

SIGCHI Seminar

Software Globalization

Ron Broome

rbroome@onerealm.com



Overview

- **Terminology** - *What is the problem of “globalization” (G11N)? Why is globalization important?*
- **Problems** - *What are some of the general issues? What are particular issues for user interfaces?*
- **Solutions** - *How can these globalization issues be addressed? What tools are available to help?*

Terminology

- **Locale** - Features of the user's environment (OS) that depend upon (natural) language, country or cultural conventions
- **Character** - The smallest element of a writing system or script
- **Character set** - An ordered set of characters represented by a numeric encoding scheme (**CharSet**, code page)
- **Font** - A graphical representation of characters

Terminology

- **Literal** - In source code, fixed text represented in double quotes (strings), single quotes (character), or by a numeric value
- **Resource** - A representation of GUI components, text literals, icons, bitmaps, cursors, etc. maintained externally from source code, bound to an application for use at runtime

Character Set Encoding

- **Single-byte (SBCS)** - A character set encoding, such as ASCII, in which each character is represented by **one** byte. Limited to 256 characters for 8-bit chars.
- **Multibyte (MBCS)** - A character set encoding, such as Shift-JIS, in which each character may be represented by one or more bytes.
- **Wide Character** - A character set encoding in which each character is represented by a fixed width (16-bit, 32-bit).
- **Unicode** - A 16-bit wide character set encoding containing a unified representation of the world's characters (ISO 10646 specifies code equivalent, also UTF-8 multibyte version)

Problem Addressed

"It's just a linguistic process! Why does it take so long and cost so much!"

-Bill Gates

*"We found that the 'linguistic process', was only **one-third** of what we did. We were spending **most** of our time on substantial retesting, language-specific functionality and 'remedial engineering' because of oversights in the development process."*

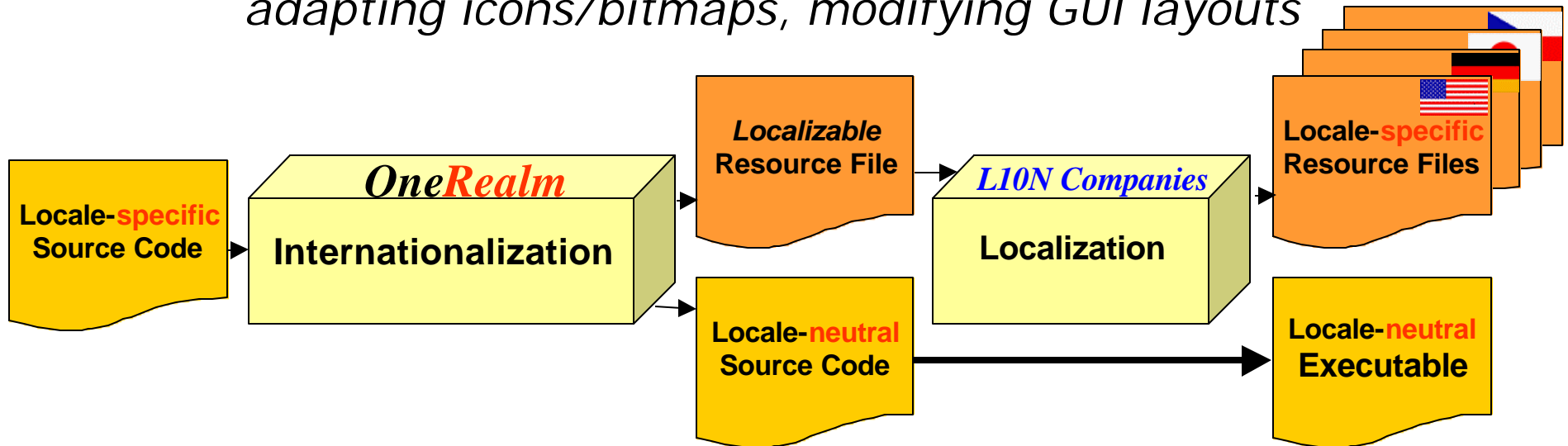
- David Brooks

*Microsoft's Senior Director of International Product Strategy
[quoted in article: "Microsoft's Localization Strategy" in
Volume 9 Issue 5 of Multilingual Computing]*

Problem Addressed

What is the **globalization** software development process?

- **Internationalization** (I18N) - preparing software that is locale-neutral
- **Localization** (L10N) - translating the text strings, adapting icons/bitmaps, modifying GUI layouts



Why is this important?

