



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk**

Training

Internationalization for Distributed Systems

Chris Mayers

Abstract

Organizations wish to deliver software products to international markets.

The technical problems are many, and must address issues of national language, culture, and location. Distributed systems pose no specific new problems, but are likely to suffer the problems in greater degree.

This presentation is a survey of the technical issues, originally given at a workshop for BT on 7 April 1995. It is supplemented by handouts from various publications; it is not intended to stand alone.

APM.1455.00.02

Draft

7th April 1995

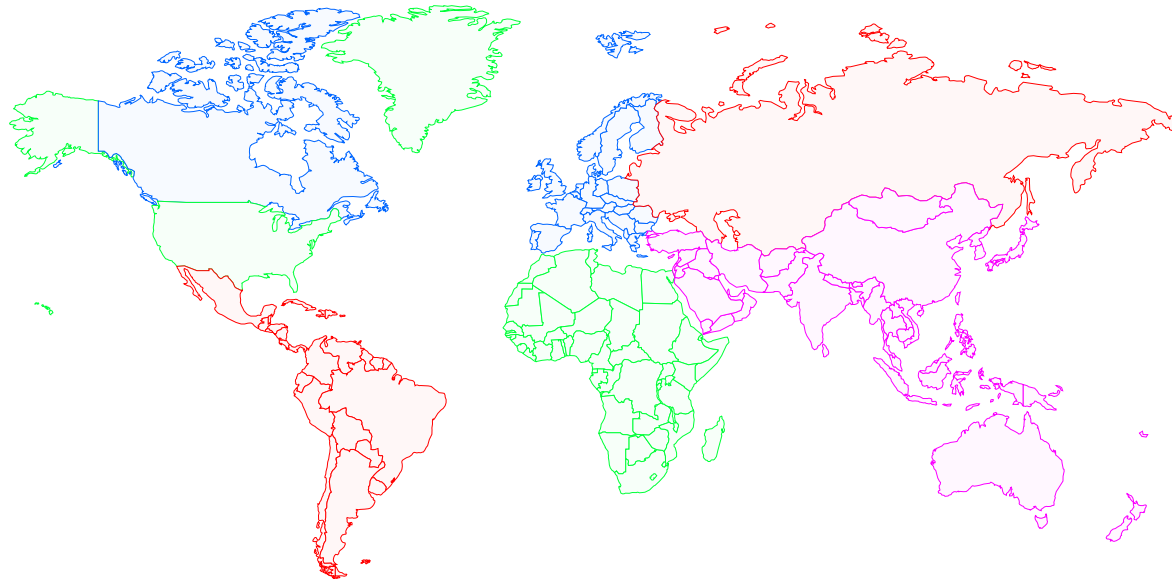
Briefing Note

Distribution:

Supersedes:

Superseded by:

Internationalization for Distributed Systems





In this session

- **Define the purpose and scope of software internationalization**
 - as it particularly affects distributed systems
- **Explain relevant standards**
- **Advise on how to avoid common pitfalls**



What does software internationalization (i18n) cover?

- **Need to consider the impact on**
 - the main user interface (menus, icons,...)
 - on-line help
 - on-line manuals
 - on-line tutorials, samples,...
 - on-line support information (product registration, contact addresses,...)
- **Printed manuals need to be considered too**
 - they can include screen shots of the user interface
 - on-line and printed manuals may be derived from a common source



Why is software internationalization difficult?

- **The issue spans disciplines**
 - it's not just the user interface
- **It has an impact throughout the development lifecycle**
- **Technical expertise is hard to find**
- **Keeping track of relevant standards is time-consuming**
- **Different operating systems tackle the issue in different ways**
 - they use *different* standards...
 - ... and proprietary interfaces too



Disciplines involved in internationalization

- **User interface**
- **Authoring**
- **Operating system, programming language, and tools**
- **Translation**
- **Testing**
- **Project management (to keep a grip on all that lot above!)**



User issues in software internationalization

- **National language**
 - e.g. French
- **Culture**
 - e.g. Canadian
- **Location**
 - e.g. Britain



Locale

- The term 'locale' covers
 - National language
 - Culture
 - ... for example "French (Canadian)"
- In some systems, 'locale' doesn't cover location
 - ... for example, a French Canadian person working in Britain still has a British *location*
- But in many systems it does



