Internationalisation proposal

\\stephen\C:\MPE\PROJECTS\Forth202x\Internationalisation\i18n.propose.v9.doc
Revised : 14 July 2018

Authors:
Stephen Pelc, MicroProcessor Engineering, sfp@mpeforth.com
Willem Botha, Construction Computer Software, willem.botha@ccssa.com
Peter Knaggs, pjk@bcs.org.uk

Contributions from:
Greg Bailey, Athena Programming, greg@minerva.com
Nick Nelson, Micross Electronics, njn@micross.co.uk

Contact:
Stephen Pelc
MicroProcessor Engineering
133 Hill Lane
Southampton SO15 5AF
England

Tel: +44 (0)23 80631441, +44 (0)78 0390 3612
Net: sfp@mpeforth.com

Rationale

Forth Applications designed to run in many countries and languages cannot yet make enough assumptions about strings and character sets to be portable. The LOCALE word set is designed to provide words for portable internationalisation of application programs. The proposal does not attempt to cover text processing in general, but only to permit conversion of a limited set of application defined text for internationalisation.

In practice, many applications are not localised by the software developer, but by their agents in other countries. The LOCALE word set permits the software developer to provide tools that will produce text files that can be edited and converted to another language locally without dependency on computer language or operating system specific tools such as resource compilers and managers. At the same time, the proposed word set does not inhibit the use of sets of statically compiled strings for each language, it just does not define the mechanism.

The basis of the LOCALE word set is that all strings for internationalisation are compiled as LOCALE structures, and all access to the strings is through these structures. It appears that the following word set is adequate in the first place. The word set is designed to cope with character sets that are of different size to the native set.

The word set is split into a base and extension sets to indicate what factors need to be language sensitive. It is also likely that all LOCALE structures will need to be linked in case reindexing of hash tables or other internal structures is necessary.

The word **L”** is proposed for language sensitive strings, and behaves in a similar way to the ANS word **C”**, but returns a string identifier known as a locale string identifier (lsid) from which the required language string can be extracted. The reason for this is so that text

information in the native development language is still available in the source, making source maintenance easier because the intention of the string is still available to the developer. In addition, the Forth compiler can be extended to produce a text file containing the native strings.

The number of items to be displayed which are, or may be, language sensitive is large. Not all applications may need to deal with all of them. In addition, many applications need to be able to perform text substitution, for example:

```
Your balance at <time> on <date> is <currency-value>.
```

We can provide for both these requirements by using the text macro expansion facility already standardised in Forth 2012. For example we can provide an initial string in the form:

```
Your balance at %time% on %date% is %currencyvalue%.
```

This proposal assumes that character handling will be performed using the Extended-Character word set defined in Forth 2012.

Terminology and assumptions

LOCALE

We use the word **locale** to mean the mixture of country, language, font, date/time formatting and so on in use when an application program runs.

Character sets

The language and character set encoding used by a Forth system at development time is referred to as the Development Character Set (**DCS**). The development character set of a particular Forth is assumed never to change. It is furthermore assumed that character manipulation in the Forth system is defined in terms of the DCS, and that the action of character operations such as **`CMOVE`** is locked to the DCS.

The language and character set encoding used by any underlying operating system is referred to as the Operating Character Set (**OCS**). The OCS may or may not be the same as the DCS.

The language and character set encoding used at application run time is referred to as the Application Character Set (**ACS**). It is assumed that the largest character in an ACS fits in the native cell of the development Forth system. The only LOCALE word set use of individual characters is for setting macro escape characters (see later). The ACS may or may not be the same as the OCS.

The DCS is usually seven or eight bit ASCII in the majority of today's Forth systems, but we will see Unicode systems in the near future. The OCS is defined by the host machine, and is defined by the user of the application. Thus, an application written in a Forth designed for ISO-Latin1 may be running on an O/S with a Chinese OCS, and a visitor may switch the application into yet another ACS, such as Russian. Such scenarios are rare within the US and Europe, but are common elsewhere in the world. Countries such as South Africa exist with 11 official languages, and some languages such as Portugese and English are spoken in many different countries.

LOCALE structures

We do not wish to constrain or influence implementation techniques in any way. A specific string for internationalisation needs to be referred to by a single parameter, which we call the "locale string identifier", or *lsid*. This is an opaque type, in other words the programmer

should make no assumptions about what it means, except that different strings have different *lsids*. In many cases, an *lsid* may well be an address.