- **THE NEXT BIG PROBLEM!!!!** - Developing applications that can be used regardless of the native language of the user.
- **GLOBALIZATION IS NOT A FEATURE!!!!** - Preparing applications for the world is not easily done at the *end* of a project.
- **THIS IS AN INTERFACE ISSUE** - Handling the interface issues of locale is the fundamental problem of globalization.

Approaches to Internationalization

C/C++

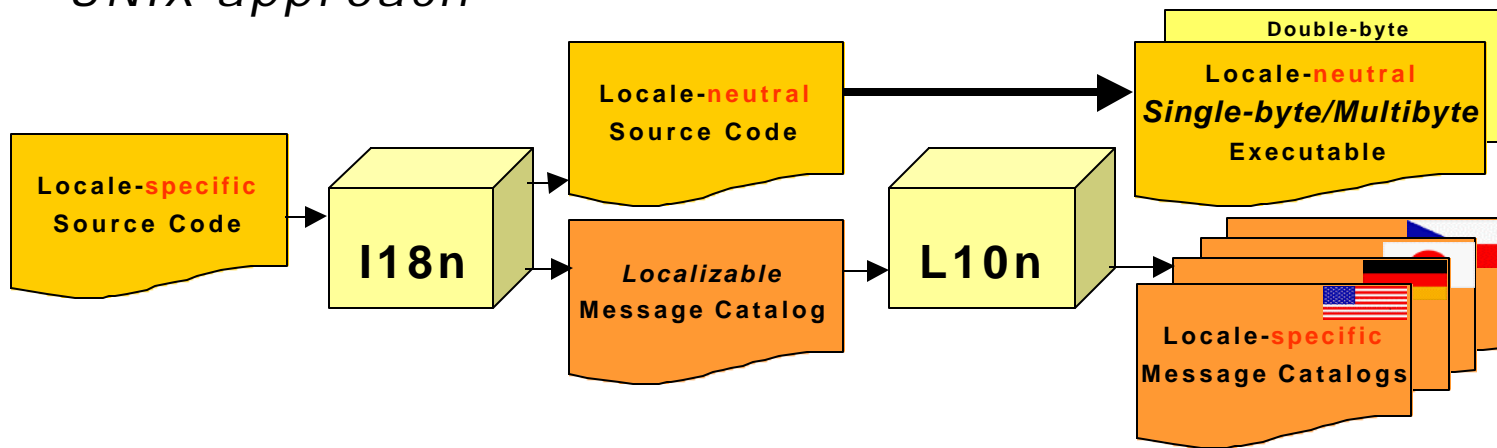
- **Multiple source code/multiple executables approach** —
Bad, expensive
- **Single source code/multiple executables approach** —
Even better, less expensive — MS approach
- **Single source code/single executable approach** —
Best, least expensive in the long-term

Java

- **Single source code/PropertyResourceBundle** —
Easiest, text property text files, problematic for web applets
- **Single source code/ListResourceBundle** —
Better execution speed, compiled byte code, harder to maintain

