IST-015685

TEL-ME-MOR

The European Library: Modular Extensions for Mediating Online Resources

Instrument: Specific Support Action

Thematic Priority: Information Society Technologies

No. & Title of Deliverable

D3.1

Report on TEL UNICODE requirements

Due date of deliverable: End July 2005

Actual submission date: 23rd June 2005

Start Date of Project: **1st February 2005** Duration: 24 **Months**

Organisation name of lead contractor for this deliverable:

Name: **National Library of Switzerland**

Version: ***No.1***

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	**CO**

Table of Contents

Purpose

To identify issues brought up by portalling databases with different scripts, character set or individual character processing. This analysis should help the European Library office and The European Library users to make searches more accurate and avoid silence due to misunderstandings of how characters are used.

Method

After a preanalysis of the possible issues, the report author prepared a questionnaire. This questionnaire was e-mailed to the TEL-ME-MOR list with a request to NMS national libraries to fill and return it within about 3 weeks. The replies led to a set of recommendations to the European Library office for the integration of new resources, applicable retroactively to existing ones as well.

Changes since version 0.1

1. Entirely reworked the chapter describing character set issues
2. Integrated the questionnaire and its replies
3. Added recommendations

Changes since version 0.2

1. Corrected some entries in the replies to the questionnaire

Summary

This document first provides general explanations about character sets. It describes in more detail why something may go wrong in the various steps of the information chain, from request to delivery, despite the correct character set conversions already working on the European Library portal. The main issues are:

1. The presence of many different character sets.
2. The presence of special characters that are treated differently from country to country, language to language, or even database to database, changing the input accuracy needed, a fact that most users ignore, as well as user expectations in results presentation.
3. The impossibility or difficulty for users to input these characters.
4. Hurdles arising from the mixture of multiple scripts.

The replies to the questionnaire documented the details of the four issues. It showed that Unicode was already widely used and the first issue above was not too serious. It confirmed the importance of the second and third issues and allowed to evaluate how strong the fourth is likely to be. The solutions envisaged to be as user-friendly as possible while dealing with these issues include

1. A move to Unicode, preferably in its UTF-8 encoding, everywhere it is possible but not yet done.
2. The resort to resources making a bridge between the different practices, such as shared authority files.
3. A web-based input help.
4. Considering an optional automatic transcription between scripts.

Basics about character sets and their issues

Computers store and process information in a succession of *bits*, memory elements able to have only two possible values, 0 and 1. To extend the number of possible values, bits are combined in groups where the number of possible values is equal to 2 to the power of the number of bits, since each added bit doubles the number of possible values. Examples:

2 bits: 00, 01, 10, 11 = 4 values = 2^2

3 bits: 000, 001, 010, 011, 100, 101, 110, 111 = 8 values = 2^3

etc.

One such group of bits can then represent a character. The succession of possible values paired to the characters assigned to them makes a character set.

Each constructor of early computer models defined their own character set, but the need for standardisation was felt in the late 1950's to allow data to be exchanged.

The first standard character set, ASCII (American Standard Code for Information Interchange), had 7 bits, hence 128 available characters and allowed texts in English to be correctly stored. At this stage, languages using another script or the Latin script with diacritics had simply no way yet to process texts in their own languages. ASCII was internationalized in 1967 as ISO 646, with the definition of 10 character positions reserved for "national variants".

Data transmission progressively became reliable enough to use the 8th bit for characters instead of a "parity check", which opened the road to the use of computers in any language with an alphabetical script.

A large number of new 8-bit standards were developed to enrich the available character palette and meet the needs of various languages and user groups. About 180 of them incorporate ASCII and provide compatibility for the first half of the character set but not for the second, as the example below will show.

A person working on a PC configured for Western Europe writes in a simple text file the sentence "Le théâtre sera réparé après le mois d'août, là où même le plâtre paraît neuf." If this file is then transmitted and opened on a MacIntosh or a very old MS-DOS PC also configured for Western Europe, or on a PC configured for Eastern Europe, the sentence will read, respectively:

"Le thÈ‚tre sera rÈparÈ aprËs le mois d'ao°t, l‡ o˜ mÍme le pl‚tre paraÓt neuf."