Possible levels of internationalization enablement

- **No enablement**
- **Location support only (single language and culture, in many locations)**
- **West European and Latin American languages only**
- **All European, Middle Eastern, and Far Eastern languages**



Basic technical approach to internationalization

- **Separate the neutral components from those that depend on national language, culture, or location**
 - create a base variant, typically in US English
- **For each new locale, localize the non-neutral components**
 - translate the language-dependent components
 - localize the culture-dependent components



Distributed systems issues in internationalization

- **Portability**
 - of an application, across different operating systems and hardware platforms

- **Interoperability**
 - between applications, on different operating systems and hardware platforms

- **Interchangability**
 - of information between applications



Reversed assumptions and internationalization

- **Single language -> Multiple languages**
- **Single version -> Multiple versions**
- **Single standard -> Multiple standards**



Technical issues in internationalization

- **Character set and language**
- **Colour**
- **Sound**
- **Time**



Standard character sets

- **ASCII (7-bit): ISO 646**
 - no accented characters
- **Latin 1 (8-bit): ISO 8859-1**
 - accented characters
 - currency symbols
 - supports many West European and Latin American languages
- **Unicode (16-bit): ISO 10646-1**



Proprietary character sets

- **IBM calls them 'code pages'**
 - **Code page 437 (English) is the default for DOS**
 - **Code page 850 (Multilingual Latin 1) is one alternative...**
 - **...but it is *not* the same as ISO 8859-1 Latin 1**
 - **... this causes problems transferring text between DOS and Windows**
- **Other manufacturers have their own character sets too**



Character set issues

- **Sorting (collating sequence)**
- **Upper-casing**
- **Conversion with other character sets (folding)**
- **... all of these depend on language *and* culture**
- **When text is stored, the associated character set must be recorded too**
 - **e.g. in the database**



Language support

- **The user must be able to enter text in their own language**
 - even if the user interface is in another language
- **Supporting multilingual text is rather harder, even within one character set**
 - each language needs separate dictionary, thesaurus, hyphenation, grammar,...
 - ... user may need to tag text accordingly
- **Arbitrary multilingual text has additional problems**
 - right-to-left and left-to-right text on the same line?



Colour issues

- **Many issues to consider**
 - people with visual impairments
 - colour perception phenomena
 - viewing conditions (background lighting)
 - cultural factors
 - support for monochrome/grey scale screens
 - colour printing (if required)
- **Colour should be generally under user control**
- **Colour should be used to reinforce; it should never be the only cue**



Colour in Microsoft Windows

- **Minimum is nowadays VGA (16 colours, 640x480 resolution)**
- **Users control colour via the Control Panel**
 - **'Desktop' icon: patterns and wallpaper**
 - **'Color' icon: colour schemes**
- **Users cannot control the colour of images (or icons)**



Location issues - examples

- **Postal addresses**
- **Telephone number formats**
- **Currency**
- **Preferred time and date formats**
- **Weights and measures**
- **Timezone**
- **... and application-specific needs**



Location in Microsoft Windows

- **Some support for location**
 - for example, time and date formats, but not telephone number formats
- **Only supports one location at once ('here')**
 - users select this via the 'International' icon in the Control Panel...
 - but think about a multi-currency spreadsheet, or a multi-national address book...
 - ...support for multiple target location formats is application-specific
- **Extensions to the basic Windows API must cope with location issues**
 - for example in the Telephony API (TAPI)?



Culture issues - examples

- **Language usage (French/Canadian French)**
 - ... consistency with other components may be preferable to strict correctness
- **Icons**
- **Samples in on-line tutorials**
 - for instance, avoid any reference to food and drink!