C/C++ Single Source Approach

UNIX approach



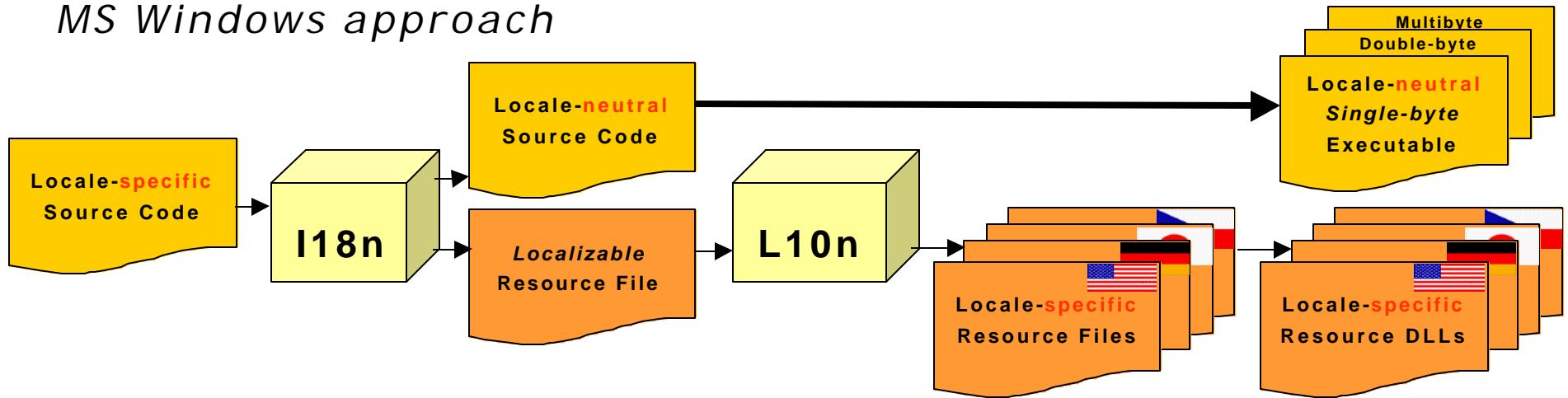
XPG specification

single-byte data types = **multibyte data types** (char)
single-byte functions = **multibyte functions** (strchr)

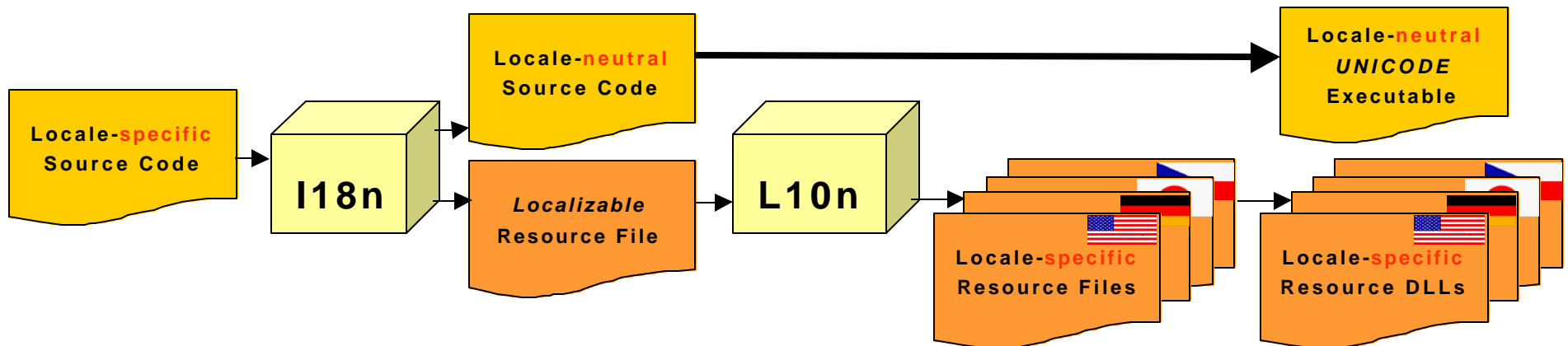
wide character data type (wchar_t)
wide character functions (wcschr)