"Le thΘΓtre sera rΘparΘ aprΦs le mois d'ao√t, lα o· mΩme le plΓtre paraεt neuf."

"Le théâtre sera réparé aprčs le mois d'aoűt, lŕ oů męme le plâtre paraît neuf."

Although the differences are bigger between platforms than between geographic areas, the display cannot be considered satisfactory.

Techniques to increase character set compatibility and language/script interoperability

Software publishers have developed various techniques to overcome this issue, at various levels.

Loading more than one character set

This possibility exists now on all operating systems. It can be implemented in several ways:

1. within the operating system, providing display only or display and keyboard input for (almost) all applications.
2. within a particular application, such as a text editor or a browser, providing a correct display of text files and web pages.

However, the installation of added character sets requires administrator rights, is optional and not always included in default installations. Since the usual user of a PC within an institutional network is unlikely to have administrator rights for her machine, accessing information provided in a character set different to hers may be difficult, hard to read or even impossible.

Our tests with the partners' web pages presenting the project TEL-ME-MOR demonstrated the limits of this approach. Character set support in browsers at SNL includes Eastern Europe (e.g. ISO 8859-2 or win-1250) but not Baltic (e.g. ISO 8859-4 or 10 or win-1257) nor Greek (e.g. ISO 8859-7 or win-1253). With this configuration, the web pages of the Czech and Slovenian partners, encoded in ISO 8859-2 and win-1250 respectively were correctly displayed. On displaying the web page of the Estonian partner, encoded in ISO 8859-4, the browser suggests the installation of the Baltic character set support, which fails because of insufficient rights. Some characters were then displayed incorrectly, for instance the name of the national figure in honour of whom the Lithuanian National Library is named appearing as "Ma¾vydas" instead of "Mažvydas". One would then fear the web page of the Cypriot partner to be illegible for similar reasons. However, because that page was made on the same platform and with the same text editor as the ones installed at SNL and contained references to them, the browser could use the character set support of the text editor, which luckily includes Greek.

The solution of loading several 8-bit character sets looks too much subject to chance to be really applicable.

Unicode and ISO 10646

Unicode is a 16-bit character set aiming to cover all needed characters in a single character set. 16 bits allow for 65,536 characters, quite enough to handle all alphabets and syllabaries. This number has been extended to 1,114,112 by defining a virtual 32-bit section for ideographs. Like 8-bit character sets incorporating ASCII, it incorporates Latin-1 (ISO 8859-1), providing compatibility for Western Europe without conversion.

ISO 10646 encompasses Unicode as its first part and can include more characters but so far has defined no character outside the Unicode range.

In 8-bit character sets, the link between a character's code and the numerical value actually processed by, and stored in, computers is straightforward: it is always the same in (almost) all cases and applications. Unicode on the other hand makes a distinction between a character's code as an abstraction and how it is translated into bits by an *encoding scheme*. The choice of an encoding scheme depends on what is to be done with the characters, e.g. storage or processing, and the constraints specific to these actions. There are 3 main schemes, called UTF-32, UTF-16 and UTF-8.

In UTF-32, each character is encoded on 32 bits, or 4 bytes. It is really relevant only where mostly ideographs are concerned. In UTF-16, each character is encoded on 16 bits, or 2 bytes, except characters whose code is higher than 65,535 (hexadecimal FFFF), where 4 bytes are needed. This encoding scheme is preferred in applications where the predictability of the position of a particular character within a string should not depend on what precedes it. In UTF-8, each character is encoded on 1 to 4 bytes according a specific algorithm. ASCII characters need 1 byte, all other European characters (including non-Latin scripts) and Middle-Eastern scripts need 2 bytes, and 3-4 bytes are needed for Asian scripts and some special characters such as mathematical symbols. This encoding scheme is preferred where space or transmission speed is an issue and is therefore widespread on the web. The Hungarian, Latvian, Lithuanian and Slovak partners used UTF-8 on their web page presenting the project.