LOCALE strings

At application run time, locale strings need to be manipulated. Locale strings are described in terms of address units. For brevity, locale strings are also referred to as *lstrings*.

Country and language constants

There are a number of standardisation efforts for country and language codes. Since the objective of this document is to provide for source portability of applications, we do not need to mandate numeric or string values, but only to define language and country source names that can be used as Forth words.

Assuming that text processing is mostly affected by language selection, and that formatting is heavily influenced by both country and corporate standards, we suggest that country be defined by the ISO3166:1998 two letter country codes (Alpha-2). For this standard an algorithm has been defined to produce unique numeric codes for each country. A set of language codes (ISO639:1998) also exists.

Octets and Bytes

Since the vast majority of character sets are defined in terms of 8 bit units commonly referred to as bytes or octets, it is likely that the implementation of any internationalisation code will require the presence of byte/octet access words, regardless of the underlying DCS character size. For Forth systems running on byte-addressed machines (the vast majority) a byte corresponds to a Forth 2012 *pchar* in the majority of Forth systems.

The presence and definition of an octet/byte access mechanism is outside the scope of this proposal.

The optional LOCALE word set

Environmental queries

Append the table below to table xxx

String value	Data type	Constant?	Meaning
`LOCALE`	Flag	No	LOCALE word set present
`LOCALE-EXT`	flag	No	LOCALE extension word set present

Additional documentation requirements

Error conditions

- use of an invalid locale string identifier (*lsid*),
- a locale string is too big for a destination buffer.

LOCALE words

SET-LANGUAGE \ lang -- ior ; lang is a language code
Sets the current language for the LOCALE system. The *ior* is returned zero if the operation succeeds, otherwise it returns a non-zero implementation-dependent *ior*. If the operation does not succeed, the current language remains unchanged.

GET-LANGUAGE \ -- lang
Returns the language code last set by **SET-LANGUAGE**. The default language is implementation defined.

SET-COUNTRY \ country -- ior ; country is a country code
Sets the current country for the LOCALE system. The *ior* is returned false if the operation succeeds, otherwise it returns a non-zero implementation-dependent *ior*. If the operation does not succeed, the current country remains unchanged.

GET-COUNTRY \ -- country
Returns the country code last set by **COUNTRY**. The default language is implementation defined.

L” \ -- ; C: lsid ; **L” <native text>”**
Interpetation:
The interpretation semantics for this word are undefined.

Compilation: \ "ccc<quote>" --
Parse *ccc* delimited by a " (double-quote) and append the run-time semantics given below to the current definition.

Runtime: \ -- lsid
Return *lsid*, an identifier for a locale string. Other words use *lsid* to extract language specific information.

LOCALE@ \ lsid -- addr len(au)
Return the address and length in address units of the string (in the current language) that corresponds to the native string identified by *lsid*. The format of the string at *addr* is implementation dependent. The length of the string is returned in address units so that it may be copied by **MOVE** without knowledge of the character set width.

LOCALE extension words

These words are provided here to give portability of implementation techniques. They are building blocks for a practical implementation.

LOCALE-INDEX \ lsid --
Updates the internal data structure. Useful if structures are added and changes to internal structures are required.

LOCALE-LINK \ lsid1 -- lsid2
Given the address of one LOCALE structure, returns the address of the next.

LOCALE-TYPE \ addr len --
Displays the LOCALE string whose address and length in address units are given.

NATIVE@ \ lsid -- c-addr len
Given a LOCALE structure, returns the address and length of the corresponding DCS native string that was compiled by **L"**.

Reference implementation

See the file `i18n.v8.fth`.

Change history

14 July 2018
- Changed ambiguous consitions to errors.
- Updated contact details

25 April 2001
- Added GET-ESCAPE to provide restoration capabilities.

25 March 2001
- Minor text changes
- Produced reference implementation for MPE VFX Forth v3.40

21 June 1999
- Wordsmithed at ANS meeting

20 June 1999
- Tightened up some wording
- Added references to more standards.

14 June 1999
- Added an ambiguous condition to **SUBSTITUTE**.
- Changed **COUNTRY** and **LANGUAGE** to **SET-COUNTRY** and **SET-LANGUAGE** returning an *ior*.

30 May 1999
- Derived from parallel discussion document