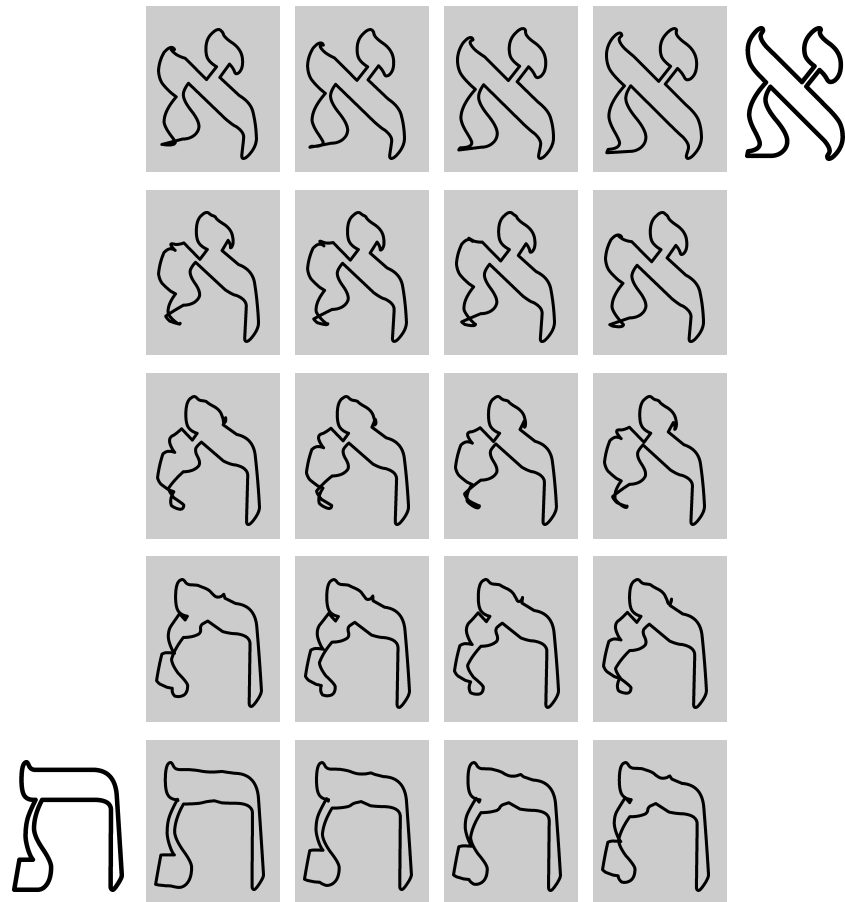


Makor



מְקוֹר

A System for Typesetting
Biblical and Modern Hebrew
with Omega and T_EX
Version 2.0, January 2003

Alan Hoenig
ahoenig@suffolk.lib.ny.us

Makor:
A System for Typesetting Biblical and Modern Hebrew
with Omega and T_EX

ALAN HOENIG
Department of Mathematics
John Jay College, City University of New York
 ahoenig@suffolk.lib.ny.us

January 17, 2003

CONTENTS

1	WHY USE מְקוֹר?	4
	<i>Earlier versions of מְקוֹר</i>	5
2	GETTING AND INSTALLING MAKOR	5
	<i>Installation</i>	5
	<i>Testing the installation</i>	7
	<i>Other files</i>	7
3	WHY USE T _E X?	8
	<i>The T_EX life cycle: a brief introduction</i>	9
	<i>Creating a simple Makor document with Lambda</i>	9
	<i>Creating a simple Makor document with Omega</i>	10
4	SOME MAKOR CONVENTIONS	11
	<i>Consonants</i>	11
	<i>Vowels</i>	12
	<i>More about ⚡</i>	13
	<i>Punctuation, dashes, foreign sounds, and so on</i>	13
	<i>Numbers</i>	14
	<i>Pattern breaking</i>	14
	<i>Indentation and white space around a paragraph</i>	14
	<i>Adjustments</i>	15
	<i>Fine tuning</i>	15
	<i>Cantorial trope</i>	16
	<i>Examples; irony</i>	18
5	FONTS	18
	<i>All the characters in a font</i>	18
	<i>Font selection</i>	19
	<i>A Makor reference card</i>	21
	<i>Missing bold fonts</i>	21

	<i>Old Hebrew</i>	24
	<i>Additional fonts</i>	24
	<i>Working with TrueType</i>	24
6	CONVENTIONS	26
	<i>Variations on the מקור convention</i>	26
	<i>Non-Makor conventions: ArabTeX, Biblia Hebraica Stuttgartensia</i>	27
	<i>Yiddish and Ladino</i>	28
	<i>Makor “dialects”</i>	29
	<i>Archaic systems: Old Hebrew; Palestinian Babylonian vocalization</i>	31
	<i>Standard conventions; rolling your own...</i>	32
7	HEBREW ALIGNMENTS AND TABLES	33
	<i>Protection</i>	34
	<i>Suppressing reversal</i>	35
8	VARIOUS AND SUNDRY	36
	<i>A trick or two</i>	36
	<i>Processing this document</i>	38
	<i>Caveat!</i>	39

LIST OF FIGURES

1	<i>Learning to read Hebrew.</i>	3
2	<i>A portion of the Book of Esther.</i>	6
3	<i>Fully vocalized Makor output (see file vocal.tex).</i>	9
4	<i>A page of Talmud.</i>	12
5	<i>Hebrew intermingled with English.</i>	16
6	<i>מקור keyboard conventions for cantorial trope.</i>	17
7	<i>A font table for Ezra.</i>	20
8	<i>The Siloam water tunnel inscription.</i>	21
9	<i>Available Makor fonts.</i>	22
10	<i>A Makor reference card.</i>	23
11	<i>A portion of Genesis using Hadassab types.</i>	25
12	<i>Yiddish input conventions.</i>	28
13	<i>The Yiddish Jabberwock!</i>	30
14	<i>A German dialect of Makor.</i>	31
15	<i>Archaic Hebrew vowel systems.</i>	31
16	<i>Standard Makor environments.</i>	32
17	<i>A bit of Esther: trope, but no vowels.</i>	33
18	<i>Sample Hebrew alignments.</i>	35
19	<i>Using a Hebrew table to display a verb conjugation.</i>	36
20	<i>The Siloam water tunnel inscription: enhanced display</i>	37
21	<i>A portion of Ecclesiastes.</i>	40

ABSTRACT

This manual describes the *Makor* (מָקוֹר) system for typesetting Hebrew by means of Omega, an extension of the T_EX typesetting language. *Makor* is appropriate for authors using English-language keyboards for entering Hebrew text in a convenient manner. *Makor* lets you enter vowels and cantorial diacritics, selects final letters by itself as appropriate, and has special conventions for entering numerals and tables. The software plus fonts is freely available from the Web.

All of the Hebrew text in this manual was produced using *Makor*. I produced all other displays using programs that are part of the T_EX or Omega distribution. I urge all potential authors to flip through to see if *Makor* would be suitable for you.

חגיגת הארוסין של חיותה

טביון ליברכט

אֶת חֲגִיגַת הָאָרוּסִין שֶׁל חַיּוֹתָה קִבְּעוּ בְּמַהִירוֹת, אַחֲרֵי
 2 שֶׁהֶצְעִירִים הוֹדִיעוּ עַל כָּךְ פְּתֹאֵם לַהוֹרִים.
 חֲמִישֶׁה-עֶשֶׂר יָמִים לִפְנֵי הַחֲגִיגָה, עֲדִין לֹא חָשַׁב אִישׁ
 4 עַל הַבוּשָׁה שֶׁעָלּוּל סָבָה מְנַדֵּל לְהִבְיָא עַל כּוֹלָם.
 כּוֹלָם הָיוּ עֲסוּקִים: עָרְכוּ רְשִׁימוֹת שֶׁל הָאוֹרְחִים,
 6 הַמְאָכְלִים, הַכְּלִים שֶׁיֵּשׁ לְהַכִּין, הַשְּׂכָנִים שֶׁמוֹכְנִים
 לְעִזּוֹר, הַמְתַּנּוֹת שֶׁחַיּוֹתָה מְבַקֶּשֶׁת לְקַבֵּל.
 8 בָּלָה, אִמָּה שֶׁל חַיּוֹתָה, הַבִּיטָה בְּבֵית וְנִבְּהָלָה.
 הַהֶתְמָם עַל בְּדֵי הַרְפּוּד נִרְאוּ לָהּ קוֹפְצִים לְעִין,
 10 הַוִּילוֹנוֹת כְּבֹר יִשְׁנִים, עַל נִיר הַקִּיר בְּחֹדֶר הָאוֹכֵל רוֹאִים
 אֶת הַסִּימָנִים שֶׁל הַכְּסֵאוֹת.
 12 וְכֵן נוֹסְפָה גַם רְשִׁימָה שֶׁל חֲנוּיוֹת לְבֵדֵי רְפּוּד
 וְיִלְבָּעֵלִי-מְקָצוּעַ.
 14 בְּתוֹךְ כָּל הַרְעֵשׁ נִזְכָּרָה בָּלָה שֶׁהִיא עֲדִין לֹא הִחְלִיטָה
 אִיזוֹ שְׂמָלָה תִּכְּוֶן לְעֲצָמָה לְחֲגִיגָה.

engagement ארוסין

shame בושה
liable, likely עלול
list רשימה

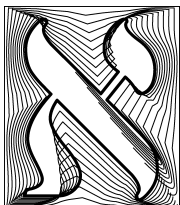
be frightened נבהל

stain כתם
upholstery בדֵי רפוד

professional בעל-מקצוע

FIGURE 1: Learning to read Hebrew. The simple appearance of this primer belies the difficulties involved. Interested readers should carefully go over the source for this figure (`primer.tex`).

1 WHY USE מקור?



If you need to include Hebrew text in a document—especially in combination with English or other left-to-right (LTR) language—you know what horrors you’re in for. Text entry itself can be challenging, and vowel placement adds to your worries. *Makor* attempts to eliminate these issues. So if you need to include Hebrew within a report or article, here’s why you should use *Makor*:

- It’s *easy* to enter consonants *and* vowels into a document.
- *Makor* automatically decides if a final form for a letter is necessary.
- *Makor* takes care to position vowel symbols properly with respect to different letters.
- There are easy ways of finagling these (and other) aspects—altering the placement of a vowel, forcing or suppressing the final form of a letter, and so on.
- Authors can decide whether to include vowels or not.
- Authors can enter cantorial diacritics (trope) into the text.
- *Makor* also supports Yiddish with a separate input convention and special Yiddish characters.
- Alternative typographic conventions are supported, such as ךּ instead of ך and ם for ך.
- Fonts include the special characters (שׁ ך and so on) so that it’s possible to typeset the Hebrew Bible with *Makor*.
- Numbers are entered normally, and *Makor* takes care to typeset them properly within the Hebrew (that is, the numbers are LTR even though the surrounding text is RTL).
- *Makor* knows the traditional Hebrew numbering system (see, for example, figure 2), and can number Biblical verses automatically.
- *Makor* understands the conventions of Arab_TE_X, so you can process Arab_TE_X Hebrew documents in *Makor*.
- *Makor* understands the conventions of BHS, so you can process Biblia Hebraica Stuttgartensia in *Makor*.
- One of *Makor*’s fonts allows scholars to typeset Old Hebrew.
- Authors and scholars can typeset using the archaic Palestinian or Babylonian vowel systems.

- When you revise the document—either by changing your text or altering layout parameters (such as the width of a column, or whatever)—these changes automatically propagate into your text.
- Layouts of arbitrary complexity are possible.
- All of the versatility that's part of T_EX and of Omega is always available to the author using this system.
- ...And much, much more!

Earlier versions of מקור

An earlier version of T_EX worked with any extended T_EX that provided for right-to-left (RTL) typesetting. The current version demands the use of Omega, and as a result, the current version of *Makor* is far more robust, and contains several additional capabilities.

I deprecate this former version and urge those users to switch to the current version as quickly as possible.