Unicode today is supported on all platforms and in many applications, including database systems and printers. Since Unicode incorporated characters from all the other existing character sets (bar some East Asian ones) a lossless conversion is always possible from these character sets to Unicode, the reverse being of course not true.

Are character sets still an issue ?

Yes, there are issues to consider at the three main steps of user interaction with the portal:

1. Inputting search terms and search resources in a character set that is not installed on the user's computer.
2. Dealing in searches with 8-bit character sets and different ways of indexing the same special characters from database to database.
3. Processing returned records according to different users expectations.

Search requests input and distribution

The aim of The European Library is to provide access to several resources through distributed searches. These resources are not necessarily encoded in the same character set, thus leading to the need for conversions. Several cases must be considered, starting with Latin characters:

1. If all characters of the search terms encoded in Unicode have an equivalent in the non-Unicode character sets of the targets, a lossless conversion is possible. This is the case for ASCII characters, but also for some special characters present in many 8-bit variants. For instance, all ISO Latin 8-bit character sets have ‘ä’, ‘ö’ and ‘ü’. These characters can then be sent as such in the requests (leaving it to the targets to ignore the diacritic or not).
2. If some characters of the search terms encoded in Unicode have no equivalent in some of the non-Unicode character sets of the targets, a lossless conversion is impossible. The loss may however be totally or partially unimportant: the missing character could be a letter with a diacritic generally not taken into account in library systems anyway, which would remove it for searching. In this case, the portal could do it upstream, before sending requests in character sets where the original character is absent.
3. A loss induced by the conversion may actually help by extending the search. For instance, if a user searches ‘Łodz’, this term will be sent as is in requests to targets having Unicode or the East European character set. ‘Ł’ being absent from the other 8-bit character sets, libraries using one of those have been forced to reduce this character to ‘L’ and encode ‘Lodz’. A search term reduced that way will still be successful in accessing the relevant records.

In the case of the central index, a conversion will have occurred at the time of the record deposit, so that a conversion is not needed any more at the time of searching.

The issue on input is that keyboards are nationally, regionally or linguistically localized to allow input of special characters of the national language(s), plus a few others almost arbitrarily chosen. Despite the fact that Unicode is now the character set lying at the heart of today’s operating systems, keyboard drivers are still restricted to keys and key combinations needed for the national language(s) for which their keyboards are designed. For instance, a keyboard designed for French, a language where an acute accent may appear on e, can put one on vowels, á é í ó ú ý, but not on any other letter although the Unicode character set contains many (ć Í ń ŕ ś ű ẃ ý ź, and also on ç g k m and p in the Arial Unicode MS font), nor can it put a caron (or haček) on any letter. Input by other means, when possible, is clumsy. Within the Latin script, this would not be a problem if all special characters were a letter with a diacritical mark and if all diacritical marks were “weak”, i.e. ignored in indexing, like the distinction between upper and lower case. This is however not the case: numerous languages have special characters, including diacritical combinations, that hold for a character on its own. ‘Ä’ is not reduced to ‘A’ in most databases in German, neither is ‘Å’ in Swedish, or is ‘Ł’ equivalent to ‘L’ in Polish. Needless to add that Greek and Cyrillic bring challenges of their own, since their entire alphabets are unavailable on a Latin keyboard.

There are several ways to circumvent these problems. One is to tailor requests according to targets, with variations depending on what is searched:

1. For names, this is actually an ancient and well-known problem, best illustrated by a transliteration example but valid even when working in an environment with Latin characters only: many know the classical case of Чайковский being transliterated into innumerable forms, from Čajkovskij (one of the rare schemes being lossless both ways) to Tchaïkovsky. Here, the solution lies in rich authority files, either in the partners’ systems, or as a shared authority file such as LEAF. For instance, the authority record for the well-known composer in the Lithuanian partner’s system has variant forms corresponding to no less than 32 different transliteration schemes (including forms where first name and last name do not use the same one). With such a comprehensive and high quality authority record, any kind of transliteration is likely to lead to a successful search in that database. If all partners’ authority records could be pooled into LEAF and the name search terms first submitted to it to get all variants, requests could be expanded and input would no longer be an issue for those names present. When all forms of a searched name cannot be found in the authority files, distributing a request to Latin and non-Latin databases would theoretically require a transliteration. However, transliteration schemes are rarely lossless, especially to Latin, because they try to render phonetics in the various target languages, instead of the original spelling (they would then be more properly called *transcription* schemes). This is particularly acute in the case of