C/C++ Single Source Approach

MS Windows approach

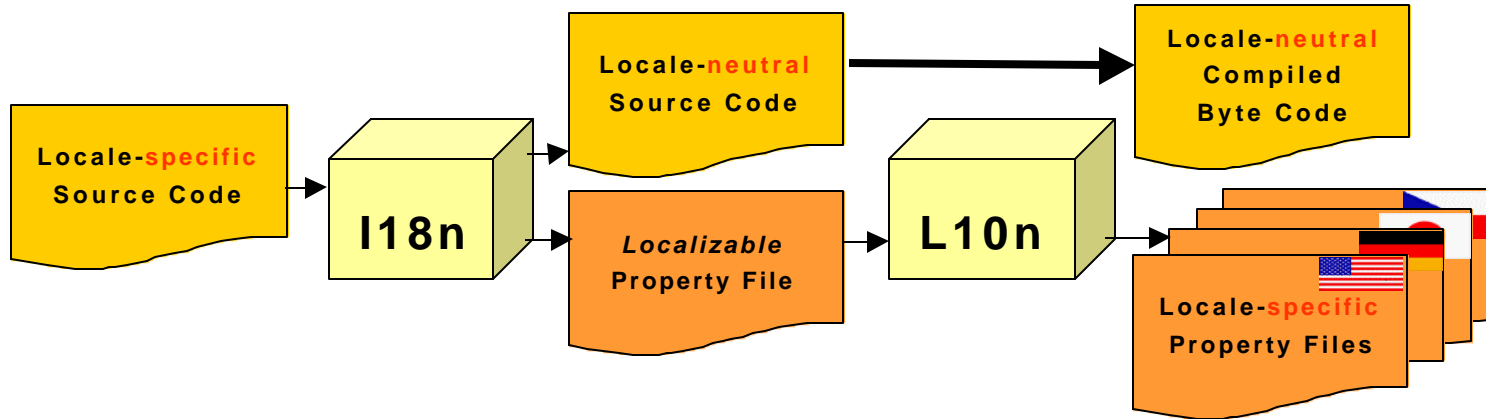


MS Windows approach with UNICODE

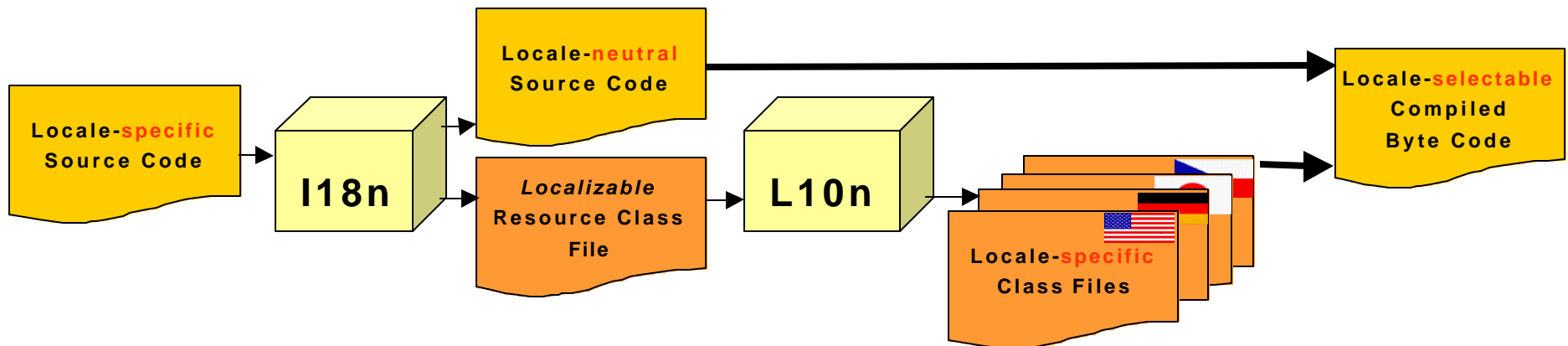


Java Single Source Approach

PropertyResourceBundle approach



ListResourceBundle approach



GUI Internationalization

What does locale-neutrality mean for the GUI?

Text strings	Externalization of local-sensitive embedded strings “ ” from both function calls and variable initializations
Character literals	Externalization of local-sensitive embedded character literals ‘ ’ from both function calls and variable initializations
Currency formats	Externalization of currency formatting \$ ¢ ¥
Numeric formats	Externalization of numeric formatting . ,
Time/date formats	Externalization of time or date formatting 3/1/99 1-3-99
Weights and measures	Externalization of weights or measures formatting lbs. mi. kg.
Mailing addresses	Externalization and re-coding of address formatting
Color usage	Removal of hard-coded colors
Icon/image usage	Removal of hard-code icons/images

GUI Internationalization

Fonts	Remove hard-coded font specifications
Character height	Mostly a localization issue, but be on the lookout for hard-coded character height changes or allow enough room for any language
Tracking/kerning (spacing between characters)	Mostly a localization issue, but be on the lookout for hard-coded kerning changes or allow enough room for any language
Diacritical marks	Mostly a localization issue, but be aware of spacing requirements
Leading (line spacing)	Mostly a localization issue, but be on the lookout for hard-coded leading changes or allow enough room for any language
Keyboard shortcuts and accelerator keys	Mostly a localization issue, but be aware of hard-coded values in source code
File/path names	Externalization and re-coding of pathname usage
Paper sizes and page layout	Should be user selectable
GUI layout	Mostly a localization issue, but be on the lookout for hard-coded resizing or repositioning of GUI components

Other Coding Issues

What about algorithms and coding practices that are locale-sensitive?

Sorting/collating	Usage of locale-sensitive sorting/collating functions
Parsing	Parsing is often very locale-sensitive and may need to be handled in locale-sensitive source code
String extraction	Often difficult and may need to be handled in locale-sensitive source code
String concatenation or formatting	Long formatting strings or string concatenations should be avoided
Search and replace	Often difficult and may need to be handled in locale-sensitive source code
Case conversion	Not used in all written languages
Text delimiter (spaces, punctuation)	Not used in all written languages
Hyphenation	Not used in all written languages
Syllabification	Not used in all written languages

Role of Tools

- **Design**

Library Products: Provide out of the box APIs for commonly required I18N functionality

- **Code/Implementation**

Source Code Analysis: Automate the detection and correction of the problems in the source

Role of Tools

- **Localization**

Localization Process control including multiple-translator access, leveraging of previous translations

- **Translation**

Automatic/Assisted Translation tools

- **Test**

Test tools

Internationalization Tools

- **Library Tools**
- **Source Code
Analysis/Correction/Management Tools**

Library Tools

- Important attributes:
 - Text encoding conversion
 - Handle multiple encodings
 - Text manipulation for required encodings
 - Uniform encoding/platform API
 - High-level objects, i.e. Date, Time, Currency, etc.