2 GETTING AND INSTALLING MAKOR



You can always download the latest version of this software from the Web; the appropriate site is www.ctan.org. I mention this site from time to time throughout this document, and it goes by the name of 'CTAN'. It contains lots of other software suitable for use with T_EX, L^AT_EX, and Omega, including several other packages for typesetting Hebrew, such as ArabT_EX (which has a Hebrew module), Hebrew L^AT_EX, and a worthy package called `cjhebrew`. Please make sure that you download all the files as one large compressed file, as this makes installation particularly easy. Depending upon your method of download, this file might be `makor.zip`, `makor.tar.gz`, `makor.tar.Z`, or something similar. For our discussion, I'll refer to it as `makor.zip`, though it might have one of these other names.

Incidentally, the full address for the latest version of *Makor* is

www.ctan.org/tex-archive/language/hebrew/makor

Installation

I take it as given that T_EX—including Omega, Lambda, and their friends—has been properly installed on your system. If not, consult your local T_EX wizard, or local guides available at CTAN. Another source is the T_EX Users Group (TUG; www.tug.org).

Make sure you know where the 'root' of your local T_EX directory is. Most likely, it has a name like '`...texmf/`'. Henceforth, I will call this the `<texmf-root>` directory.

LIBER ESTHER

3 א אחר ׀ הדברים האלה גדל המלך אחשורוש את־המן בן־המדתא האגני וינשאהו וישם את־כסאו מעל
כל־השרים אשר אתו: ב וכל־עבדי המלך אשר־בשער המלך פדעים ומשתחוים להמן כי־כן צוה־לו המלך
ומרדכי לא יכרע ולא ישתחנה: ג ויאמרו עבדי המלך אשר־בשער המלך למרדכי מדוע אתה עובר את
מצות המלך: ד ויהי [באמרם] פאמרם אליו יום ויום ולא שמע אליהם ונידו להמן לראות תעמדו דברי
מרדכי כי־הגיד להם אשר־הוא יהודי: ה וירא המן כי־אין מרדכי כרע ומשתחנה לו וימלא המן חמה:
ו ויבו בעיניו לשלח יד במרדכי לבדו כי־הגידו לו את־עם מרדכי ויבקש המן להשמיד את־כל־היהודים
אשר בכל־מלכות אחשורוש עם מרדכי: ז בחדש הראשון הוא־חדש ניסן בשנת שתים עשרה למלך
אחשורוש הפיל פור הוא הגדל לפני המן מיום ׀ ליום ומחדש לחדש שנים־עשר הוא־חדש אדר: ח
ח ויאמר המן למלך אחשורוש ישנו עם־אחד מפגז ומפרד בין העמים בכל מדינות מלכותך ודתיהם שנות
מכל־עם ואת־דתי המלך אינם עשים ולמלך אין־שנה להניחם: ט אם־על־המלך טוב יכתב לאבדם ועשרת
אלפים כפר־כסף אשכול על־ידי עשי המלאכה להביא אל־גנזי המלך: י ויסר המלך את־טבעתו מעל ידו
ויתנה להמן בן־המדתא האגני צרר היהודים: יא ויאמר המלך להמן תכסף נתין לך והעם לעשות בו
כטוב בעיניך: יב ויקראו ספרי המלך בחדש הראשון בשלושה עשר יום בו ויכתב ככל־אשר־צוה המן אל
אחשורפני־המלך ואל־הפחות אשר ׀ על־מדינה ומדינה ואל־שרי עם ועם מדינה ומדינה ככתבה ועם ועם
כל־שנו בשם המלך אחשורש נכתב ונחתם בטבעת המלך: יג ונשלוח ספרים ביד הרצים אל־כל־מדינות
המלך להשמיד להרג ולאבד את־כל־היהודים מנער ועד־זקן טף ונשים ביום אחד בשלושה עשר לחדש
שנים־עשר הוא־חדש אדר ושללם לבז: יד פתשגן הכרב להנתן דת בכל־מדינה ומדינה ולוי לכל־העמים
להיות עתדים ליום הזה: טו הרצים יצאו רחופים בדבר המלך והדת נתנה בשושן הבירה והמלך
והמן ישבו לשתות והעיר שושן נבוכה:

FIGURE 2: A portion of the Book of Esther, showing both cantorial and vocal diacritics. Notice too that מְקוֹר can handle both spoken (q're) and written (k'tab) variants of text. The verse numbering was done automatically. See the file `esther.tex` for additional details.

Place yourself in this `<texmf-root>` directory, and uncompress `makor.zip`, **making sure that you preserve directory structure** (this should be the default behavior).

Next, move to the `<texmf-root>/tex/makor2` directory. Place the contents of the file `psfonts.mkr2` into your file `psfonts.map`, which is located someplace like

```
<texmf-root>/dvips/base
```

Now, update the filename database (the procedure to do this is a key part of your T_EX implementation), and that's it!

Testing the installation

Within this same `<texmf-root>/tex/makor2` directory are some useful files. Enter the commands

```
omega mkrotest
```

and

```
lambda mkrltest
```

at the terminal. This exercises 'plain' Omega and Lambda, which is Omega+L^AT_EX. You should get two `.dvi` output files, which you can preview or print after you enter these commands:

```
odvips mkrotest
```

```
odvips mkrltest
```

I use GhostView for previewing and printing PostScript files. If you prefer to work with `.pdf` output, substitute the command 'dvi₂pdfm' for 'odvips'.

Other files

Most of the Makor2 package consists of font-related files. A few other useful utilities are present, however. For example, you can use the file `fonttbl.tex` to print all the characters of a standard computer font (not more than 256 characters). For further discussion, please see below, page 18.

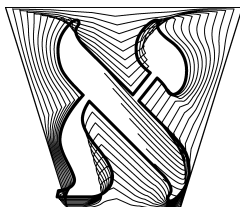
A somewhat similar file is `fontgrid.tex`, which prints out superlarge images of selected font characters superimposed on a grid. This file is useful for creating the specialized font metrics for a Hebrew font, and directions for its use appear within this file itself.

Also present is `refcard.tex`, which you can use to print a two-page summary of Makor2 and its conventions. See below, page 21, for further discussion.

The `makefonts` directory contains all the files that I used to create the virtual files that are part of *Makor2*. I include them for the sake of completeness, but strongly recommend that all but experts give this material a wide berth.

The remaining `eps` and `doc` directories contain the files you'll need to recreate this manual for yourself. See page 38 below for additional information.

3 WHY USE T_EX?



The point of T_EX (and of Omega) is to create documents whose appearance is of sufficient quality to qualify for the most demanding publishing requirements. T_EX automatically generates indexes, tables of contents, and automatically inserts other typesetting minutiae, such as ligatures, where they belong. T_EX was originally created to handle English-language typesetting, and it's easy to generalize this to work with any Latin-based language. Omega includes all the capabilities of T_EX, and adds a mechanism for handling non-Latin typesetting in a reasonably natural way. Although *Makor* works **only** with Omega, I will frequently refer to 'T_EX' or 'L^AT_EX' or 'Omega' even though I *only* mean 'Omega' in these contexts.

The Omega or T_EX life cycle is quite different from the usual Windows or Macintosh *gestalt*. If you're a typical computer user, you expect as a matter of almost divine right to type stuff at the keyboard and see the document composed and formatted before your eyes, onscreen. But there are reasons why this method may not work to your advantage; I can mention only a few here.

Optimal line breaking depends, for example, not only on what you've typed so far, but also on the words in the remainder of the paragraph. In extreme examples, the very last word of the paragraph can determine the linebreak of the first line. How then can any other program claim to give you world-class formatting if it composes paragraphs on the fly?

I expect my program to decide for me whether a medial or final form of a letter is needed. Yet, in a what-you-see-is-what-you-get program, it's difficult for the program to know when I've reached the end of a word. By the time you type a space or punctuation, a medial letter has already appeared on the terminal. WYSIWYG programs never backtrack to revise themselves.

Finally, although the menu selection system which accompanies typical commercial software offers legion choices, no set of menus can encompass all the formatting choices that an author might make. Therefore, anyone serious about the look of the written word must think seriously about the kind of programming interface—so that you can implement any format you need—that T_EX provides.

As with any sophisticated computer tool, advanced T_EX use can be challenging and rewarding. Nevertheless, it's easy to make T_EX (and *Makor*) do simple and straightforward things. Further information about T_EX appears below.

וַיִּמַת שֵׁם מֹשֶׁה עֶבֶד־יְהוָה בְּאֶרֶץ מִזְרַח מִדְבָּר מִצְרָיִם בְּיַד הַיְהוָה וְלֹא־יָדָע אִישׁ אֶת־קְבֻרָתוֹ עַד הַיּוֹם הַזֶּה׃ וּמֹשֶׁה בֶן־מֵאָה וָעֶשְׂרִים שָׁנָה בָּמָתוֹ לֹא־כָהָתָה עֵינָיו וְלֹא־נָס לָחָה׃ וַיִּכְבּוּ בְנֵי יִשְׂרָאֵל אֶת־מֹשֶׁה בְּעֶרְבַת מִזְרַח מִדְבָּר מִצְרָיִם יוֹם וַיִּתְּמוּ יָמָיו בְּכִי אֲבֵל מֹשֶׁה׃ וַיְהִי־שָׁעָ בֶן־נֹחַן מִלֵּא רֹחַ חֲכָמָה כִּי־סָמַךְ מֹשֶׁה אֶת־יָדָיו עָלָיו וַיִּשְׁמְעוּ אֱלֹהֵי בְנֵי־יִשְׂרָאֵל וַיַּעֲשׂוּ כַּאֲשֶׁר צִוָּה יְהוָה אֶת־מֹשֶׁה׃ וְלֹא־קָם נְבִיא עוֹד בְּיִשְׂרָאֵל כַּמֹּשֶׁה אֲשֶׁר יָדָעוּ יְהוָה יְהוָה אֱלֹהֵי יִשְׂרָאֵל לְכָל־הָאֲדָמָה וְהַמּוֹפְתִים אֲשֶׁר שָׁלַח יְהוָה לַעֲשׂוֹת בְּאֶרֶץ מִצְרָיִם לְפָרְעֹה וְלְכָל־עַבְדָּיו וְלְכָל־אֶרֶץ־מִצְרָיִם׃ וְלֹא־יָדָע הַיְהוָה וְלֹא־יָדָע הַיְהוָה וְלֹא־יָדָע הַיְהוָה׃

FIGURE 3: Fully vocalized Makor output (see file vocal.tex).

The TeX life cycle: a brief introduction

Creating a document—article, report, book, whatever—with TeX is quite different from using a commercial word processor or publishing program. You begin by preparing a strictly ASCII text file containing the text of your report, say, amongst which are interspersed various formatting commands.

Next, you run through the TeX program, as you run a file containing the text of a computer program through a program compiler. In TeX's case, the output is a file with the extension dvi, a device-independent file which contains very general instructions for rendering letters and other graphic elements on the printed page.

The dvi file is not something you can deal with directly, so we need one final step. Some subsequent program has to render the file. For example, I typically run my stuff through the excellent program dvips, which converts the .dvi file to a PostScript (.ps) file; actually, you need the Omega variant odvips.

In practice, it takes no time to implement these tasks, and they can be combined into batch or shell scripts to make the typesetting process as easy as possible.

Before using Makor, you'll need to install it (as with any software program). See above, section 2, page 5, for installation instructions and discussion.

Creating a simple Makor document with Lambda

For people who are curious, or perhaps just a bit unsure, here's how you use Makor to create a simple Lambda document incorporating both Hebrew and English, where 'Lambda' is the Omega version of 'TeX'.

Using a text editor, create a Makor TeX test file called mkr1test.tex (or something similar) that contains the following purely ASCII text.

```
\documentclass{article}
\usepackage{makor2}

\begin{document}
Hello, world! \[sh^aul0m, '0laum!\] Goodbye, world!
```

```
\end{document}
```

For those familiar with L^AT_EX, this document is largely familiar. The `document` section of the file contains text delimited by Hebrew typesetting switches `\[` and `\]`. The ASCII within these toggles will get translated and typeset into fully vocalized Hebrew by Omega.

To enable this process, issue the command

```
lambda mkrltest
```

from the DOS-prompt or its equivalent. If Omega and *Makor* are both properly installed, and if you haven't made any errors when you created your test file, then Omega will tell you that the file `mkrltest.dvi` is now present in the current directory. The typeset contents of this file cannot directly be viewed or printed without converting it to PostScript or Acrobat pdf format.