Greek: what many transcription schemes render as ‘i’ could originally be spelt ‘ι’, ‘ει’, ‘η’, ‘οι’ or ‘υ’. An automatic transliteration is therefore unlikely to help much.

2. Subjects can also be pooled into a common authority file to expand searches to the various human and indexing languages. MACS and MSAC address this issue.
3. In simple word searches, an “automatic” translation could be offered as a stopgap solution.

Note that these options may work well only with differentiated search fields, i.e. in the advanced search. The ordinary keyword search will either bring silence, if none of these options are applied, or noise, if they are applied indiscriminately. Their practicalities will be further studied later in this workpackage.

Because these external resources can never be exhaustive (or if they cannot be set up at all), an added solution is to have a search input web page offering a software extended keyboard within it allowing the input of characters unavailable on some keyboards but necessary for a successful search. A suggested code for such a software extended keyboard is available, working on Internet Explorer, that could be integrated into the European Library search page, translated, and completed with code working on other browsers.

As already mentioned, the same special character may be indexed as a letter on its own in a database and as its base character in another database, according to language or national tradition or system capabilities. When a special character is reduced to its base character for indexing, systems accept the original special character as input and convert it to its base character for searching. However, users do not obligatorily know the special characters outside the language(s) they speak and, even when they do, may find burdensome to have to encode all special characters “just to be sure” that a search will be always successful, since they will not obligatorily search in *all* available resources. They should therefore be made aware of which characters in which resources and in which cases must be correctly encoded to avoid silence.

Results display

Mixing character sets in a single web page is no easy task, since the character set is declared in the `<head>` statement and is not supposed to change along the page’s body. One solution is the one chosen by the Cypriot partner for its web page describing the project. It looks quite efficient, but is partly proprietary and might work less well outside Windows environments.

Another solution is to use frames, as each one has its own `<head>` statement, hence the possibility of having a different character set on each frame.

There is actually no absolute need to mix character sets on the result screen: everything not returned in Unicode should be converted to it.

The European Library portal seems to have chosen something between the two last solutions above: the top of the page and the navigation part on the left hand side make one frame, and the result part on the right hand side makes another frame. In theory, this would allow to have one character set for the language chosen by the user and another character set to display records from the selected resource (the portal displays results from one database at a time). However, since these are dynamic pages, their character set might not be changed depending on content and conversions would still be needed for results. The navigation frame uses the West European 8-bit character set, even when one chooses Slovenian as interface language, and the results frame uses Unicode UTF-16, even when the selected resource has returned records in UTF-8 (occasionally making such results illegible if the character set conversion module is inactive, which was the case on 22 March 2005).

Correctly displayed pages can be correctly printed as well, except maybe on extremely old printers whose driver would be restricted to the local 8-bit character set.

Although names, loan words and common roots can make parts of a text understandable enough to users, the text of the results may remain obscure to them, even when they may lead to non-textual resources where knowing the language would be unnecessary. Considering an optional automatic translation is outside the scope of this report. However, if this option cannot be offered, being unable to read a script will scrap the chance of guessing the content by names, loan words and common roots. Another expectation users may then have is therefore results optional transcription. Contrary to search formulation, where the loss of the correct spelling through transcription makes it impracticable, a transcription of results, tailored to the interface language, would help users.

Finally, an often forgotten user expectation is localized ordering or collation. The alphabetical order may widely vary from language to language, or country-to-country in much the same way the special

characters may or may not be considered as individual letters. This is currently an academic question in the case of The European Library, since results are unsorted or sorted by a non-alphabetical criterion, but should The European Library offer this option at a later stage, this issue should definitely be addressed. The example below shows the level 1 collation (higher levels would concern weak diacritics and case) of English (among others) and Estonian:

English: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Estonian: A B C D E F G H I J K L M N O P Q R S Š Z Ž T U V W Õ Ä Ö Ü X Y

It is clear that users may be seriously confused by a collation they are not familiar with.