Library Tools

Product	Company	www.
Basis	Basis Technology Corp.	basistech.com
Global C	Uniscape	uniscape.com
ICU	IBM	ibm.com/java/tools/ international- classes/
GEL	OneRealm	onerealm.com Q3/99

Analysis/Correction/Management Tools

- Analysis for detection of I18N errors
 - Text strings/character literals etc. **displayed to user**
 - Unsafe function detection: **GUI functions**
 - Unsafe coding construct detection
 - Programming languages supported

Analysis/Correction/Management Tools

- Correction
 - Automatic correction of errors
 - Guidance for programmer intervention
 - Overriding non-errors for “reprocessing”
 - Configurability/extensibility
- Management of internationalization
 - Progress statistics
 - Resources
 - Code

Analysis/Correction/Management Tools

Product	Company	www.
Global Checker	Uniscape	uniscape.com
I18N Expeditor	OneRealm	onerealm.com

Application Localization

- Translation Tools
- Localization Tools

Translation Tools

- Important attributes
 - Glossary
 - Automatic conversion
 - Alignment/formatting
 - Translation memory
 - Documentation translation

Translation Tools

Product	Company	www.
Trados	Trados	trados.com
TransLINGO	Fujitsu S/W	fsc.fujitsu.com/language
Transcend	Transparent Language	transparent.com
HtmlQA	Translation Craft	tcraft.com
MLTS	CIMOS	??

Localization Tools

- Important attributes
 - Project glossary
 - Multi-user environment
 - Geometry management
 - Context-sensitive leveraging

Localization Tools

Product	Company	www.
Corel Catalyst Jargon	Corel Corporation MDR Telemanagement Ltd.	catalyst.corel.ie mdr.com
Global Xchange	Uniscape	uniscape.com
SDL Workbench	SDL International	sdlx.sdlintl.com
Locale	Accent Software Int.	accentsoft.com

- ***I18n Expeditor*** - GUI tool for assessing and correcting ***internationalization*** problems in source code
- ***I18n Reporter*** - Command-line tool for assessing problems in batch mode
- *Internationalization issues that can be addressed with OneRealm's tools*