I prefer PostScript output, so I issue the next command in this process

```
odvips mkrltest
```

to create the file `mkrltest.ps`, which I can now view or print with a current version of GhostView. (All these programs, along with Omega and Lambda itself, should be present as a result of a proper installation of T_EX and Omega. You can also use GhostView to convert your `.ps` into the Acrobat `.pdf` format.) It will look something like this: Hello, world! **שְׁלוֹם, עוֹלָם!** Goodbye, world!

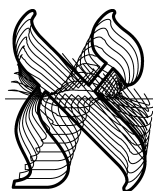
Any file you create for typesetting with Makor + Lambda should follow this schematic.

Creating a simple Makor document with Omega

For those of you who prefer to use plain Omega, the only change you need to make to the above instructions has to do with the form of the input *Makor* test file `mkrotest.tex`. In this case, the file has the simple form:

```
\input makor2
Hello, world! \[sh^aul0m, '01aum!\] Goodbye, world!
\bye
```

and you run the command `'omega mkrotest'` followed by `'odvips mkrotest'`.



If you know Hebrew, perhaps you will agree that the input required to produce the short display on the previous page is a good match for the Hebrew itself. Nothing special was done to get the final form for the מ's—*Makor* chose them automatically. Experienced T_EX-users know that short selection above relied upon *default values* of the system, and these can be changed many ways to alter the typeset appearance. In this document, we will explore the defaults built in to *Makor*, beginning with the conventions you need to produce Hebrew letters.

You get letters, vowels, and cantorial diacritics using the lowercase English consonants and vowels, and the special characters ' , ' , : , . , ^ , _ , | , * , and the uppercase O. Of course, punctuation and numerals are used in the usual ways as needed.

All the conventions we now discuss are summarized below in figure 10 on page 23. (This table includes alternative keyboard conventions as well.)

Consonants

Where possible, you get the Hebrew consonant that you expect from the corresponding English input. Thus, you type **b** to get ב, **f** to get פ, and so on. Some Hebrew consonants don't correspond to single Latin letters, so we use reasonable compromises. For example, type **sh** to get ש, and type **ts** and **ch** to get צ and כ. We use ' and ' to get נ and ז; these are long-standing conventions that apply to all Semitic languages. We use a prepended period to specify relatively rare consonants; thus, we get ט and ז from .s and .t.

Almost all Hebrew letters can occur in a dotted form. We pronounce some of these dotted forms differently, and their input is straightforward. We've seen above how to get ב already. We get כ and פ via t and p. At other times, we need the dot for grammatical purposes, even though this may not change the pronunciation. One way to get this dotted form is to append an asterisk, which visually approximates the appearance of a dot. Thus **h*** produces ח. But dots are also added for a grammatic reason—when they correspond to a doubled consonant. We can use this interpretation to make it easy to “dot” a letter—simply double it. So we get ש and כ from yy and dd, and so on.

Makor makes other conventions possible as well. For example, we use the circumflex ^ to get the upperdot on the ש. Thus, ש comes from either ^s or ^sh, and we get ש from shsh^.

At times, you may want to suppress the automatic generation of final letters, or conversely, force the typesetting of a final form. To this end, use the ^ and _ characters to force and suppress these forms. Thus, **gm_** yields the abbreviation for ‘gemara’: גמ. and you get the silly ככ from p^p . (Mnemonic: just as the ^ is suspended in mid-air, so too the final form of a letter is followed by ‘air’, the following word space.)

Sometimes, perhaps to fool *Makor* that you've reached the end of a word, you may

ארבעה אבות מיקין אבות קרי להגן דכמיין בקרא בהדיא. ובגמרא מפרש הי יהו חולדות: השור והעזר כו'. בסדר שהן כמזבין בפרשה סדרן במשנה. דפרשה ראשונה נאמרה בשור שניה צור: מבעה - מפרש בגמ' : העזר. כי תנא אש: לא הרי השור כהני

ארבעה אבות מיקין אבות דוכמא דלא מני הן כמו הך ובגמרא גבי שלש עשרה אבות מיקין ובארבעה מחוסרי כפרה (כרימות דף ט:) ואית דוכמא דקמיין הן דקמיין ארבעה שומרין הן (שטעוה ד' מטי') וארבעה ראשי שנים הן (כ"ה ד' ב' וטס) : (גליון : וא"ת אמאי לא קאמר

ארבעה

אַבוֹת גְּזִיקִין, הַשּׁוֹר וְהַבּוֹר וְהַמְּבַעַה וְהַקְּבֵר. לֹא הָרִי הַשּׁוֹר כְּהָרִי הַמְּבַעַה, וְלֹא הָרִי הַמְּבַעַה כְּהָרִי הַשּׁוֹר, וְלֹא זֶה וְזֶה שְׁנֵי בִּהְיוֹן רוּחַ חַיִּים, כְּהָרִי הָאֵשׁ שְׁאִין בּוֹ רוּחַ חַיִּים, וְלֹא זֶה וְזֶה שְׂדֵרְכָן לִיְלֶכֶד וְלִהְיוֹק. הַצַּד הַשְּׂהוּה שְׂבָהוּן שְׂדֵרְכָן לִהְיוֹק וְשִׁמְרֵתוֹן עֲלֵידָה, וְכִשְׁהוֹיִק חֵב הַמְּזִיק לְשֵׁלֶם תְּשִׁלּוּמֵי בְּזֶק בְּמִישַׁב הָאָרֶץ: גַּמְ' מְדַקְתָּנִי אֲבוֹת מְכַלְלֵי דַאכָא תּוֹלְדוֹת.

המבעה - כלומר אי כתב רחמנא לא נפק מצעה מיניה ואמעו להכי איצטרכו למיכתב ולהכי נקטו ברישא כהני המבעה ולא נקטו להי כסדר לא הרי השור כהרי הבור משום דמו לא הוי מני למימני לא זה וזה שיש בהן רוח חיים ועוד טעמא אחרינא דהא רבוחא אשמעי שאע"ג שיש לעיניהם רוח חיים לא נפיק חד מהכריה ובגמ' מפרש מאי לא הרי דקאמר: כהני האש שאין בו רוח חיים ואי לא תכביה רחמנא הו"ו חמייע ליעטר: לע זה וזה כו'. אלו שלשמן דרכן לילן ולהויק: ראש השנה כו' - מפרש בגמ' לאחויי מאי: במיעב הערין - מעידת נכסיו יגבה דמי הויק אס רוצה לפרוע לא קריקע: גמ' - טעמא - בשוגג: סקלה -

השור והבור - פירוש בקונטרס בסדר שכתבו בפרשה סדרן במשנה ואף על גב דלמ"ד תן שור לרגלו לא הוי כסדר הפרשה דרגל נפק לן מושלח את בעייה דתמיב כמר בזה מ"מ שם שור כמיב קודם בפרשה דהיינו נגייה דקין ולמ"ד מצעה זה אדם אע"ג דלכתר בעניה כמיב בפרשת אמור מנה בהמה אש למנה דהיינו

אדם דאויק שור לא חס לטתמו כסדר הפרשה לפי שרחוק כל כך ושנאו כסדר לא הרי דסיפא שמצעה קודם להבער: **לא** הרי השור כהרי המבעה - פירוש אין קולמו של שור וקולמו של המבעה כדמפרש לקמן בגמ' למ"ד תנא שור לקרנו ומבעה לשיני משום דשור כוונמו להויק ומבעה אין כוונמו להויק ולפיך אי כתב רחמנא שור לא אמי מצעה מינה שהיא קל מינה ואין פירושו כשאר מקומות שכתלמוד לא ראי זה דהמש פירושו אין חומרא של סה כחומרא של זה ולכן אין החומרות גורמות זה הדין אלא הדין השוה שבהן גורם הדין ושיניה כלן התלמוד פירושו מבשאר מקומות משום דהיכר התמור מחילה בלא זה חה שיש בהן רוח חיים:

ולא זה וזה שיש בהן רוח חיים כהני האש - גבי שור ומבעה לא הו"ו לפרש החומרא כי הכא משום דמד מהד קל למנוע חומר בחד מה שאין בתכורו זה דלא מני הך לא הרי האש כרי השור ומבעה דקמיני לעיל לא הרי המבעה כרי השור משום שלע היה יכול למנוע חומרא מה שאין בשניהם דאי משום דכתב אחר מעורב בו ואין הולך לדעתו כמו שצ"ל אין זה חומרא מדלא חשיב לה גבי חומר בהם משום דהא דלמיין לקמן (דף ג:) דף ג) אבות וסכינו מאי שניא אש דקין אחר מעורב בו ואין הולך לדעתו ה"ק מאי שניא אש שאע"ף שכתב אחר מעורב בו ראי להמחייב בו משום שהעו מונך ושמימנו כו' ובפיפא גבי צור הוה מני למימר לא הרי הבור שמתילת עשייתו למק

FIGURE 4: A page of Talmud.

use the 'unskip' command `<` or its synonym `\>` to add a word space, and to then remove it. Thus, `\[ts\<ts\]` is almost the same as `\[ts^ts\]` (ץץ).

Vowels

Makor takes great care with placement of vowels. Thus, although a vowel is centered more-or-less geometrically on a letter like **ב** (בּ בְּ בִּי בִּי) and so on), they are placed—properly—all the way to the right for **ר**: **רְ רִי רִי רִי** etc.

One or more English vowels map to a Hebrew vowel, and the vowel follows the consonant to which it belongs. Thus, to get **מַיִם**, type `mayim`. For some vowels, you need two letters, such as `ei` or `au` to give `ַ` or `ָ`. Some vowels have alternative encodings, and you

choose the one that seems most reasonable for you. For example, ae is an alternative to ei, and you might use this, because it seems somehow more fitting to get בְּנֵי יִשְׂרָאֵל from

b'neiy yi^ˆs'rau'ael

We use + as an alternative to au for the qamats vowel ָ, because + looks a bit like this vowel. We use the colon to help generate the *bataf*-vowels: ִֿ from h: e. The vertical bar helps yield the *meteg*-vowels: d l i yields ֶֿ, and so on.

Sharp-eyed readers will have noted that the right single quote is used to generate the schwa vowel, as well as the letter aleph. *Makor* takes care to distinguish between the two cases, and does a good job at doing the right thing. In case you need to enforce one or the other of the cases, again use ^ˆ and _ˆ. Thus, _ˆ' always yields ְ, whereas ^ˆ' always typesets א. (Mnemonic: both the ^ˆ symbol and the letter aleph are above the line, whereas the underscore and the shewa are at or below it.)

The sole use of uppercase in the *Makor* keyboard conventions occurs in vowel representations. Whereas o generates the defective cholam, you need O for the full cholam וֹ.

More about א

In certain conditions, a holamdot followed by א needs proper placement. This is done automatically; hence אֹב from bo'. In other situations, the holamdot should be on the preceding consonant. Thus, to get אֹב we type bo\/'aum, using the *pattern breaker* (see below). In these two words, look carefully at the positioning of the holamdot.

Punctuation, dashes, foreign sounds, and so on

By and large, Israeli punctuation matches that of the rest of the world. One difference from American grammar is the frequent occurrence of the sequence '?!' to indicate some level of amazed incredulity. Just for fun, I have included in my Hebrew fonts the 'interrobang' symbol †, which you get by typing either !? or ?!. Perhaps this symbol will now sweep the world of Hebrew typesetting by storm? See below if you want to typeset the usual sequence.

Type ** to get the 'newsheqel' symbol ₪

In the usual way, we type --- and to get em-dashes. If you type -, you get ¯, not quite a hyphen in the usual sense. If the font contains a hyphen, you get it by typing =.

In modern Hebrew, it's customary to use single quotes \ ' to indicate a foreign, non-Israeli sound. Thus, ts_e\ 'q and g\ 'oon'g' l generate צֶ'ק and גִּי'ונג'ל (pronounced 'check' and 'jungle'). *Makor* takes care of the case that these foreign sounds occur at a word ending. Hence, rabiynOvits' typesets רַבִּינֹבִיטִ'ן. Note well—\ ' does *not* work; type only ' in this case.

Numbers

The convention in Hebrew (and Arabic too!) is that ‘Arabic’ numerals are to be read in the usual left-to-read manner, despite the fact that surrounding Hebrew text is read in the opposite way. *Makor* understands numerals, and for simple cases, takes care of the directional switch automatically.