Questionnaire and results

A questionnaire of 9 questions, each completed by a short explanation about its aim, was sent to NMS partners to fill. This section provides the original questions and comments, the results and a posteriori comments where appropriate.

Used character sets

1. For each resource you intend to make accessible through the European Library portal, please indicate in which character set(s) requests are accepted.

The aim of this question is to know whether a conversion from UTF-8 is needed

2. For each resource you intend to make accessible through the European Library portal, please indicate in which character set records are stored.

The aim of this question is to know whether there would be a way to bypass a conversion loss, e.g. a database in Unicode accepting requests in a 8-bit character set because it is locally assumed that most PCs still do not yet have Unicode support

3. For each resource you intend to make accessible through the European Library portal, please indicate in which character set(s) records are returned (if different from 2).

The aim of this question is to know whether a conversion to UTF-8 is needed.

Library	resource	Accepted in requests	Used in storage	In returned records
Cyprus		UTF-8	UTF-8	UTF-8
Czech Rep.	NKC	win-1250 and UTF-8	UTF-8	win-1250 or UTF-8
	SKC	win-1250 and UTF-8	UTF-8	win-1250 or UTF-8
	Manuscr.	win-1250	UTF-8	win-1250
	idem (OAI)	N/A	UTF-8	UTF-8
	Kramer.	UTF-8	UTF-8	UTF-8
Estonia	ESTER	UTF-16	Multiple	UTF-8
	DigAr	UTF-8	UTF-8	UTF-8
Hungary	Amicus	UTF-8	UTF-8	Z39.47
	MOKKA	ISO 8859-2	Z39.47	Z39.47
	HEL (OAI-PMH)	N/A	ISO 8859-2 + entities	ISO 8859-2 + entities
	NDA (OAI-PMH)	N/A	UTF-8	UTF-8
	Corvina (OAI-PMH)	N/A	UTF-8	Can be anything
Latvia		UTF-8	UTF-8	UTF-8
Lithuania		UTF-8	UTF-8	UTF-8
Poland		ISO 8859-2	ISO 6937-2	ISO 8859-2
Slovakia		UTF-8	UTF-8	UTF-8

Library	resource	Accepted in requests	Used in storage	In returned records
Slovenia	OPAC others	win-1250 UTF-8	Maybe not a standard UTF-8	win-1250 UTF-8

The Hungarian and Estonian partners pertinently pointed out that the character set of returned records may be different depending on the protocol used. The table above assumes that the protocol is Z39.50 or indifferent, unless specified otherwise.

One can see that Unicode has been now widely adopted and that character set conversions will be limited to two for input (ISO 8859-2 and win-1250) and three for output (Z39.47, ISO 8859-2 and win-1250). Note that there are a few discrepancies between some of these replies and their equivalents in deliverable 2.1 – NMS Requirements Analysis (which is actually a little unclear by making no distinction between input, storage and output):

- Cyprus annonced ASCII in D2.1 but UTF-8 here; the truth is obviously UTF-8.
- Estonia announced Latin-1 and ALA in D2.1 but UTF-8 and UTF-16 here; the truth is obviously Unicode, since a search with Cyrillic characters is successful.
- Poland made the same replies to the two questionnaire. However, "UTF-8 without diacritical marks" (i.e. ASCII) has been ignored here because it is an option of the web interface, not Z39.50.

Authority file(s) for names

4. Do(es) the resource(s) you intend to make accessible through the European Library portal have an authority file for names and, if yes, are authority records generally rather exhaustive ?

The aim of this question is to know whether a centralization through LEAF makes sense: if all (or most) partners have good authority files, these will do the job without the need of a diversion through LEAF.

