# MOVING TO A UNICODE-BASED LIBRARY SYSTEM: THE YESHIVA UNIVERSITY LIBRARY EXPERIENCE

By: Leah Adler

**Description:** When Yeshiva University Library moved from a non-Unicode automated library system to a Unicode-based system, it found itself faced with many issues, some of which were anticipated, some of which were not. The Hebrew alphabet and diacritics in the Roman alphabet posed special problems.

***Leah Adler*** is Head Librarian of the Mendel Gottesman Library of Yeshiva University and also serves as YU's Systems Librarian. She was the

In January of 2003, the Yeshiva University Libraries migrated their VTLS Integrated Library System from the so-called “Classic VTLS” to the Unicode-based Virtua system. At last year’s convention I reported on the move in general. In today’s talk I want to concentrate on some Unicode related aspects of the move.

On the Unicode home page (www.unicode.org) we find the following quote:

*Unicode provides a unique number for every character,*
*no matter what the platform,*
*no matter what the program,*
*no matter what the language.*

A Unicode-based library system provides unique characters not only for many different languages and scripts, but also for any technical symbol and sign you would possibly need in a written document.

Let’s look at the Microsoft Unicode Character Map.

(Internet Explorer -> Start -> Programs -> Accessories -> System Tools -> Character Map)

Here you see, besides the Latin characters, for example the Question Mark, the Inverted Question Mark (used in Spanish language documents), the Registered Sign and the Pound Sign, and on and on. Each character, as quoted before, has its unique number. We have stand-alone diacritics, like the Tilde, and prearranged combinations of letters with diacritical marks, like the Latin Small Letter N with Tilde.

Now think romanization of Hebrew: Besides regular Latin characters we use Latin characters combined with diacritical marks, like k with dot below, which represents the Hebrew letter Kof.

In pre-Unicode systems we had to use multiple characters in order to represent letters with diacritical marks. For the romanized Kof we used a dot followed by a k, or a squiggly bracket followed by a dot followed by a k, or we left the dot out altogether, just using a plain k, because the combination seemed too unsightly.

In a Unicode-based system, a letter with a diacritical mark is represented by one character, like k with dot below. Therefore, a system which moves from non-Unicode to Unicode based representation has to translate the multi-character combinations into single characters. A squiggly bracket followed by a dot followed by a k will become the single character ḳ (k with dot below) . This is doable.

However, there is a caveat: In many cases Unicode allows for more than one way to represent a combined character. Let's again take the k with dot below as an example and let's look at the Character Map:
(Look at Unicode Subrange – Latin, half way down). Here is the pre-combined k with dot below, number U+1E33. Another way to represent the k with dot below is

to take a regular k and combine it with a combination character “dot below”, which is number U+0323. These two ks with dot below are both Unicode characters, but they are not the same character; they do not have the same Unicode number, and therefore they may not cluster in your database. In order to demonstrate what I mean, I created two short records in our database, using my grandson’s name in romanization. His name is Yehudah Brofsḳy. I entered his name once with the pre-composed k with dot below, and once with the post-composed k with dot below. The system does not cluster the two forms of the name.

So, talk to your system’s vendor before he migrates your database to a Unicode-based system. Ask him which Unicode characters he intends to use, so you will know which ones to use in the future. Tell him to make sure that your entries in the migrated database will cluster with records you will be importing from the utilities, like OCLC and RLIN. Or even better: Talk to him about having the system save different Unicode numbers which represent characters of the same visual typography (as we saw in k with dot below) in the same form. This is called “Normalization”.  And another point: Discuss with your vendor the best way for you to enter Unicode characters after the migration. Ask him to provide you with an easy way to enter these characters. Copying and pasting them from the character map is not an easy way.

Let’s move the discussion to the Hebrew character set: (see Character Map, Unicode Subrange, Hebrew): At the top you see the טעמי המקרא, cantillation marks, or, in Unicode parlance, “Hebrew Accents”. These are followed by the נקודות, vowel points, and by the Hebrew letters. You also find Hebrew letters with degeshim and other marks.