More complicated cases-involving decimal points and whatnot—will be treated more appropriately in an upcoming version of this software.

Pattern breaking

Omega works by examining patterns in the input and replacing various patterns by Hebrew characters according to rules that are part of the *Makor* system. Occasionally, we need to break them. For example, we may want the sequence !? or ** instead of ? or **!**. To this end, we use the command `\/`. This command sort of looks like a wedge facing down, and you can think of it as a wedge that breaks up the pattern. Thus, to get תת type `t\/s`.

Indentation and white space around a paragraph

Because of the inherent left-to-rightness of Omega, you have to be the tiniest bit alert about white space and paragraph indentation. For example, if you type

```
\[laukhaen chakoo-liy n'um-ydwd l'yOm qoomiy lau'ad kiy  
mish'pau.tiy l|e':e.sof gOyim l'qauv'tsiy mam'laukhOs lish'pokh'  
' :alaeYhem za'miy kol ch:arOn 'apiy kiy b''eish^ qin'ausiy  
t|''+khael k+l-h+'|+rets.\]
```

you get:

לְכֹן חֲכוּרֵי נְאֻם־יְרוּד לְיוֹם קוּמִי לְעַד כִּי מִשְׁפָּטִי לְאַסֶּף גּוֹיִם לְקַבְּצֵי מַמְלָכוֹת לְשַׁפֵּךְ
עֲלֵיהֶם זַעֲמִי כֹל חֲרוֹן אַפִּי כִּי בְּאֵשׁ קִנְאָתִי תֹאכַל כָּל־הָאָרֶץ.

This is no good for two reasons: (1) the first line is indented improperly; and (2) the last line is ‘filled’ with white space on the right side.¹

Simple Makor commands remedy both deficiencies. First off, you can suppress the improper left indentation with the usual `\noindent` command. If you terminate the Hebrew paragraph with `\HPAR`, you get proper paragraph filling, so that

```
\noindent\[laukhaen chakoo-liy ... k+l-h+'|+rets:\HPAR\]
```

yields

¹Incidentally, this text, which is drawn from the book of Zephania, 3:8 (that is, **זְפַנְיָה ג׳ ח׳**), is a good test of Hebrew typesetting. It contains all of the letters of the Hebrew alphabet, including the final forms in addition. Thus, this verse is the Hebrew equivalent of “the quick brown fox ...”

לְכֹן חֲפוּלֵי נְאֻם־דּוֹד לְיוֹם קוּמֵי לְעַד כִּי מִשְׁפָּטֵי לְאַסֶּף גּוֹיִם לְקַבְּצֵי מַמְלָכוֹת לְשַׁפֵּךְ עֲלֵיהֶם
וְעָמִי כָל חֲרוֹן אַפִּי כִּי בְּאֵשׁ קִנְאָתִי תֹאכַל כָּל־הָאָרֶץ.

However, to enforce a Hebrew indentation on the right, use the command `\HINDENT`. Also, the command `\CENTERLASTLINE` is a variant of `\HPAR`, and will (as you would imagine) center the last line of the paragraph (something I can't quite figure how to do in standard TeX typesetting). Thus,

```
\noindent\[\HINDENT laukhaen ... k+l-h+'|+rets:\CENTERLASTLINE\]
```

yields

לְכֹן חֲפוּלֵי נְאֻם־דּוֹד לְיוֹם קוּמֵי לְעַד כִּי מִשְׁפָּטֵי לְאַסֶּף גּוֹיִם לְקַבְּצֵי מַמְלָכוֹת לְשַׁפֵּךְ
עֲלֵיהֶם וְעָמִי כָל חֲרוֹן אַפִּי כִּי בְּאֵשׁ קִנְאָתִי תֹאכַל כָּל־הָאָרֶץ.

which is common convention in some Hebrew liturgical typesetting.

Adjustments

From time to time, a vowel may not appear exactly where you'd like it. TeX, of course, contains a plethora of commands which allow for precise placement of typographic elements, but these may be awkward to use in the context of setting Hebrew. For this reason, you can embed adjustment commands—finagles, if you will—to fix things up. **All these fine tunings apply only to the next consonant+vowel pair**; usual metrics are restored thereafter.

The first kind of adjustment will simply move the next vowel (and this vowel only!) a fraction of the character width to the left or right. You enter this adjustment by surrounding the decimal fraction with a pair of underscores.

For example, we normally type `rau` to get רַ, in which the qamats is properly centered under the tail of the ר. For some reason, you may wish to place the vowel at the center of the character instead. If you type `_.5_aurau`, you get רְרַ, which also emphasizes that the adjustment only lasts for one character.

Fine tuning

We use the expression *fine tuning* to refer to a more careful adjustment, where you want to carefully specify position changes in both horizontal and vertical directions. If you surround a pair of valid, comma-separated TeX dimensions by underscores, then *Makor* uses this set of dimensions as an offset to the original position. For example, the silly `_.5pt, -3.5pt_aurau` gives rise to רְרַ. Note the convention of signs used in fine tunings.

Rabbinic Hebrew (RH) does not differ greatly from Biblical Hebrew (BH) in its inflection of the noun, although the neutralization of final *mem* and *nun* means that the masculine plural is often, as in Aramaic, יָ. Apart from the more frequent use of the archaic feminine suffix ת־ as in פְּהֶגֶת *‘priest’s wife’* and אֵלְמָת *‘dumb woman’*, RH also employs the suffixes יִת and וִת for example אַרְמִית *‘Aramaic’* and עֲבָדוֹת *‘servitude’*. RH developed distinctive feminine plural suffixes in אֹת־ (Babylonian) or יוֹת־ (Palestinian), for example מְרַחֲצוֹת/מְרַחֲצוֹת *‘bath-houses’* and יוֹת־, as in מְלְכוֹת *‘kingdoms’* for BH מְלָכוֹת, for nouns ending in וִת in the singular. Masculine plural forms sometimes differ from those that would be expected, or are normally found, in BH, for example, נִזְקִין from נִזֶק *‘damage’*, שְׁנָרִים from שׁוֹר *‘ox’*, שְׁוֹקִים from שׁוּק *‘market’*, צְדָרִים from צֶדֶק *‘side’*, חֲצָאִין from חֲצַי *‘half’*, and שְׁלוֹחִין from שְׁלִיחַ *‘envoy’*. The same is true of feminine nouns, for example אֲוִתִּיּוֹת from אוֹת *‘letter (of alphabet)’*, בְּרִיתוֹת from בְּרִית *‘covenant (without plural in BH)’*, and אִמָּהוֹת from אִם *‘mother’*.

Some masculine nouns take the feminine plural suffix וֹת, for example, חַנּוּת from חָן *‘favour’*, כְּלָלוֹת from כָּלֵל *‘rule’*, תִּינוּקוֹת from תִּינוּק *‘baby’*, תְּיָלוֹת from תָּיִל *‘army’*, עִירוֹת from עִיר *‘city’*, and מִימּוֹת from מַיִם *‘water’*. Similarly, there are some feminine nouns which take the masculine plural suffix יִם—יוֹנִים from יוֹנָה *‘dove’*, נְמָלִים from נְמָלָה *‘ant’*, and בֵּיצִים from בֵּיצָה *‘egg’*, for example. Occasionally, both types of plural are evidenced, as with יָמִים/יָמִוֹת from יוֹם *‘day’* or שָׁנִים/שָׁנוֹת from שָׁנָה *‘year’*, with each form having a slightly different shade of meaning and the ‘feminine’ variant only used with suffixes. In RH we sometimes find plurals of nouns only attested in the singular in BH, for example אֲבָרִים from אֲבָר *‘limb’*, דְּשָׁאִין from דֶּשֶׁא *‘grass’*, and תְּמִדִּים from תְּמִיד *‘daily sacrifice’*. Likewise, there are singular forms of nouns only attested in the plural in BH, for example אֲלִמוֹנוֹ *‘coral-wood’*, בֵּיצָה *‘egg’*, and בְּצָל *‘onion’*. The dual is used more than in BH, with existing forms retained and new ones created, for example מְסַפְרִים *‘scissors’* and בְּנֵתִים *‘meanwhile’*. (1993: A. S’aenz-Badillos, *A History of the Hebrew Language*, Cambridge University Press, pp. 188-89.)

FIGURE 5: Hebrew intermingled with English.

Cantorial trope

I found it tough to think of a convincing entry convention for the cantorial diacritics. Each of these symbols has a Unicode name, though, and *Makor* accepts this name, surrounded by square brackets, to stand for the trope. Thus, we get בֶּ from be [etnahta]. You may feel this is a lot of typing, but you are free to define command shortcuts. Thus, after

```
\def\8{[etnahta]}
```

we get the same result by typing \ke\8. Of course, you can (and should!) define command names that are more meaningful to you.

If you don't like the names that *Makor* uses for these accents, complain to the Unicode naming committee. Figure 6 lists these names and the associated trope symbols.

Metafont is a sibling program of T_EX that can be used for high quality font design. I used this utility to create the trope characters, which may explain their lack of quality. If any reader knows of a better trope font that could be used instead, I'd be grateful for the tip.

ה̣	h[etnahta]	ה̣ ^o	h[telishagedola]
ה̣ [˙]	h[segol]	ה̣ ^r	h[pazer]
ה̣ [˚]	h[shalsholet]	ה̣	h[munah]
ה̣ [˙]	h[zaqefqatan]	ה̣ [˙]	h[mahapakh]
ה̣ [˙]	h[zaqefgadol]	ה̣	h[merkha]
ה̣	h[tipeha]	ה̣	h[merkhakefula]
ה̣ [˙]	h[revia]	ה̣ [˙]	h[darga]
ה̣ [˚]	h[zarqa]	ה̣ [˙]	h[qadma]
ה̣	h[pashta]	ה̣ ^o	h[telishaqetana]
ה̣ [˙]	h[yetiv]	ה̣ [˙]	h[yerahbenyomo]
ה̣	h[tevir]	ה̣ [˙]	h[ole]
ה̣	h[geresh]	ה̣ [˙]	h[iluy]
ה̣ ^r	h[gereshmuqdam]	ה̣	h[dehi]
ה̣ ^r	h[gershayim]	ה̣ [˙]	h[zinor]
ה̣ ^{oo}	h[qarneypara]	ה̣ ^o	h[masoracircle]

FIGURE 6: מקור keyboard conventions for cantorial trope.

Makor makes it possible to fine tune these accents as well. If a single accent appears above a letter, the syntax is as before, except you enclose the correcting factor by circumflexes instead of underscores. Thus,

```
^1^r[shalsholet]
```

yields ה̣[˚] and

```
^3pt,1.5pt^r[shalsholet]
```

typesets ה̣[˚]; normally, you get ה̣[˚] from r[shalsholet].

Under a letter, it's possible to have *two* accents, and you can specify a semicolon-delimited second pair of coordinates to indicate an additional amount of offset between

the accents. Here's an example to illustrate the meaning of this gobbledegook. Normally, if you type `zau [etnahta]` you get $\overset{\underset{\cdot}{\cdot}}{\underset{\cdot}{\cdot}}{zau}$. Possessed by a spirit of anarchy, you could enter `_1pt, 1pt; 3pt, 2pt_zau [etnahta]`; this produces $\overset{\underset{\cdot}{\cdot}}{\underset{\cdot}{\cdot}}{zau}$. That is, the dimensions in the first pair offsets the vowel-trope pair, while the second pair adjusts the spacing between the two diacritics.

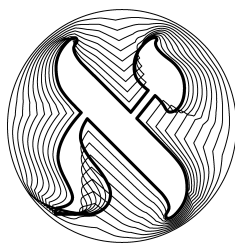
I find myself rarely needing any of these adjustments. If you find yourself entering more than one every few pages, it may mean that the font as a whole needs fine tuning. This is not difficult, but the procedure is out of place in this document. Please contact me if you feel there is a problem with one of the *Makor* fonts.

Examples; irony

The sources for all the examples in this document are part of the *Makor* distribution. Please refer to them frequently for guidance in using the *Makor* conventions.

[Ironically, the name in English for this software—*Makor*—des not follow the above rules. If it did, it would be `mauq0r`! Sorry for this inconsistency, but it's too late now.]

