A truly International Standard

Peter Knaggs

University of Paisley, High Street,
Paisley. PA1 2BE Scotland.

pjk@bcs.org.uk

May 19, 1998

Abstract

When ANS Forth was developed a number of issues where unaddressed. The ability to provide multi-lingual programs being one of them. This is understandable, as they where developing an *American* Standard. When this standard was accepted by the ISO as an international standard they have asked the ANS Forth committee to address the issues of international programming.

1 Background

When the American National Standards (ANS) X3/J14 committee where considering the ANS Forth standard (X3.215-1994) they took a very "open" attitude to this. When the final document was accepted in 1994 it was probably the most openly discussed standard to date, and most defiantly the most openly discussed Forth standard. There are a number of areas the committee did not address itself to, one of these being '*internationalisation*'. The committee did receive at least one proposal on this matter, but correctly decided that they where dealing with an American standard and thus need not open this particular can of worms.

In 1997 the International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC) accepted ANS Forth as an "International Standard" (ISO/IEC 15145:1997(E)). When accepting the standard the British Standards Institute (BSI) made comments concerning the lack of internationalisation and has asked the X3/J14 technical committee (TC) to address this issue when it reconvenes in 1998. To quote from the "Disposition of Comments Report" (International Organization for Standardization 1997):

> The comments from the United Kingdom concerning Internationalization issues and requirements for embedded systems programmed in Forth will be addressed when the TC reconvenes in 1998. At that time, we will consider these and other needs that have arisen in this evolving technology.

This paper looks at some of the issues that will need to be addressed in the forthcoming TC meeting. A topic of interest to the European Forth community (Borrell 1990; Nelson 1995; Nelson and Shermer 1996).

2 Issues

There are in essence four areas of regional variance.

2.1 Date Format

There are two separate date formats the need to be specified, the 'short' and 'long' formats. There are three areas of difference for the 'short' date format:

1. The "date separator" — the character that

is used to separate the day, month and year fields.

2. The order of the day, month and year fields.
3. The size of the three fields. For the day and month fields we need to know if the leading zero should be displayed. For the year field it is necessary to know if the century should be displayed or not.

Some countries prefer the month to be displayed as a three letter abbreviation rather than a number. This option has not been catered for in the above description.

In the 'long' format the names of the day and month (in both abbreviated and full form) may be displayed, and are thus subject to language differences. There are five additional formatting considerations:

1. Whether the day of month is displayed, if so then whether it is shown with or without a leading zero.
2. Whether the day of week is shown, if so then in abbreviated or full form.
3. Whether the month is shown as a number (with or without leading zero), in abbreviated or full form.
4. Whether the year is shown in full (with century) or abbreviated (without century) form.
5. The order of the fields (day of week, day of month, month and year) including any separating text, such as commas or spaces.

2.2 Time Format

The time format is the least complex of all the variants to be considered. In many ways it is similar to the short date format. There are only five formatting considerations to be taken into account.

1. The "time separator" — the character that is used to separate the hour, minute and seconds fields.
2. The order of the hour, minute and seconds fields.
3. Whether the time is displayed as 24- or 12-hour clock.
4. The pre- and post- meridian indicators.
5. When using the 12-hour clock, whether the meridian indicator is before or after the time.

Technically we should allow for the hour, minute and seconds fields to be displayed with or without a leading zero, however no county (to the authors knowledge) allows the minute and seconds fields to be displayed without a leading zero. The hours field is normally dependent on the method of display, that is to say in the 12-hour clock the leading zero is not displayed, while in the 24-hour clock it is.

2.3 Number Format

There are a surprising number of formatting considerations that need to be taken into account when displaying a number. First let us consider the unsigned integer:

1. The "group separator" — the character that is used to separate large numbers into smaller groups for ease of reading.
2. The size of the groupings.

Whilst is has become traditional in the computing world to display large numbers without thought of formatting, it is traditional in the non-computing world to group large numbers into smaller groupings in order to ease reading. For example, in the UK it is traditional to provide groupings of three digits separated by a comma:

4294967295	computing display
4,294,967,295	non-computing display (UK)

For smaller numbers this is not an issue (numbers < 10000). For larger numbers it becomes important that the user (not necessarily computer literate) can interpret the number correctly at a glance, thus the grouping becomes an issue.

Let us now turn to the signed integer, in addition to the considerations for the unsigned integer, there are:

1. The positive and negative sign symbols.
2. Whether the sign symbol is displayed before or after the number.
3. Whether there is a space between the sign symbol and the number.
4. Whether the positive symbols is displayed for positive numbers.