Library	resource	Authority file(s) for names and their exhaustivity
Cyprus		Currently no authority file but there is a project to make one
Czech Rep.	NKC SKC Manuscr. Kramer.	Yes, ca. 200,000 personal names The same as above No No
Estonia	ESTER DigAr	Yes, authority records rather general, diversion through LEAF welcome No
Hungary	Amicus MOKKA HEL NDA Corvina	Yes, but not complete: 13,000 records of the estimated 80,000 needed No No No No
Latvia		Yes, but authority records are not yet comprehensive (ca. 53,000 personal and institution names)
Lithuania		Yes, there are exhaustive authority files for names and place names
Poland		Yes, and they are exhaustive – name, title and subject authorities
Slovakia		Yes, of about 30.000 personal names, at present rather not exhaustive but gradually enriched
Slovenia		Not for the resources accessible through The European Library but there is one for the manuscripts collection; the application is pretty basic but a

Library	resource	Authority file(s) for names and their exhaustivity
		good basis

The question of authority files will be addressed again later in this workpackage but one can already see that pooling existing authority files is very likely to be of interest.

Languages

5. Indicate the majority language of your country, its minority languages when applicable and the percentage of the population speaking them as mother tongue.

The aim of this question is to find out languages that are spoken in more than one country, in order a) to add another confirmation of the sense of 'portalling' their libraries, b) possibly to help defining default profiles where resources of the same or similar languages would be offered together and c) to detect possible synergies, e.g. on indexing languages.

Library	Languages (and for minority languages: percentage of population speaking it)
Cyprus	Greek, Turkish (about 18%), Maronite Arabic (0.6%), Armenian (0.3%)
Czech Republic	Czech
Estonia	Estonian, Russian (30%)
Hungary	Hungarian, 13 minority languages (3% together)
Latvia	Latvian, Russian (29%), 7 other minority languages (2% together)
Lithuania	Lithuanian, Polish (7%), Russian (6%), others (3.5% together)
Poland	Polish, German (1.3%), Ukrainian (0.6%), Byelorussian (0.5%)
Slovakia	Slovak, Ruthenian (0.3%)
Slovenia	Slovene, Italian, Hungarian (less than 0.5%)

Special characters

6. (for NMS with Latin script) Do(es) the language(s) of your country and/or the resource(s) you intend to make accessible through the European Library portal contain special characters **that are not assimilated to a basic Latin letter** (a-z), i.e. searched and sorted separately (e.g. 'Ł') ? Please detail.

The aim of this question is to define a soft keyboard for Latin special characters.

Library	resource	Special characters
Czech Rep.	NKC	Č Ř Š Ž in searching; collation on level 1, after base letter
	SKC	Č Ř Š Ž in searching; collation on level 1, after base letter
	Manuscr.	Á Č Ď É Ě Í Ň Ó Ř Š Ť Ú ů Ý Ž in searching, but users will be able to choose whether diacritical marks are ignored or not in searching; collation on level 1, after base letter
	Kramer.	Č Ř Š Ž in searching; collation on level 1, after base letter
Estonia	all	Š Ž Õ Ä Ö Ü in searching; special collation on level 1
Hungary		Diacritical marks ignored in searching; collation on level 2
Latvia		Diacritical marks ignored in searching; collation on level 2
Lithuania		None, diacritical marks ignored in searching; collation on level 2

Poland		Ą Ć Ę Ł Ń Ó Ś Ź Ż in searching; collation on level 1, after base letter
Slovakia		Č Ř Š Ž in searching; collation on level 1, after base letter; special collation on level 1 for CH, after H
Slovenia	OPAC	Č Š Ž, diacritical marks ignored in searching; collation on level 1, after base letter
	others	Č Š Ž in searching; collation on level 1, after base letter

This question has often been misunderstood because the distinction between “weak” and “strong” diacritics was not made clear enough in the questionnaire. Trying to assess the situation from outside has proved difficult because public interfaces often include some character set conversion, results sets are not always sorted and a browse function is not always available. However, clarifications were provided and make the above table reliable enough. These results show that:

- Users must be made aware of these characteristics for successful searches
- A soft keyboard is definitely needed
- Collation issues must be addressed when the time comes.