5 FONTS



Everybody loves fonts, and the fonts that are part of *Makor* provide a feast for the eyes. Dealing with fonts is a two-pronged operation, involving selection of a particular font and access to all the characters of the font. Incidentally, all of *Makor's* fonts are free (or some variant thereof), and I am continually seeking additional *big quality* fonts to add to the package. I'd be grateful if any readers with suggestions for additional fonts could get in touch with me. (Thanks!)

All the characters in a font

The vast majority of a the characters in a font are accessible by means of *Makor's* keyboard conventions. As we've seen, if you want $\overset{\cdot}{\underset{\cdot}{\cdot}}{l}$, you type `l`, and so on. Nevertheless, it's possible that a font contains additional characters that may not be so readily accessible. For that reason, the *Makor* package includes a file called `fonttbl.tex`, which allows you to print out all characters in a file. Within the first half-dozen lines, there is a line like

```
\def\myfont{ezra2 at 12pt}
```

and you should replace the font name and font size as appropriate. a quick list of the fonts that are part of *Makor* forms figure 9. (Note that there are nine gray fonts, named `ezrag12` through `ezrag92`; `g=gray`.) The `fonttbl` document doesn't depend upon Omega, so you can compile it with \TeX ; that is, first issue the command

```
tex fonttbl
```

and then convert it to PostScript or pdf via either of the commands

```
dvips fonttbl
```

or

```
dvipdfm fonttbl
```

Figure 7 displays the font table for the standard Ezra font. The first line of the table, the first sixteen characters of the font, are reserved for future use. Every font contains some special, perhaps even oddball, characters, and *Makor* sticks them into the second row. The last row is empty, and the remaining slots are filled with the characters of a *Makor* font.

Makor provides a `\PrintChar` character to access some of the more esoteric characters, but this command takes the character number as the argument, so it's useful to print out this table and retain it for future reference. For example, since the reflected nun appears in position 136, you could print it via `\PrintChar{136}`. Of course, this is somewhat clumsy. A better approach would be to create a command

```
\def\reflectednun{\PrintChar{136}} %% Omega (plain TeX) syntax
```

or

```
\newcommand\reflectednun{\PrintChar{136}} %% Lambda (LaTeX) syntax
```

Font selection

Makor expects an author to select fonts one by one. In the absence of instructions to the contrary, the Ezra normal font is automatically selected for Hebrew typesetting; this is the font used up to this point in this manual.

Using a font involves two distinct instructions. The first simply makes the font visible to *Makor*. That is, it associates a name that you create with a particular font at a particular size. Thus, after

```
\hfontdef{RSH}{rashi}{10pt}
```

the name RSH conjures up—to *Makor*—a 10 pt version of Rashi script. (`\hfontdef=Hebrew font definition`.) To invoke this font, you need to include the command

```
\hfont{RSH}
```

by your text. (`\hfont=use Hebrew font`.) As with any scope command in any version of T_EX, such a command obeys T_EX's grouping laws. The `\hfont` should follow the `\[` command.

Figure 9 displays a bit of all the currently available *Makor* fonts. The leftmost field in each line shows a sample of a font, while the middle field give the *Makor* name, which is the name you use in a `\hfontdef` or `\hfont` command. The final field gives a human name for the font, along with the design size of the sample. As you can see, there is a huge visual variation in fonts at the same size. Notice too that all modern fonts have *Makor* names ending in '2', indicating their suitability for use with *Makor2*.

Font Table for font ezra2 at 12pt

Produced on December 12, 2002 at 17:43.

0	Γ	1	Δ	2	Θ	3	Α	4	Ξ	5	Π	6	Σ	7	Τ	8	Φ	9	Ψ	10	Ω	11	↑	12	↓	13	'	14	ı	15	ı
16	ı	17	ı	18	ı	19	ı	20	ı	21	ı	22	ı	23	ı	24	ı	25	ı	26	ı	27	ı	28	ı	29	ı	30	ı	31	ı
32	ı	!	"	#	\$	%	&	')	(*	+	,	-	.	/															
48	0	1	2	3	4	5	6	7	8	9	:	:	<	=	>	?															
64	@
80		.	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
96	א	ב	ג	ד	ה	ו	ז	ח	ט	י	כ	ל	מ	נ	ס	ע	פ	ק	ר	ש	ת	י	י	י	י	י	י	י	י	י	י
112	ן	ס	ע	פ	ק	ר	ש	ת	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	
128	ן	ס	ע	פ	ק	ר	ש	ת	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י
144	א	ב	ג	ד	ה	ו	ז	ח	ט	י	כ	ל	מ	נ	ס	ע	פ	ק	ר	ש	ת	י	י	י	י	י	י	י	י	י	י
160	ן	ס	ע	פ	ק	ר	ש	ת	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י	י
176	ש	ש	ד	ד	ל	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר
192	ש	ש	ד	ד	ל	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר	ר
208	□	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א
224	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א
240	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א	א

FIGURE 7: A font table for the default Ezra font.

· ΔΥΟΘ· ἀδφςα· ρδΔ· ἀζα· ἀςΥ· ἀδφςα[· ψΧ]
· Χψκ· WΛW· ΔΥΟΘΥ· ΥΟΘ· Λκ· Wκ· ςςρρρ ρ ρ ρ
· ἀρς· ΧΖα· ΖΥΥΟΘ· Λκ· κρρ· Wκ· Λρ· · [δφ]ςαΛ
· ψδρρρ· ΥΥα· ἀδφςα· ψΖδΥ· ρ ρ ρ ρ · ςψΖψ· ρρδ
· ψΖψα· ΥΥΛΖΥ· ςςρρ· ΛΟ· ςςρρ· ΥΟΘ· ΧρρΛ· Wκ
· ἀψκ· ςΛκΥ· ψΖΧκψδ· ςΥρρρ· Λκ· κρΥψρ· ςψ
[ψ]δρρρ· Wκρ· ΛΟ· ρρρ· ςρρ· ἀζα· ἀψκ· Χ[κ]ψΥ

FIGURE 8: The Siloam water tunnel inscription; see `siloam.tex` for details.

A Makor reference card

Figure 10 shows a portion of a reference card which summarizes all the *Makor* coding and usage conventions. You can generate and print a more extensive such card if you wish. To do so, run the file `refcard.tex` through Omega and `odvips`, and print it on a letter-size sheet or card. Before doing so, you may have to (or you may wish to) change some of the fonts used in the reference card. The initial part of this file suggests the means for doing this.

Missing bold fonts

If you look closely at figure 9, you’ll notice that not every variant for every font is present. In particular, it’s often difficult to create a decent bold-faced font. For that reason, *Makor* provides two related macros that might be useful in this regard. Following the discussion in *The T_EXbook* pertaining to “poor man’s bold,” *Makor* provides commands `\pmb` and `\PMB` to embolden their arguments. They work better for smaller and larger fonts respectively. So if you type

```
'vgdh \pmb{wzch.ty} khlmn.s
```

you get אבגדה וזחטי כלמנס. These macros work by placing several copies of the text slightly offset from each other, so the results are a tad **fuzzy** or even **fuzzy** (I used `\pmb` and `\PMB` here), but it may be better than nothing. Note that it is *not* possible to typeset extended texts this way, since it is not possible to typeset text which breaks at lines with these macros. (If the look of this bold is sufficient for your needs, and you need extended passages in poor man’s bold, you’ll need to create a suitable virtual font for this purpose.)

אבגדהזחטי כך למם נן סעפף זץ קרשת	ezra2	Ezra Medium (at 12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrai2	Ezra Italic (Oblique; 12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag12	Ezra Grey1 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag22	Ezra Grey2 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag32	Ezra Grey3 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag42	Ezra Grey4 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag52	Ezra Grey5 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag62	Ezra Grey6 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag72	Ezra Grey7 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag82	Ezra Grey8 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	ezrag92	Ezra Grey9 (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	osh2	OmegaSerifHebrew (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	oshi2	OmegaSerifHebrew Italic (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	hasida2	Hasida Monowidth (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	hasidai2	Hasida Monowidth Italic (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	frankruehl12	Frank Ruehl (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	frankruehli12	Frank Ruehl Italic (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	frankruehlb12	Frank Ruehl Bold (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	frankruehlbi12	Frank Ruehl Bold Italic (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	nachlight2	Nachlieli Light (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	nachlighti2	Nachlieli Light Italic (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	nachbold2	Nachlieli Bold (12 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	nachboldi2	Nachlieli Bold Italic
אבגדהזחטי כך למם נן סעפף זץ קרשת	rashi2	Rashi Script (at 9 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	rashii2	Rashi Italic (9 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	rashib2	Rashi Bold (9 pt)
אבגדהזחטי כך למם נן סעפף זץ קרשת	rashibi2	Rashi Bold Italic (9 pt)
XW9 9270 94LY Z⊗HΣY <14K	oldheb	Old Hebrew (at 14 pt)

FIGURE 9: The Hebrew alphabet in all of the currently available Makor fonts.

Consonants	Star Convention	‘Doubles’	Basic Vowels	More Vowels
כ v ג g ד d ה h ו w ז z ח ch ט .t י y כ kh ל l מ m נ n ס .s ע ‘ פ f צ ts ק q ר r ש sh ש sh^ ש ^s, ^sh ת s ת th	א ’* ב v* ג g* ד d* ה h* ו w* ז z* ח ch* ט .t* י y* כ kh* ל l* מ m* נ n* ס .s* ע ’* פ f* צ ts* ק q* ר r* ש sh* ש sh^* ש ^s* ת th* ת s*	א ’^’ ב vv ג gg ד dd ה hh ו ww ז zz ח chch ט .t.t י yy כ khkh ל ll מ mm נ nn ס .s.s ע ‘ ‘ פ ff צ tsts ק qq ר rr ש shsh ש shsh^ ש ^shsh ת thth ת ss	א ’ae ב vei ג g’ ד da ה he ו wi ז zu ח cho ט .tau י .t+ ו oo י w* י 0	י y:a כ kh:e ל l:+ מ m נ n " ס .s i ע ’ a פ file צ ts + ק ql’
Conventions	Special Dagesh	Defective Cholam	End of Word	Commands
char^ force final letter char_ suppress final ^’ forces א _’ forces,	ב b כ k פ p ת t	אַ bo’ אָם bo\’ aum	Makor automatically selects these letterforms at word endings. End of Word אַ kh אַ k אַ m אַ mm, m* אַ n אַ nn, n* אַ f אַ p אַ ts אַ tsts, ts* אַ ts’	\[start Heb. typesetting \] finish Hebrew \ pattern breaker \noindent suppresses indentation \HINDENT enforces a Hebrew indentation \HPAR terminates a Hebrew paragraph \CENTERLASTLINE centers last line of paragraph \< unskip command \> synonym for unskip \fontdef{HH}{ezra2}{9pt} defines a Hebrew font for use \font{HH} uses the named font \PrintChar{242} identifies a character by slot, and prints it \MakorEnvironment{MKR} selects typesetting environment
Cantorial Accents	Miscellanea	Adjusting Vowel Placement		
ה h[etnahta] ה h[segol] ה h[shalsholet] ה h[zaqefqatan] ה h[zaqefgadol] ה h[tipeha] ה h[revia] ה h[zarqa] ה h[pashta] ה h[yetiv] ה h[tevir] ה h[geresh] ה h[gereshmuqdam] ה h[gershayim] ה h[qarneypara]	ה h[telishagedola] ה h[pazer] ה h[munah] ה h[mahapakh] ה h[merkha] ה h[merkhakefula] ה h[darga] ה h[qadma] ה h[telishaqetana] ה h[yerahbenyomo] ה h[ole] ה h[iluy] ה h[dehi] ה h[zinor] ה h[masoracircle]	- - - = --- " " ‘ , אַ ts’ ת123א ’123th ? !?, ?! אַ **	_ .4_rau moves the au vowel 40% of the width of the glyph starting at the left and moving to the right. _3.5pt,-3.5pt_rau displaces the au vowel 3.5 pt to the left of where it should be and 3.3 pt up from where it should be. ^ .9^r[shalsholet] puts the upper cantorial accent 90% of the way to the right of the left edge. ^3pt,1.5pt^r[shalsholet] displaces the upper accent by 3 pt to the left and 1.5 pt up. _1pt,1pt;3pt,3pt_zau[etnahta] to displace the pair of vowels and the increase the offset between them.	