Of special interest to us are the Yiddish digraphs Double Vav, Vav-Yod, and Double Yod, as well as the Geresh and the Gershayim. Of all the many Unicode

characters that VTLS used when they migrated our database to Virtua, these were the ones that posed the more serious problems. You have to remember that Yeshiva was Virtua's Hebrew guinea pig, so some pitfalls were to be expected. In my presentation last year I mentioned these characters and the problems associated with them, but I hope that you don't mind if I repeat some of it today. It will hopefully help some of you when you migrate to Unicode.

Yiddish digraphs:

Yiddish digraphs as you see them in the Character Map comprise one Unicode character each. In Classic VTLS, Yeshiva's pre-Unicode library system, Yiddish digraphs, when imported from RLIN, did not display legibly and were therefore immediately spotted and manually converted into two separate letters each. In Unicode-based Virtua, Yiddish digraphs display beautifully and are indistinguishable from two separate letters. Good news? Not necessarily.

I created a short record entitled ביי מיר. In the word ביי I used the Unicode character for Yiddish Double Yod. Let's search the title in our catalog the way a user would, using the Hebrew keyboard. No ביי מיר to be found. The system does not treat the Unicode character for Yiddish Double Yod as two separate Yods and therefore does not file the Double Yod under Yod but rather at the end of the Hebrew alphabet, which in our case is the end of the Bet sequence.

Geresh and Gershayim:

These are the Hebrew Unicode equivalents of the apostrophe and the quotation mark, which are used in words like ר' or רמב"ם.

Look what happened in our database after we downloaded responsa, or שו"ת titles, from RLIN: We have a split שו"ת-title file! In the first file quotation marks are used in the word שו"ת, and in the second file Gershayim are used. Gershayim are not treated by Virtua as quotation marks, which causes the split file. After this presentation our staff will manually replace the Gershayim in each שו"ת heading with a quotation mark, and we will get rid of the split file.

We at Yeshiva have a special program going: cataloging Ladino books. Ladino, like Yiddish, is written in Hebrew characters. Ladino script does not use digraphs, but it does use very many apostrophes embedded within words. In records imported from RLIN, the character used as an apostrophe is usually the Geresh character. Again, Virtua doesn't treat Geresh as an apostrophe. Let's search for the title ז'ואן. It is not there. But when we search at the end of the ז file, we find it. Once we replace the Geresh in ז'ואן with an apostrophe, the title will file in its appropriate place.

So, before you migrate to a Unicode-based system, think of all the special Unicode characters that you may have in your database and point them out to your vendor, so that he will take them into account when he prepares your system for Unicode. And should you discover problems after migration, as in our case, it's not too late. Reprogramming and reindexing can and should be done.

VTLS is planning to reprogram Virtua in a way that will make the software treat Yiddish digraphs as if they were two separate letters, and the Geresh and Gershayim as if they were apostrophes and quotation marks. This is called "Folding".

Let me read to you a paragraph from the *Committee on Cataloging : Description and Access, Library of Congress Liaison Report to: ALA/ALCTS/CCS/CC:DA, Annual Meeting, June 2002; Submitted by Barbara B. Tillett, LC Liaison to ALA/ALCTS/CCS/CC:DA*

***Normalization and folding rules.*** *A group at LC headed by NDMSO is in the process of revising existing normalization and folding rules for Latin script based cataloging data, as well as drafting for the first time rules for other scripts including, Arabic, Chinese, Cyrillic, Greek, Hebrew, Japanese, and Korean.*

*Normalization involves insuring that various encodings of modified letters are consistent.* [In Yeshiva's case: k with dot below should display, sort and index consistently regardless of its Unicode numerical value.] *Folding involves replacing modified letters or special characters with unmodified or simplified forms for certain activities, such as indexing (for example, an Ae@ ligature might be replaced with the normal letters "a" and "e*".) [In Yeshiva's case: Yiddish digraphs will be replaced with two separate letters.]
The folding process will not physically replace one character with another, but will give the special character sorting and indexing properties of another, simple character.
The paragraph continues: *LC plans to make its conclusions available to others for the Unicode-based versions of their software. Endeavor will use them for the new version of Voyager. LC's normalization and folding rules should be available via the Web later this year.*

Unfortunately I don't know whether LC's plans materialized. At this convention we are lucky to have with us key people from the LC Hebraica cataloging team. I hope they can shed some light on the progress of this proposal.