Non-Latin scripts

7. Does your country include a significant part of the population using a non-Latin script and personal computers with a non-Latin keyboard ? which script ?

This is just an introductory question to 8 and 9; beside Cyprus, whose reply is obvious, at least some Baltic states are expected to reply yes to this question as they have strong Russian minorities.

8. (for NMS replying yes to question 7) Do the keyboards of most personal computers used by these people allow an encoding in Latin script and, if yes, is the switch from their script to Latin easy enough, i.e. at one or two mouse click(s) or keystroke(s) ?

The aim of this question is to evaluate the need for a full Latin soft keyboard

9. (for NMS replying yes to question 7) Do(es) the resource(s) you intend to make accessible through the European Library portal contain a significant part written in that non-Latin script ?

The aim of this question is to evaluate the immediate need for Cyrillic support, a) by a Cyrillic soft keyboard and b) an added conversion if a resource contains Cyrillic only material in a 8-bits character set (though this should already appear in the reply to question 1).

Library	Non-Latin script	Easy Latin encoding	Frequent in resources
Cyprus	Yes, Greek	Yes	Yes
Czech Republic	No		
Estonia	Yes, Cyrillic	Yes	Yes
Hungary	No		
Latvia	Yes, Cyrillic	Yes	Yes
Lithuania	Yes, Cyrillic, but mostly from Latin keyboards	Yes	No
Poland	No		
Slovakia	Yes, Cyrillic	Yes	No
Slovenia	No		

These questions may seem to have been focused towards Cyrillic, not at all because Greek was forgotten, but because its situation was clearer from the outset. These results show that a full Latin soft keyboard is unnecessary, since users working primarily with the Cyrillic script can easily encode Latin characters. The reverse is not true, at least outside the countries where Cyrillic users are

numerous, making a Cyrillic soft keyboard useful. The presence of material in Cyrillic script, in addition to Greek, also reinforces the need for optional transliterations, unless automatic translation is offered.

Recommendations and proposals

TEL-ME-MOR and library info web pages

Since recent browsers support Unicode and are generally available free of charge, we recommend to put all info web pages in UTF-8. This is very simple to do: open the file, change the character set declaration in the `<head>` statement, and save the file after having chosen UTF-8 as character set (in MS-Word, under 'coded text').

Library systems

Since conversions are possible and offer a good enough level of compatibility and accessibility throughout the resources offered by the European Library portal, upgrading local 8-bit databases to Unicode support is not an immediate necessity. However, it would be a good idea to think about it when an upgrade is considered for any other reason. Since UTF-8 encodes Unicode in bytes, database systems working on 8 bits may become Unicode-compatible. What is then needed is a conversion module between the local 8-bit character set and UTF-8, inserted between the database and its interface(s).

The European Library portal

We recommend the European Library office:

1. To move the entire portal, or at least the navigation frame, to UTF-8, the best option for the web. On the search terms input and navigation frame, this will allow the input of all kinds of characters used by partners' resources and correctly display labels and navigational texts. A software extended keyboard should be added to allow the input of characters significant for searching but absent from some hardware keyboard. On the results frame, moving to UTF-8 might improve performance, since UTF-8 is more frequently used in returned records and more compact than UTF-16.
2. To address the questionnaire retroactively to current The European Library members, since these character set issues have been overlooked until now. For instance, a search on 'Böll', 'Boll' and 'Boell' on current default collections shows that the British Library integrated catalogue ignores the Umlaut ('Böll' and 'Boll' give the same result set, 'Boell' does not), whereas SNL's Helveticat takes it into account ('Böll' and 'Boell' give the same result set, 'Boll' does not).
3. To complement the collections descriptions by information about the use of special characters, where appropriate.
4. To add to the search page 3 displayable soft keyboards, for Latin special characters, Greek and Cyrillic, with information about script prevalence, their characters' use and usual transliteration (prototype available).
5. To study the feasibility of adding to the results page an automatic transcription option from Greek and Cyrillic to Latin, tailored to the interface language, respectively from Latin and Cyrillic to Greek on the interface in Greek.