FIGURE 10: Portion of a *Makor* reference card. The full card fits on two sides of a letter-size sheet. You get the Hebrew character in the left column by typing the keystrokes in the right.

Old Hebrew

The Old Hebrew font is far sparser than other fonts, so you should be more careful about how you use it. It is another Metafont hack by the current author, and readers knowing of a better font should get in touch with me.

Figure 8 shows the famous Siloam inscription rendered in *Makor's* Old Hebrew glyphs.

Additional fonts

It's a straightforward (albeit a trifle tedious and time-consuming) to add additional fonts to the *Makor* system. As an example, I adapted the beautiful MasterFont Hadassah fonts (available from Studio Rosenberg, 159 Yigal Alon St., Tel Aviv 67443, Israel; סטודייו ישראלי (רוזנברג, רח' יגאל אלון 159, תל אביב 67443, ישראל). Unfortunately, as they are proprietary, they cannot be distributed with this package, but see figure 11 to see how they look.

Working with TrueType

These MasterFont fonts arrived in TrueType format, which are not readily understood by all variants of T_EX. (However, T_EX's sibling pdfT_EX does, for example.) Here are some tips for working with TrueType fonts. This information may be of use, as the TrueType format is becoming more and more common.

In many cases, I use the superb `ttf_edit` utility to massage the fonts into ones better suitable for use by *Makor*. (This program's author, Richard Kinch at `kinch@truetex.com` can help you get hold of this program.) Then, I used the invaluable `TrueTypeToType42` utility by Thomas Barton (available from the CTAN archives) to create PostScript files that `dvips` recognizes. Another alternative is the `ttf2mf` utility, a means for converting TrueType into Metafont source files, which is available from various archives.

Finally, you should know about the wonderful `pfaedit` utility, which is a general program for working with fonts in a wide variety of formats. With this program, you can manipulate individual characters as well, like commercial programs such as *Fontographer*. `pfaedit` is freely available from various locations on the Web. Please note, though, that this is primarily a Unix/Linux-type program, and though it has been ported to other operating systems, such as Win32, you'll need to download additional software to get it working. (In my case, I had to download Cygwin, a Unix-like terminal for Windows, and XFree86, a free implementation of X-Windows, to get `pfaedit` to work on my Win95 computer.)

אבגדהוזחטיךכלםנסעףפץצקרשת אבגדהוזחטיךכלםנסעףפץצקרשת
אבגדהוזחטיךכלםנסעףפץצקרשת אבגדהוזחטיךכלםנסעףפץצקרשת

GENESIS I

בראשית א

*¹In the beginning of God's creating the
²heavens and the earth — when the earth
was bewilderment and void, with dark-
ness over the surface of the deep, and
the breath of God was hovering upon
³the surface of the waters — God said,
“Let there be light,” and there was light.
God saw that the light was good, and
God separated between the light and the
⁴darkness. God called the light: “Day,” and to the darkness He called:
⁵“Night.” And there was evening and there was morning, one day.*

2 **כ** רֵאשִׁית בְּרָא אֱלֹקִים אֶת
הַשָּׁמַיִם וְאֶת הָאָרֶץ: וְהָאָרֶץ
הָיְתָה תֵהוֹ וְבָהוּ וְחֹשֶׁךְ עַל-פְּנֵי
תְהוֹם וְרוּחַ אֱלֹקִים מְרַחֶפֶת עַל-פְּנֵי
3 הַמַּיִם: וַיֹּאמֶר אֱלֹקִים יְהִי-אֹר
4 וַיְהִי-אֹר: וַיַּרְא אֱלֹקִים אֶת-הָאֹר
כִּי-טוֹב וַיַּבְדֵּל אֱלֹקִים בֵּין הָאֹר
5 וּבֵין הַחֹשֶׁךְ: וַיִּקְרָא אֱלֹקִים לְאֹר
יוֹם וּלְחֹשֶׁךְ קָרָא לַיְלָה וַיְהִי-עֶרֶב
וַיְהִי-בֹקֶר יוֹם אֶחָד:

אבגדהוזחטיךכלםנסעףפץצקרשת אבגדהוזחטיךכלםנסעףפץצקרשת
אבגדהוזחטיךכלםנסעףפץצקרשת אבגדהוזחטיךכלםנסעףפץצקרשת

FIGURE 11: A portion of Genesis. The Hebrew is MasterFont Hadassah. See genesis.tex for details.

6 CONVENTIONS



We use the term ‘convention’ to refer to system by which authors direct *Makor* to do their typesetting. So far, we’ve only encountered the standard *Makor* convention, by which, for example, you type `l` to get ל, you type `shsh`e[etnahta]` to get ששֵׁׁ, and so on. However, there are other conventions available, and this section introduces them to you. The friendly term ‘conventions’ is related to the less friendly piece of jargon ‘ocp list’, which may be familiar to you if you’ve glanced at the Omega documentation. This is a list which tells Omega which rules to apply to the input. *Makor* creates these lists dynamically based on your wishes. We’ll describe the more common conventions before concluding this section with a brief discussion on how to create your own special *Makor* environment.

We specify a new *Makor* environment by means of the command

```
\MakorEnvironment{...}
```

or its synonym

```
\MakorConvention{...}
```

where you replace the ellipsis by a valid environment name. By default, the standard *Makor* environment, named MKR, is in effect.

Variations on the מקור convention

Several variations to the standard MKR environment (set of conventions) are already part of the *Makor* system. The first concerns an alternative to the vowel placement in the final ך character. Normally, you see ך ך but if I type this as

```
\[\MakorEnvironment{MKRalt}kh' khau\]
```

I get ך ך instead. To get ךׇ ךׇׇ ךׇׇׇ I type something like

```
\[\MakorEnvironment{MKRalt}l'khau laekh'  
  \MakorEnvironment{MKR}l'khau laekh'\]
```

Similar comments apply to a ‘bent lamed’ convention which must have originally been adopted to save space. There is a ‘bent lamed’ MKR variant. I get ךׇׇ ךׇׇׇ ךׇׇׇׇ by typing

```
\[\MakorEnvironment{MKRbl}l'khau laekh'  
  \MakorEnvironment{MKR}l'khau laekh'\]
```

(here, bl=bent lamed).

As if to emphasize the lesser value of vowels over consonants, most Hebrew typesetting omits the vowels. Exceptions seem to be made for children’s books and texts, certain

foreign words and names (which no one would know for sure how to pronounce), poetry, and liturgical texts. It's easy to omit vowels in a *Makor* document—simply omit the vowels! Of course, this leads to problems in proofreading. It's much easier to read

```
lakol z'maun w' 'aes l'khaul-chaefets tachas
      hashsh^+m+yim:
```

instead of `lkh1 zmn w's lkh1-chfts schs hshym`: when you want to typeset לכול זמן ועת לכל-הפץ תחת השמים: Consequently, there is a MKR variant to do that. This is the environment `MKRnv` (`nv=no` vowels). I got the preceding vowelless phrase by typing

```
\[\MakorEnvironment{MKRnv}lakol ... hashsh^+m+yim:\]
```

Perhaps systems like *Makor* which make it as easy to typeset vowels (and other diacritics) as consonants may cause the wider use of vocalized reading material.

Non-Makor conventions: ArabTeX, Biblia Hebraica Stuttgartensia

A few important Hebrew keyboard conventions exist prior to the development of *Makor*, and experiments with two of these show that it is feasible to adapt these conventions to *Makor*.

Some years ago, Klaus Lagally developed an ingenious system for typesetting Arabic with \TeX , and shortly afterwards he extended this system to handle Hebrew. (Both scripts are alike in being written RTL and in treating vowels as optional diacritics.) His system is quite robust, but authors using this system are limited to special fonts that are part of the system. By this time, there is a substantial corpus of input files adhering to this system, so I thought it would be a good idea to adapt the Arab \TeX -Hebrew conventions for use with *Makor*.

After typing `\MakorConvention{Lagally}`, input such as `.b.irU.a_h_t0vAh` typesets as בְּרוּיָח טוֹבָה (this is `b'roo_ach .t0vauh` in the standard *Makor* convention). The Arab \TeX package, with its manual, is widely available. Most readers can browse their local CTAN site (www.ctan.org) for this material.

The case of BHS (Biblia Hebraica Stuttgartensia) is more interesting. This text is a careful rendering of the Hebrew Bible, containing standard and cantorial diacritics, oddly shaped characters, oddly vocalized characters, oddly sized characters, and other miscellanea as per the most ancient extant manuscripts. The BHS encoding is more a work-in-progress than a finished project. For some reason, although large chunks of this file compile perfectly in *Makor*, I cannot run the entire BHS file as a whole through *Makor* without generating incomprehensible errors. Reader input/advice/suggestions are very welcome.

Anyway, an ASCII file with this BHS information is publicly available, but it adheres to its own system of transcription. For example, the first bit of Esther looks like

```
WA/Y:HI73Y B.I/YM"74Y ):AXA$:W"R092W$ H74W.)
```

ש ן	ן n	ײ ee	ױ u, oo	א ׳
ש sh	ס s	ײ ey	ױױ vu	א a
ת .t	ע e	ײ ay, ei	ױ v	א o
ת .s	פ p	ײ yi	ױ oi, oy	ב b
דז dz	פ f	כ .k	ז z	ב .v
זש zh	צ ts	כ kh	ח .kh	ג g
דזש j	ק k	ל l	ט t	ד d
טש ch	ר r	מ m	ײ y, i	ה h
		אויף oif		
		פויגל foigl		
		ױף _oif		

FIGURE 12: Yiddish input conventions. The last three lines summarize the conventions by which a leading א is applied automatically to certain vowels, or by which it is suppressed.

```
) :AXA$:W"R081W$ HA/M.OL"K:03 M"/H074D.W. W:/(AD-?K.80W.$
$E91BA( W:/(E&:RI71YM W./M")F73H M:DIYNF75H00
```

(quite different from the standard *Makor* conventions), but if I surround this incantation with

```
\[\MakorConvention{BHS}\BIBLE ...\CENTERLASTLINE\]
```

and run it through *Makor*, I get the proper

ײהי בימי אַחַשְׁרֹזֶשׁ הוּא אַחַשְׁרֹזֶשׁ הַמֶּלֶךְ מֵהַדֹּר וְעַד-כּוֹשׁ שִׁבְעַ וְעֶשְׂרִים וּמֵאָה מְדִינָה:
(The command \BIBLE sets a few other parameters necessary for rendering this text.)

The file I used is available from the Oxford Text Archive Website:

ota.ahds.ac.uk/texts/0525.html

and is about 3.3 meg in size.

Yiddish and Ladino

It is possible, of course, to typeset Yiddish with *Makor* by simply pretending that it's some kind of funky Hebrew, and enter it using the usual *Makor* conventions (together with some definitions to access those Yiddish characters that typically do not appear in Hebrew). But this is silly. It's easy enough to provide a Yiddish environment whose conventions are summarized in figure 12.

It's worth pointing out that some sounds are duplicated in Yiddish. For example, both *v* and *.v* produce different letters which have the same sound. Several of these duplicates have been taken directly from Hebrew, probably so certain printed Yiddish words

look like their Hebrew antecedents. These letters are the ones in figure 12 whose input is preceded by a period.

Certain sounds need a leading **Ⓐ** when they begin a word; *Makor* takes care of this automatically, as you see in the bottom of figure 12. In these last three lines, the input text `oi` is given a leading `.` depending on its position in the word. A leading underscore, as in `_oi` suppresses the leading `.`

Figure 13 provides an example of Yiddish typesetting, displaying the Yiddish translation of “Jabberwocky” by Raphael Finkel; check file `jabberwock.tex` for details of the typesetting, but the conventions for the first stanza are

```
s\'iz brilik geven, di shlikhtinke toves  
hobn gevirt un gevimlt in vobn:  
gants mimzish geven di borogoves;  
di mome-ret hot oysgegrob.
```

which seems to mirror the actual pronunciation pretty well (although I am certainly no Yiddish *maven*).