Culture in Microsoft Windows

- **No specific support for applications**
- **Applications must implement their own culture support, if required**
 - **depending on the locale setting**



Internationalization support - DOS and Windows

- **DOS**
 - minimal, and difficult to use
- **Windows 3.x**
 - much better, but badly documented
- **Windows NT**
 - better still - the Win32 interface uses Unicode
- **Windows 95**
 - also supports Win32, hence Unicode



Internationalization support - other desktops

- **OS/2**
 - wide range of other code pages supported, for IBM mini/mainframe compatibility
- **Macintosh**
- **Unix**
 - X/Open NLS and OSF/Motif
 - improved support in X11R6



Internationalization support for the programmer

- **Programming languages are still written in ASCII**
- **Most of the localization support is via libraries**
 - **C has some support for locale, but insufficient to be usable...**
 - **...use the OS libraries/application frameworks, even though this sacrifices portability**



Internationalization support - tools and components

- **Most applications are now built from a mixture of components, each providing part of the user interface**
 - standard system libraries or 'application frameworks'
 - ISV plug-in software components (e.g. VBXs)
 - database tools (e.g. Oracle Forms)
 - scripting languages (e.g. Visual Basic)
- **The user should see a consistent application**
 - every component must be translated and localized consistently...
 - ... even if this has to be done in different ways for the different kinds of component
- **This will constrain your choice of components and vendors**



Internationalization support for the translator

- **Translators get paid by the word**
 - they want to use the tools they already know
- **Windows developer tools tend to be unfamiliar and unsuitable for translators**



Helping your translators

- **Brief them on the features and functions of the software**
 - a marketing brochure isn't enough
- **Provide a glossary of concepts for your particular software**
- **Write them a translation road-map for guidance**
- **Get them to use the software, if you can**



Internationalization in the real world

- **Things change...**
 - UK telephone national codes
 - Dates of national holidays
 - National currencies, maybe?
 - National borders
- **Define which real-world changes your application is designed to cope with**
 - and also the ones it isn't



Upgrading a language variant

- **A localized product is a *variant***
 - when the base version is upgraded, all the language variants need to be upgraded correspondingly if the user interface has changed
- **Upgrading a variant requires merging the changes from...**
 - old base version -> new base version
 - old base version -> old language variant
- **... this merge is always a painful process!**
- **Minor version changes (bug fixes, updates) must be kept isolated from internationalization variants**
 - otherwise, how can you issue an emergency bug fix to the English, French, Dutch, Japanese,... variants simultaneously?



General advice - user interface

- **Exploit previous corporate experience**
- **Get involved with internationalizing a specific application if you can**
- **Get the base version right first**
 - **apply conventional user interface best practice**
- **If you have access to a usability lab, get them involved too**
- **Get feedback from paying customers**
- **Use the application yourself**



General advice - standards

- **Define corporate standards for internationalization**
 - by reference to external standards
- **Clarify what the application programmer must do achieve compliance with the standards**
 - omit the things that happen automatically
- **Back up these standards with special tools that check compliance**
 - buy these tools if you can, build them if you must...
 - ...a little checking goes a long way
- **Clarify what the test engineer must do to validate compliance**



General advice - tools and operating systems

- **Don't try to bypass the system's limitations on internationalization**
 - it will cause more problems in the long run
- **Figure out and document what the precise limitations are**
 - this may mean carrying out systematic tests
- **Make sure applications stay within these limits**
 - special checking tools can help
- **Apply pressure to the vendors**
 - to remove the limitations (eventually!)



Summary

- **Be very clear about the objectives and the costs involved in internationalization**
 - agree a realistic level of ambition...
 - ...don't offer or expect perfection
- **Consider the costs of maintaining internationalized variants**
- **Form a good team!**



Books and journals - DOS and Windows

- **IBM**

- *try National Language Design Guide, Designing Enabled Products, Volume 1, Second Ed., IBM reference SE09-8001-01*
- *... also National Language Support Reference Manual, Volume 2, Second Ed., IBM reference SE09-8002-01*

- **Windows**

- *try Microsoft Systems Journal (MSJ) - technical subscription journal*
- *e.g. June /July 1994 issues...*
- *...well-written, but covers Microsoft products only*



Books - general

- **Digital**

- *try Developing International User Information, Scott Jones et al., Digital Press 1992, ISBN 1-5555-8-084-X*

- **Macintosh**

- *try Guide to Macintosh Software Localization, Apple/Addison-Wesley, ISBN 0-201-60856-1*



More information?

- **Internet newsgroups**
 - [comp.std.internat](#)
 - [comp.software.international](#)
- **World-Wide-Web sites**
 - <http://www.unicode.org>
 - <http://www.iso.ch>
 - <http://www.microsoft.com>