Note that some countries allow negative numbers to be displayed in parentheses (i.e., "(n)"). This option is not accounted for in the above description.

Moving on to floating point numbers, the following considerations are added to those for a signed integer:

1. The "decimal separator".
2. Whether a leading zero should be displayed for numbers less than 1 or not.

This makes the assumption that the least number of significant digits are shown in the fractional part of the number. Indeed if the fractional part is zero then it not shown at all. I.e., that "1.50" is displayed as "1.5" and that "1.0" is displayed as "1". This assumption may be overcome by specifying a minimum number of decimal places.

Thus the set of considerations necessary to correctly display a (signed floating point) number are:

1. The "group separator".
2. The size of the groupings.
3. The positive and negative sign symbols
4. Whether the sign symbol is displayed before or after the number.
5. Whether there is a space between the sign symbol and the number.
6. Whether the positive symbols is displayed for positive numbers.
7. The "decimal separator"
8. Whether a leading zero should be displayed for numbers less than 1 or not.
9. No of decimal places (*optional*)

Note that in the UK the group and decimal separators are the comma and decimal point respectively. In many European countries these are reversed.

2.4 Currency Format

Most if not all currency systems (currently known to the author) are now based on the decimal system, thus the currency format automatically inherits the nine formatting considerations for floating point numbers in addition to having its own considerations:

1. The "Currency Symbol".
2. Whether the currency symbol is displayed before or after the value.
3. Whether there is a space between the currency symbol and the value.
4. Whether the sign appears before or after the currency symbol.

Some countries allow negative currency values to be displayed in parentheses (I.e., "(£n)"), this option is not accounted for in this description.

Given these thirteen formatting considerations (one could even add a "negatives in parentheses" option, making it fourteen considerations) it should be possible to display any decimal based currency in the format of the target country.

3 Addressing the Issues

A quick survey of existing systems show two fundamental methods of addressing these issues. In Unix/Posix (Sun Microsystems, Inc. 1994) the `environ` function returns a data structure which holds the setting for each configuration option for a given country. The number of configuration options provided for are more limited than those outline in section 2. In Windows ('95 and NT) (Microsoft 1995) and OS/2 (International Business Machines Corporation 1994) a set of special display routines are provided to display the date/time/number/currency in the required format. These routines take all of the identified considerations into account when displaying the object.

In this section a method of addressing these issues is presented which lies between these two extremes.

There are a number of ways the interface could be implemented. One could define the display words as deferred words, resolving to the actual display routine once the location has been selected. The display words could access a jump table to access the correct display routine for the given country code. One could define the words to operate based on an internal data structure, somewhat like the Unix system, changing the values in the data structure when the country location is set. Alternatively, one could extend the `ENVIRONMENT?` table to include the fourteen considerations, and define the display words to operate off this table.

The two table methods have the advantage of allowing for "user defined" environment, that may not have been taken into account in the pre-configured country settings. Provided that the configuration options (formatting considerations) are accessible to the user/programmer.

3.1 Setting Location

A "country indicator" is used to identify the country concerned. This can be either a country code (such as 44 for the UK) or a language string ("en-GB" for "English [UK]"). The language string is more flexible, and allows for Dutch speaking Belgium (nl-BE) and French speaking Belgium (fr-BE). However, it may not be as specific as the numerical codes. Both the country code and the language string systems are supported by ISO standards.

A word is also provided to extract the current country setting in order to allow the application to provide user messages in the appropriate language. (See Borrell (1990) or Nelson (1995) for comments on multilingual programming.)

`SET-COUNTRY` (country —)
: This will take a country indicator and set the display words to display their object in the default format required for the indicated country.

`GET-COUNTRY` (— country)
: Will return the country indicator the system is currently using.

3.2 Date/Time Format

A number of new words are defined to take a parameter (the date or time) and display this in a country specific manner. The parameter may be a day, month, year or a hour, minute, second set or a Julian date/time value.

Using a Julian date/time value will allow the system to produce a single value to indicate the date/time of any item. The date/time display words would interpret this value for display using the current country settings. However, a number of additional Julian value handling words would need to be introduced.

Alternatively, using the day, month, year or the hour, minute, second sets will allow direct interfacing to the current `TIME&DATE` word. Thus the sequence:

```
TIME&DATE .DATE SPACE .TIME
```

will display the current date/time in the format required by the current country setting.

`.DATE` (day month year —) or (Julian —)
: This can be defined to take either a Julian date/time value or a day, month, year set and display the date in the current 'short' date format.

`.FULLDATE` (day month year —) or (Julian —)
: Like `.DATE` this can be defined to take either a Julian date/time value or a day, month, year set and will display the given date in the current 'long' date format.