I believe that the *Makor* fonts are all “Ladino-ready.” That is, they contain all glyphs you might need to typeset Ladino text. Alas, I have not been able to locate an unambiguous table of sounds for a Ladino alphabet. If any reader can provide this, I will be happy to create a Ladino convention for *Makor*. This applies to any other script or dialect using Hebrew characters.

Makor “dialects”

One great advantage of Omega is that it’s so easy to customize your input scheme. We’ve seen examples of that above, and here’s one final one.

I tried hard to make the default input mechanism for *Makor* be as intuitive as possible, but, of course, intuition is a relative thing. It’s entirely reasonable for me, as a speaker of American English, to use the inputs `ts` and `z` for the letters `tsaddik` and `zayin`. But it’s not so intuitive, say, for a German user of *Makor*. Germans represent this pair of sounds by `z` and `s` respectively, and we might design a German ‘dialect’ of *Makor* to reflect this different linguistic perspective. Most input conventions are the same for a German speaker as for an English speaker, but there are some differences, so it makes sense to contemplate a *Makor* convention in which Germans, say, can enter Hebrew text in a way that makes sense for them. To this end, I’ve created the `MKRdeu` environment (`deu=deutsch`). Most of the input conventions remain the same as for the standard set of conventions, and the few differences appear in figure 14, and examples of its use appear at the bottom of this figure.

I make no claim to any expertise in German, and German readers may feel that my input choices are poor. Therefore, I welcome suggestions for improvement, as well as suggestions for other *Makor* dialects.

Jabberwocky דער יאַמערװאָך

Lewis Carroll (1871) לוי קאַראָל

Translated into Yiddish by Raphael Finkel איבערגעזעצט: רפאל פֿינקל

"Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsey were the borogoves,
And the mome raths outrabe.

— "Beware the Jabberwock, my son!
The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
The frumious Bandersnatch!"

He took his vorpal sword in hand:
Long time the manxome foe he sought—
So he rested he by the Tumtum tree
And stood awhile in thought.

And, as in uffish thought he stood,
The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
And burbled as it came!

One, two! One, two! And through and through
The vorpal blade went snicker-snack!
He left it dead, and with its head
He went galumphing back.

— "And hast thou slain the Jabberwock?
Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!"
He chortled in his joy.

"Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsey were the borogoves,
And the mome raths outrabe.

FIGURE 13: The Yiddish Jabberwock! See jabberwock.tex for details.

ש	ˆss	ס	ss	ב	w
ש	sch	ף	f, v	ו	.w
ת	th (not s)	ץ	z	ז	s
		ש	schˆ		

אֶרֶץ יִשְׂרָאֵל 'erez yiˆss'rau'ael
 כְּלֵב־זָאב kelew-s''aew
 מַעֲשֵׂה אֲבוֹת סִימֹן לְבָנָיִם ma':aˆssaeh 'auw0th ssiymaun l'wauniym

FIGURE 14: A German dialect of *Makor*.

Archaic systems: Old Hebrew; Palestinian Babylonian vocalization

As we mentioned above, you get Old Hebrew (sometimes called Paleo-Hebrew) as a font change. Thus, after something like

```

\fontdef{OH}{oldheb}{14pt}
...
\[\font{OH}'vgdh wzch.tyk lmn.s' ftsqrshth\]

```

you get XW99427 O345L Z03SY 3<194, the Hebrew alphabet.

The archaic Babylonian and Palestinian vowel systems were competitors to the Tiberian system that eventually won the day about a thousand years ago. In case you need to typeset using these alternative vowel systems, *Makor* provides an as-yet experimental way for you to do that. Figure 15 shows an example of these systems.

וְשֹׁפֵט בֵּין עַמִּים רַבִּים וְהוֹכִיחַ לְגוֹיִם Tiberian
 וְשֹׁפֵט בֵּין עַמִּים רַבִּים וְהוֹכִיחַ לְגוֹיִם Palestinian
 וְשֹׁפֵט בֵּין עַמִּים רַבִּים וְהוֹכִיחַ לְגוֹיִם Babylonian

FIGURE 15: Archaic Hebrew vowel systems. See `archaic.tex` for typesetting details.

Makor performs this typesetting within the standard MKR environment, but I've created special Ezra fonts containing the archaic vowels in place of the usual Tiberian diacritics. For example, I could define fonts such as

```

\fontdef{Pal}{ezrapal2}{14pt}
\fontdef{Bab}{ezrabab2}{14pt}

```

which I invoke after a call to a special `\ArchaicVowels` command. The examples in figure 15 was keyed according to the following schematic.


```
\def\PAL{\ArchaicVowels \hfont{Pal}}
\[\PAL ...]
```

MKR	Standard <i>Makor</i> conventions; the default.
MKRb1	Like MKR, except uses bent lameds.
Lagally	Conventions of Klaus Lagally's ArabT _E X.
BHS	Conventions of Biblia Hebraica Stuttgartensia.
BHSb1	Like BHS, but with bent lameds.
MKRnv	Like MKR, but no vowels or diacritics are typeset.
BHSnv	Like BHS, but no vowels are typeset.
MKRalt	Like MKR, but non-standard placement of vowels in some terminal letters is used.
MKRdeu	a German (deutsche) variation of the standard MKR environment.

FIGURE 16: Standard *Makor* environments.

Standard conventions; rolling your own...

Makor sets up several conventions (environments) for typesetting in a variety of contexts. Figure 16 lists these, most of which have already been mentioned. Any of the items in the left column of this figure can appear as argument to the `\MakorEnvironment` command. For example, to typeset Yiddish, your input file would contain the following symbolism:

```
\MakorEnvironment{YID}\[\hfont{...} ...]
```

(as usual, the ellipsis ... represents your text).

You may feel you need to create a variant of one of these. *Makor* will cooperate, provided you adhere to the following procedure.

First of all, *Makor* creates a temporary file to contain the instructions needed to customize your set of conventions, and you need to provide a name for this file. If you provide the name `mystuff` (for example), *Makor* creates the temporary file `mystuff.tmp`. You use this name in two places. This name is the argument of a command that *Makor* uses to begin the process. Next, you set the various parameters on or off that you want to adjust; see the file `makor2.tex` for details, but don't change anything in it! After this, issue a command which tells *Makor* to begin the one or more new conventions, not forgetting to provide a name for this new convention. Finally, *Makor* needs to slurp up this information which has been written to the temporary file.

This description has been a bit too vague to be of use, so we present an example. Assume we need an environment that suppresses vowels but does print the cantorial trope at the same time that the Biblia Hebraica Stuttgartensia input rules are observed. The following statements should appear in your source file:

```
\StartSomeOCPLists{mystuff}
\Vowelsoff \BHSon
\MakeMakorEnvironment{TRPNV}
```

\InhaleEnvironmentKnowledge{mystuff}

and you use this new environment in this way.

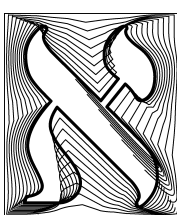
\noindent\MakorEnvironment{TRPNV}\BIBLE[...]

Figure 17 shows the same selection as figure 2, but with all vowels stripped away. You rarely (if ever?) see Hebrew texts printed like this, but perhaps this may prove useful for people studying the chanting of the Hebrew sacred texts.

א אחר הדברים האלה גדל המלך אחשוורוש אתהמן בן-המדתא האיני וינשאהו וישם את-כסאו מעל כלהשרים אשר אתו:
 ב וכל-עבדי המלך אשר-בשער המלך כרעים ומשתחווים להמן כירבן צוה-לו המלך ומרדכי לא יכרע ולא ישתחוה: ג ויאמרו
 עבדי המלך אשר-בשער המלך למרדכי מדוע אתה עובר את מצנת המלך: ד ויהי [באמרם] כאמרם אליו יום ויום ולא שמע
 אליהם ויגידו להמן לראות היעמדו דברי מרדכי כיהניד להם אשר-הוא יהודי: ה וירא המן כי-אין מרדכי כרע ומשתחוה
 לו וימלא המן חמה: ו ויבו בעיניו לשלח יד במרדכי לבדו כיהנידו לו את-עם מרדכי ויבקש המן להשמיר את-כל-היהודים
 אשר בכל-מלכות אחשוורוש עם מרדכי: ז בחדש הראשון הוא-חדש ניסן בשנת שתים עשרה למלך אחשוורוש הפיל פור הוא
 הגורל לפני המן מיום ליום ומחדש לחדש שנים-עשר הוא-חדש אדר: ח ויאמר המן למלך אחשוורוש
 ישנו עם-אחד מפזר ומפרד בין העמים בכל מדינות מלכותך ודתייהם שנות מכל-עם ואת-דתי המלך אינם עשים ולמלך
 אין-שנה להניחם: ט אם-עלה-מלך טוב יכתב לאבדם ועשרת אלפים כרכסף אשקול עלי-די עשי המלאכה להביא אל-נגני
 המלך: י ויסר המלך את-טבעתו מעל ידו ויתנה להמן בן-המדתא האיני צרר היהודים: יא ויאמר המלך להמן הכסף נתון
 לך והעם לעשות בו כטוב בעיניך: יב ויקראו ספרי המלך בחדש הראשון בשלושה עשר יום בו ויכתב כל-אשר-צוה המן אל
 אחשרפני-המלך ואל-הפחות אשר על-מדינה ומדינה ואל-שרי עם ועם מדינה ומדינה כתבה ועם ועם כל-שונן בשם המלך
 אחשוורוש נכתב ונחתם בטבעת המלך:

FIGURE 17: Selection from Esther, vowelless but with trope. See trpvn.tex to see how this was done.

7 HEBREW ALIGNMENTS AND TABLES



I'd like to describe a macro—a variation of plain TeX's \halign—that makes it possible to type Hebrew columns left to right and have them typeset right to left. I used \halign as a model rather than L^AT_EX's \tabular because I've always found that \halign is more versatile. **Caveat!** This macro works by writing certain information to a temporary file; it is this operation that allows Makor to reverse the order of columns. However, as a result, strange things are apt to happen if you go too wild in your table macros. Some of this strangeness will be discussed below. Always be alert to the fact that you may have to reorganize or your tabular data or set up the table in a different way to get it to typeset. I invite motivated readers to re-do these macros more robustly.

Simple tables, such as the one in figure 6, require no special treatment. The table in figure 19 is a slightly different story. Makor provides a

\HEBALIGN

alignment macro for producing tables like this. Although `\HEBALIGN` is no way near as robust or versatile as `\halign` (the plain TeX antecedent of L^AT_EX's `\tabular` environment), I hope it will be of use. Basically, we seek to enter columns and column data from left to right and have it typeset in right to left format.

As with all TeX environments, `\HEBALIGN` requires a template followed by any number of row data. Both the rows and the templates are arguments of a `\CR` macro, which *precedes* the template or row data. As in standard TeX, the tabbing symbol `&` separates each column. *Unlike* standard TeX, there must be the same number of `&`'s on each line. (Otherwise, the data will line up in a manner you won't have anticipated.) In particular, TeX's neat `&&` convention for the template row can *not* be used in `\HEBALIGN`. Essentially, any formatting commands can appear in the template specification. However, you cannot use the `#` tabbing symbol; there is a category code conflict. Use the Makor macro `\H` instead. Thus, a simple table could be typed as

```
\tabskip=1.5pc
\HEBALIGN{\CR{\hfil\[\H\]&\[\H\]\hfil}% end of template
          \CR{'&baeyth}% end of first row
          \CR{gimel&d}% end of second row
} % end of \HEBALIGN
```

and which typesets as the top sample in figure 18. This could also have been coded as

```
\HEBALIGN{\CR{\hfil\H&\H\hfil}% end of template
          \CR{\['\]&\[baeyth\]}% end of first row
          \CR{\[gimel\]&\[d\]}% end of second row
} % end of \HEBALIGN
```

There's also a `\HEBNOALIGN` macro, which works like the usual `\noalign` command. To get the second sample in figure 18, type

```
\HEBALIGN{\CR{\hfil\[\H\]&\[\H\]\hfil}
          \HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
          \CR{'&baeyth}
          \HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
          \CR{\omit Gimel&d}
          \HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
}
```

Protection

Owing to the aforementioned conflicts in category codes, you need to protect a command which is not primitive. For this you use the Makor command

```
\MPROTECT
```

inside the `\definition` for `\myprotects`, which Makor examines. Thus, if we type

```
\def\myprotects{\MPROTECT\multispan\MPROTECT\bf}
\HEBALIGN{\CR{\H&\H&\H&\hfil\[\H\]&\[\H\]\hfil}% end of template
```

בית ד	א גמל
בית ד	א Gimel

שְׁלוֹם!	:	Silly	:	
בית ד	א Gimel	3 3	2 2	1 1

שְׁלוֹם!	:	Silly	:	
בית ד	א Gimel	1 3	2 2	3 1

FIGURE 18: Sample Hebrew alignments.

```

\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
\CR{\multispan3:\hfil Silly\hfil:&\multispan2\hfil\[sh^aul0m!\]\hfil}
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
\CR{\[1\]&\[2\]&\[3\]&'&baeyth}% end of first row
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
\CR{1&2&3&\omit\bf Gimel&d}% end of second row
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
} % end of \HEBALIGN

```

you get the third specimen in figure 18; this table requires some comments: (1) the value of `\tabskip` is 6.0pt; (2) numerals taken from the prevailing Hebrew font may well differ from those taken from the prevailing Roman font; and (3) I needed to `\MPROTECT` both `\multispan` and `\bf`, the latter because the \LaTeX /NFSS assigns a rather different expansion than does plain \TeX . When in doubt, you can always `\MPROTECT` a command; it should do no harm in any case.