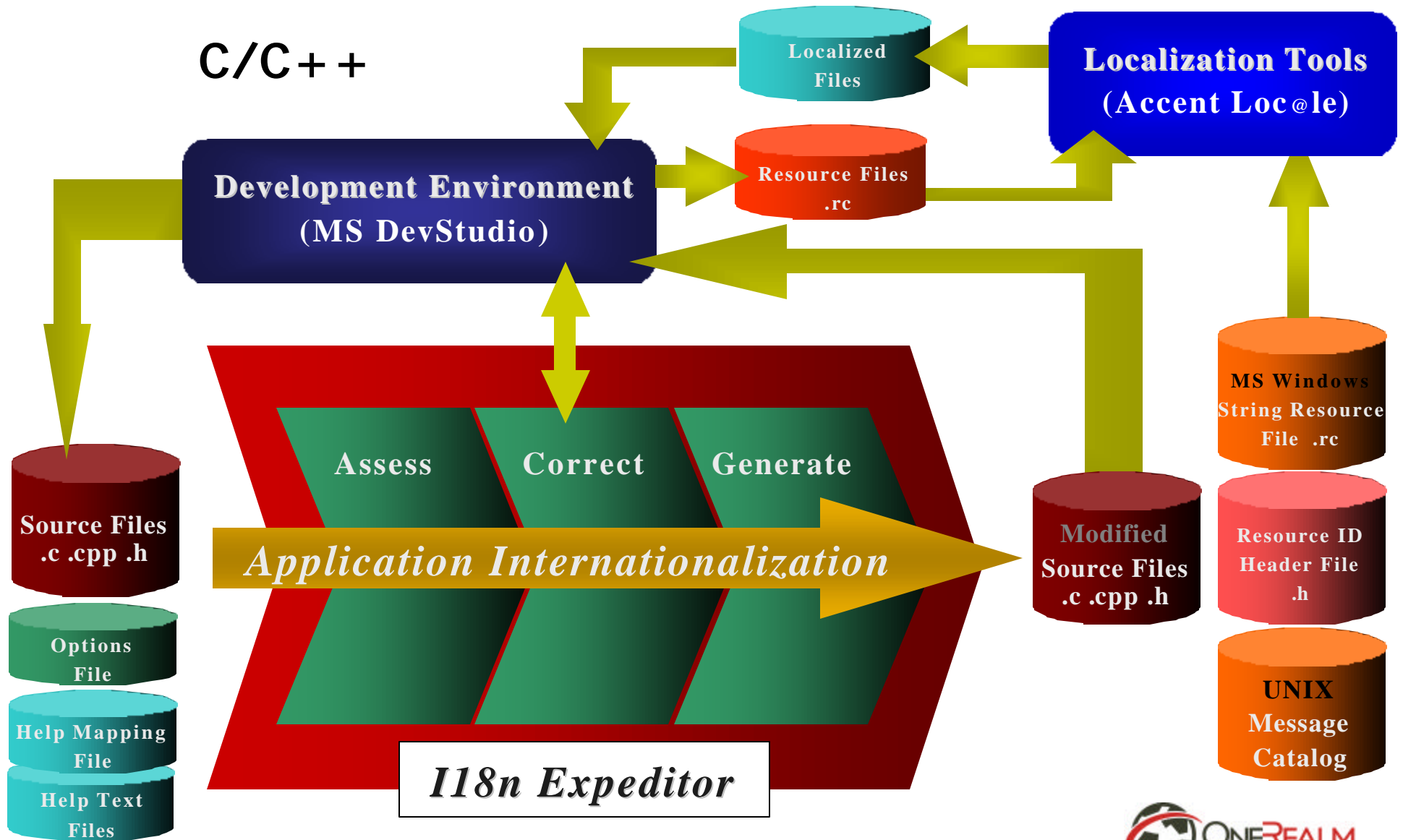
I18n Expeditor Overview

- **Assesses your *source code* per developer configurations**
 - *Focuses on “at-risk” lines of code*
- **Automatically *corrects* internationalization problems**
 - *C/C++ solutions (MS Windows, UNIX, OneRealm)*
 - *Java solutions (ListResourceBundle, PropertyResourceBundle)*
 - *User-defined C/C++ and Java solutions*
- **Or, *provides suggestions* for fixes**
 - *Detailed context-sensitive help for potential I18N violations*
 - *Extensible and modifiable for enterprise-wide solutions*
- **Generates *locale-neutral files***
 - *Modified source code files and header files*
 - *Externalized resources (C/C++ RC files, XPG message catalogs; Java ListResourceBundle, PropertyResourceBundle)*
 - *User-defined resource output formats*