`.TIME` (hour minute second —) or (Julian —)
: Take a hour, minute, second set or a Julian

date/time value and display the given time using the current format.

3.3 Number Format

Either the current numeric display words (`.`, `.R`, `U.`, `U.R`, `D.`, `D.R`, `F.`) should be modified to display the number according to the format options or, a more viable alternative would be to, define a number of new words to display numbers in a country specific manner:

`.NUM` (n —)
: Display the signed integer n in a country specific manner.

`.RIGHT` (n_1 n_2 —)
: Display the signed integer n_1 in a country specific manner, right aligned in a field n_2 characters wide.

`U.NUM` (u —)
: Display the unsigned integer u in a country specific manner.

`U.RIGHT` (u n —)
: Display the unsigned integer u in a country specific manner, right aligned in a field n characters wide.

`D.NUM` (d —)
: Display the signed double number d in a country specific manner.

`D.RIGHT` (d n —)
: Display the signed double number d in a country specific manner, right aligned in a field n characters wide.

`.FLOAT` (*F:* r —) or (r —)
: Display the floating point number r in a country specific manner.

Note that in all cases the number is automatically displayed in decimal no matter the value of `BASE`. The `BASE` variable is not effected by any of these words.

Where a number is to be right aligned, if the number of characters required to display the number is greater than the number of characters provided then all digits of the number are displayed with no leading spaces in a field as wide as necessary.

3.4 Currency Format

In an attempt to provide a single method of handling all currencies, the currency value should be an integer value treated as a fixed point number with the least significant digit representing the smallest currency unit. I.e., 1 indicates 1 pence, while 100 represents £1 (100 pence).

With this definition in mind we have the following word definitions:

`.$` (n —)
: Display the signed number n as a fixed point currency value in a country specific manner.

`D.$` (d —)
: Display the signed double number d as a fixed point currency value in a country specific manner.

`F.$` (*F:* r —) or (r —)
: Display the floating point number r as a currency value in a country specific manner.

4 Summary

Section 1 provides an introduction to the problem of internationalisation and the Forth standard's position relating international or multilingual programming. Section 2 identifies some fourteen formatting considerations required to support the provision of multilingual application. These relate to the display of the date, time, numbers, and currency.

Section 3 presents a method of addressing the issues identified in the previous section. There are a number of points of interest raised in this section.

1. There are a number of different methods available to implement the interface outlined in section 3, however, the preferred method is the use of a table of the formatting options. The table can be loaded with pre-defined defaults

for a set of known countries. However, if the table is available to the user, it would be possible to define additional environments that are not catered for in the pre-defined setting.

2. A country code is required to identify a set of pre-defined values. This may be either a country code (44 for the UK) or a language/region string ("en-GB" for "English [UK]"). Both methods are supported by ISO definitions. It is not known which is more specific.

3. The concept of using a Julian date/time value to indicate a date/time is assumed in section 3.2. This would allow a system to provide a single value to indicate the date/time and provides a flexible method of handling date/time that is totally system independent.

   It should be noted that the introduction of a Julian date/time value will require a the definition of a number of additional Julian value handling words.

   Section 3.2 provide an alternative definition of the date/time display words based on the current system without the Julian value facility.

4. Currency is defined to be represented as a fixed point integer value, with the least significant value representing the smallest unit of currency. Two words are defined to convert an integer value into the currency display required for the given country.

   A currency display word is also defined which takes a floating point number and displays this as a currency in the format required by the current country setting. This allows floating point numbers to be used as an alternative to the fixed point integer representation.

References

Borrell, R. (1990, September). Deferred Language Translation. In *euroFORML'90 Conference Proceedings*, MPE Ltd, 133 Hill Lane, Southampton SO1 5AF UK. Forth Interest Group.

International Business Machines Corporation (1994). *User's Guide to OS/2 Warp.* International Business Machines Corporation.

ISO (1997). Disposition of Comments Report: ISO/IEC DIS 15145 — Programming Language Forth. `comp.lang.forth`.

Microsoft (1995). *On-Line Manuals.* Microsoft.

Nelson, N. J. (1995, October). Internationalization of Windows applications using Forth. In *euroForth '95 Conference Proceedings*, DELTA t, Adenauerallee 54, D-20097 Hamburg, GERMANY. Forth Interest Group.

Nelson, N. J. and K. Shermer (1996, October). Far eastern Forth. In *euroForth '96 Conference Proceedings*, DELTA t, Adenauerallee 54, D-20097 Hamburg, GERMANY. Forth Interest Group.

Sun Microsystems, Inc. (1994). *On-Line Manuals.* Sun Microsystems, Inc.