Suppressing reversal

If necessary, you can suppress the reversal process of several columns. Do this by enclosing the *data* (not the template) in curly brackets, and these brackets will certainly span several columns. For example, if we make the simple change

```

\tabskip=2pc\def\myprotects{\MPROTECT\multispan\MPROTECT\bf}
\HEBALIGN{\CR{\H&\H&\H&\hfil\[\H\]&\[\H\]\hfil}
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}

```

```

\CR{\multispan3:\hfil Silly\hfil:&\multispan2\hfil\[sh^aul0m!\]\hfil}
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
\CR{-\ [1\]&\ [2\]&\ [3\]}&'&baeyth}%% <== OBSERVE EXTRA BRACKETS!
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
\CR{1&2&3&\omit\bf Gimel&d}
\HEBNOALIGN{\vskip2pt\hrule\vskip2pt}
} % end of \HEBALIGN

```

to the code for the previous table, we get the final table in figure 18.

התפלל							
גזרת שלמים				בניין התפעל			
נקבה				זכר			
צווי	עתיד	הווה	עבר	צווי	עתיד	הווה	עבר
	אתפלל	מתפללת	התפללתי	<i>I</i>	אתפלל	מתפלל	התפללתי
!	תתפללי	מתפללת	התפללת	<i>you</i>	תתפלל!	מתפלל	התפללת
	תתפלל	מתפללת	התפללת	<i>(s)he</i>	יתפלל	מתפלל	התפלל
	נתפלל	מתפללות	התפללנו	<i>we</i>	נתפלל	מתפללים	התפללנו
!	תתפללנה!	מתפללות	התפללתן	<i>you</i>	תתפללו!	מתפללים	התפללתם
	תתפללנה	מתפללות	התפללנו	<i>they</i>	יתפללו	מתפללים	התפללו

FIGURE 19: Using a Hebrew table to display a verb conjugation.

Figure 19 puts most of this stuff together. You can check out the file `hitpael.tex` to examine the innermost details of its production. When you make this examination, note carefully the shenanigans with `\tabskip`, which look odd but aren't. Remember, the template line is reversed in the typesetting process.

8 VARIOUS AND SUNDRY



In this section appear various topics that need to be in a manual, yet don't quite fit into the other sections. These include some bread-and-butter like installation as well as some reference to various tricks.

A trick or two

Sometimes it pays to peruse the guts of the `makor2` style file as well as some of the `.otp` files. For example, if you do this you will note that *Makor* surrounds glyphs with

TeX macros that are responsible for the placement of these characters and symbols on the page. You can—carefully—redefine these macros in aid of the occasional special effect. For example, figure 8 displays the old Hebrew letters that are part of the inscription of the old Siloam water tunnel in Jerusalem. It’s interesting, of course, but most modern readers of Hebrew will want to know what each character represents without having to constantly flip back and forth to a font table. This inscription contains only consonants, each of which is placed in position by a macro called `\PutCon` (`con=consonant`). We carefully redefine this macro so it places the modern Hebrew equivalent underneath the old Hebrew consonant. Here’s the redefined macro:

```
\def\PutCon#1{\oalign{#1\cr
\hidewidth\hfont{HDEFAULTFONT}#1\hidewidth
\cr}}
```

(see *The TeXbook* for further details). The enhanced display appears in figure 20.

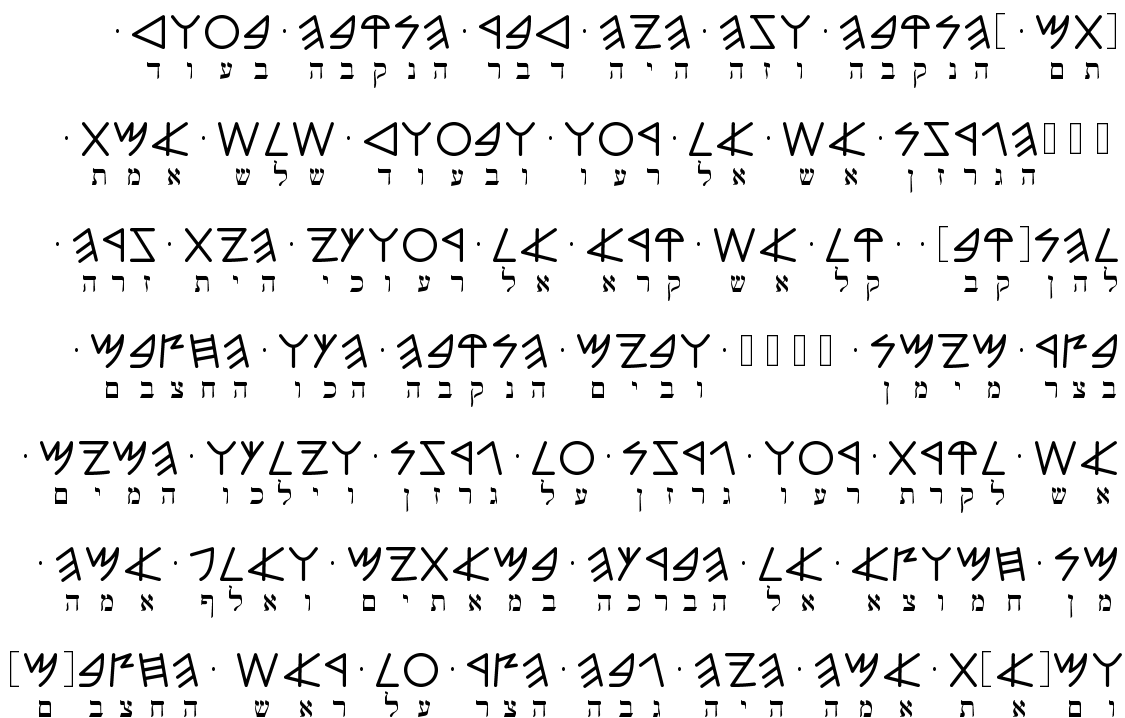


FIGURE 20: An enhanced display of the Siloam water tunnel inscription; see `siloamalt.tex` for details.

Processing this document

I used Omega1.15 to process this document. Although this is not the latest release (as of January, 2003), it appears to be far more robust than version 1.23, at least according to reports on the Omega list.

It proved impossible to compile this entire document in Lambda (that is, the \LaTeX counterpart to Omega) without overflowing Omega's memory resources. As a result, I've included most of the figures by (1) creating separate files for these figures; (2) compiling them separately and creating `.eps` files of the output; and (3) including the `.eps` images in this file by means of the utility `dvips` (actually, the Omega-aware variant, `odvips`). I've included all files with the Makor package that you'll need to typeset this manual (with one exception described below). Thus, to typeset this manual, after having properly installed Makor (see above), you type these commands

```
lambda mkr2man
odvips mkr2man
```

to generate `mkr2man.ps`, which you can view or convert to `.pdf` format. I encourage you to examine the underlying `.tex` file for this manual as an aid to learning how to use *Makor*. To better understand how a figure was typeset, refer to the appropriate source file. For example, to see how figure 1 was created, have a look at `primer.tex`.

As a result of having to include umpteen of these `.eps` files, the resulting PostScript and `.pdf` files are quite swollen. My abject apologies to all.

The only exception to my former statement—that all relevant files are part of the Makor distribution—has to do with fonts. This document is typeset using the Monotype Janson font family, a proprietary set of fonts. If you do **not** have these fonts, then comment out statements like

```
\input zmtmjn
```

or

```
\usepackage{zmtmjn}
```

which appear at the beginning of any of the `.tex` files. With these statements commented out, Lambda and Omega will use the standard \TeX fonts in the compilation.

If you **do** have these fonts installed, then do nothing. I have included the style files `zmtmjn.tex`, `zmtmjn.sty`, and `ot1mjnx.fd` which have been created using my virtual font installation packages `mathinst` and `vfinst`, which you can find on your local CTAN archive. (In these files, I assume that the Janson fonts have been renamed and installed according to the Berry font naming scheme, in which `mjn` designates Monotype Janson. These styles also assume the presence of the MathTime fonts, but unless you attempt to typeset mathematics, this should generate angry warnings but nothing else. If you plan to typeset with these fonts using plain \TeX or plain Omega, you will also need Damian Cudgley's plain font selection scheme, in the file `pdcfse1.tex`, which is part of the `pdcmacs` package on CTAN.) The file `genesis.tex`, which generated figure 11, also uses proprietary fonts, but even if you possess these Hebrew Hadassah fonts, you would need special \TeX macro files to properly typeset this figure.

Caveat!

The program `odvips` suffers from at least one minor bug, which will bite you if you attempt to recreate any of the `.eps` files that are part of this manual. Experienced users know that you can create `eps` from a one-page document with a command of the form

```
odvips -E -o foo.eps foo
```

where `foo.dvi` has previously been created by Omega. For proper inclusion, bounding box dimensions have to be present, and these four numbers are calculated *wrongly* by `odvips`. Before re-creating `foo.eps`, say, make sure you open the old file with a text editor, record these numbers, and then edit them into the new such file. The

`BoundingBox`

comment which contains this information appears within the first half-dozen lines in any `.eps` file.

«לכל זמן וזמן ועת לכל-חפץ תחת השמים: » עת ללדת ועת למות עת לשעת ועת לעקור נטוש: » עת להרוג ועת
 לרפוא עת לפרוץ ועת לבנות: » עת לבכות ועת לשחוק עת ספור ועת רקוד: » עת להשליך אבנים ועת כנוס
 אבנים עת לחבוק ועת
 ועת לאבד עת לשמור ועת
 לתפור עת לחשות ועת
 לשנא עת מלחמה ועת
 עמל: » ראיתי את-הענין
 «את-הכל עשה יפה בעתו
 לאימצא האדם
 ועד-סוף: » ידעתי כי אין
 בתייו: » וגם כל-האדם שיאכל ושתה וראה טוב בכל-עמלו
 יהיה לעולם עליו אין להותיר וממנו אין לגרע והאלקים
 כבוד היה והאלקים יבקש את-נרדף: » ועוד ראיתי תחת
 «אמרתי אני בלבי את-הצדיק ואת הרשע ישפט האלקים כירעת לכל-חפץ ועל כל-המעשה שם: » אמרתי אני בלבי על-דברת
 בני האדם לברם האלקים ולראות שהם-בהמה המה להם: » כי מקרה בני-האדם ומקרה הבהמה ומקרה אחר להם כמות
 זה כן מות זה ורוח אחר לכל ומותר האדם מן-הבהמה אין כי הכל הבל: » הכל הולך אל-מקום אחד הכל היה מן-הקער
 והכל שב אל-הקער: » מי יודע רוח בני האדם העלה היא למעלה ורוח הבהמה הירדת היא למטה לארץ: » וראיתי כי אין
 טוב מאשר ישמח האדם במעשיו כיהוא חלקו כי מי יביאנו לראות במה שיהיה אחריו:

מקור

קהלת ג

FIGURE 21: A portion of Ecclesiastes.