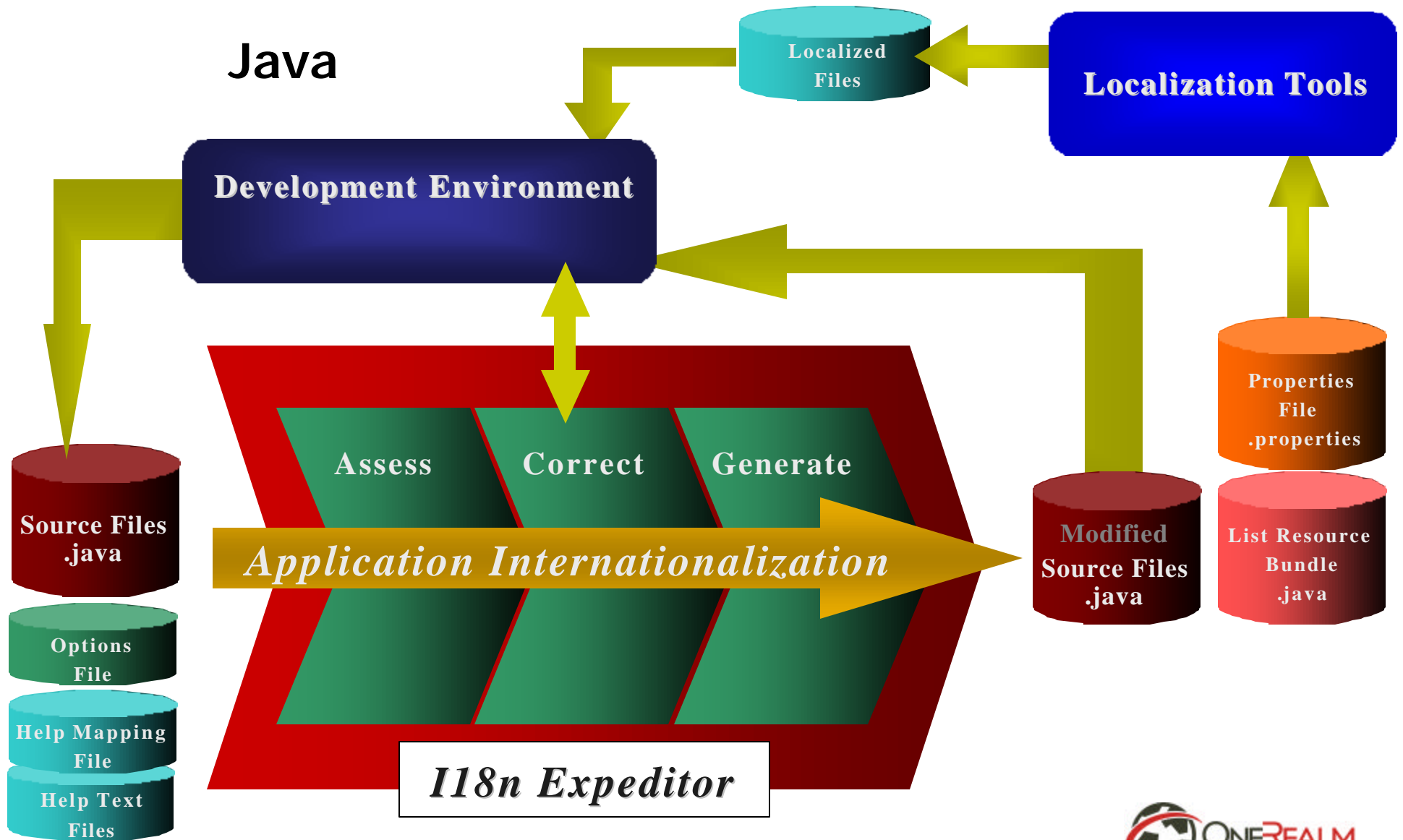
Supported Platforms & Languages

- *Source code assessment for these languages:*
 - **C/C++**
 - **Java**
- *Source code assessment can be made*
independent of platform
- *Executables for these platforms:*
 - **MS Windows 95/98/NT** (I18n Expeditor *and* I18n Reporter)
 - **Sun Solaris** (I18n Reporter *only*)

I18n Expeditor Work Flow



I18n Expeditor Work Flow



Assessment

I18n Expeditor *and* **I18n Reporter** *evaluate source code per your configuration:*

- **Files imported** *directly*, or for C/C++ from MS DevStudio (5.0, 6.0) projects
- **Options file and dialog** allows assessment with your *compiler/project settings*

I18n Expeditor *and* **I18n Reporter** *detects the following potential I18N violations:*

- **Embedded text:** *string literals, static strings, character literals*
- **Problem I18N functions:** *unsafe C/C++, locale-sensitive Java, deprecated Java*
- **Functions with** *locale-sensitive and/or locale-insensitive arguments*
- **C/C++ macros** *with embedded I18N problems*
- **String patterns** *which are locale-sensitive or locale-insensitive*
- **C/C++ character array** *pointer arithmetic and array references*

