton, Ontario
CANADA L9C 3A5

Phone : (416) 575-3201

FAX : (416) 575-0283

BBS : (416) 389-9064

GENIE Address : JMGSOFT