Correction

I18n Expeditor *assists you in the correction of your source code:*

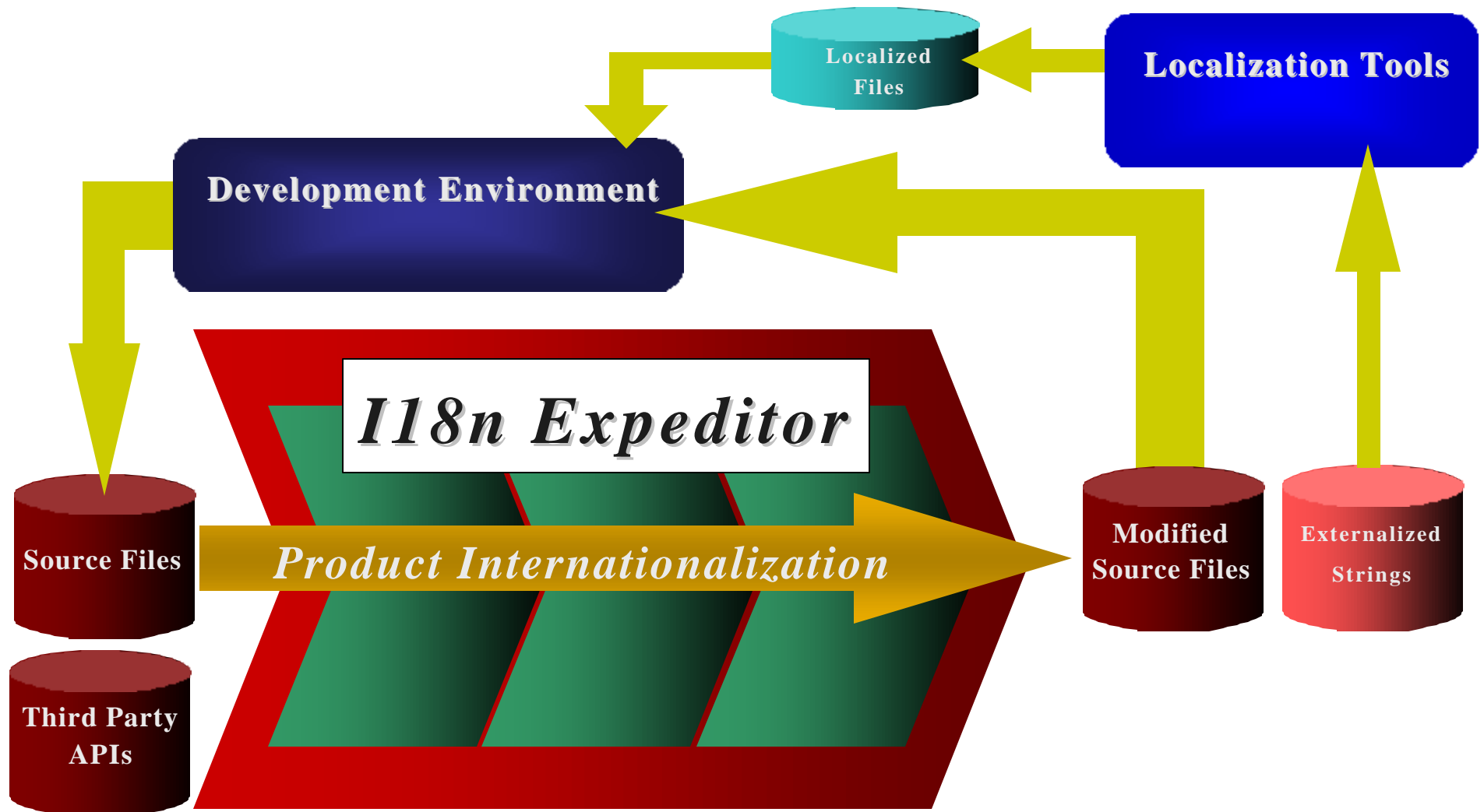
- *Externalize string literals per your configuration*
 - **C/C++ functions** *MFC CString constructor, UNIX catgets function, OneRealm functions*
 - **Java methods**
 - **User-defined functions or methods**
- *Externalize character literals*
 - **C/C++ OneRealm function**
 - **Java functions**
 - **User-defined function**
- *Describes solutions from I18N problem context-sensitive help*
 - **Problem descriptions**
 - **Function/method prototypes and header files**
 - **Example code for problem and solution**
 - **Standard (ANSI, Sun) and platform-specific notes**
 - **Useful as a tutorial and a reference**
- *Project management*
 - **Mark problems for later review**
 - **Override non-problems**

Code Generation

I18n Expeditor generates files for your localization phase:

- *Modified source code*
 - **Inserted C/C+ include file statements**
 - **Replacement C/C++ function calls for text externalization**
 - **Replacement Java methods calls and method definitions for text externalization**
 - **User-defined C/C++ replacement functions and Java replacement methods**
 - **User-defined comments for overrides, marked and fixed code**
- *C/C++ string resource ID header file*
 - **Macros for unique string resource IDs**
 - **User-defined macro templates for resource IDs**
- *Java resource access class file (implements externalization methods)*
 - **Unique string resource identifiers**
 - **User-defined identifier templates**
- *String externalization files (localization files)*
 - **C/C++ MS Windows resource files, UNIX XPG message catalog formatted files**
 - **Java property files or ListResourceBundle class files**
 - **User-defined file formats (e.g., for database imports, custom externalization)**

I18n Expeditor Summary



Summary

- *Software Globalization (G11N):*
Internationalization (I18N) vs Localization (L10N)
- *Issues of internationalization:*
GUI, Coding algorithms
- *Tools:*
Internationalization, Localization
- *OneRealm's tools* for assessing and correcting internationalization problems in source code