White Paper

International String Class

Tivoli Internationalization

David Kumhyr

Contents

1. Preface

This document describes the Tivoli International String Class (IString) and its associated user interface controls. The IString class should be used for user entered text in all Java based Tivoli products.

Comments and criticisms are welcomed, please pardon my Imperial English[1] spellings. Please address comments to the author:

David B. Kumhyr
dkumhyr@us.ibm.com
+1 512/838-4564

[1] While spelling this paper is Imperial English all programmatic terms and APIs use US English spellings.

1.1 Acknowledgements

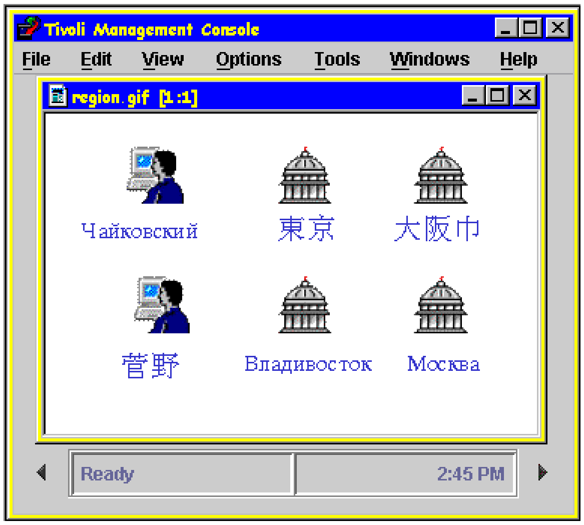
I would like to extend my appreciation to the following people who provided assistance and information that enabled the development the IString class and it's associated controls. They are listed not in order of importance, but by their English collation order:

David Hetherington, IBM/Tivoli Austin, Texas
John Linton, IBM/Tivoli Research Triangle Park, Nort Carolina
Kenneth A. McKethan, IBM/Tivoli Research Triangle Park, Nort Carolina
Joe Ross, IBM/Tivoli Austin, Texas
Yoshikazu Sato, IBM Yamato, Japan
Keiichi Yamamoto, IBM/Tivoli Austin
Stanford Yates, IBM/Tivoli Research Triangle Park, Nort Carolina

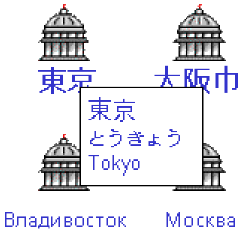
2. Executive Overview

In the current world business environment handling different languages and character sets is an everyday problem. Tivoli's enterprise management products are designed to support the business enterprise model. Handling multiple languages and character sets has become become our problem too. With IString we are equipped to address these problems in a simple and effective manner that none of our competitors can match. IString provides significant competitive advantage for our products over all other enterprise solutions.

The simulated Tivoli policy region below depicts a real scenario for many of our customers. While there are some problems with multi-language TMRs using romance languages, even basic character recognition becomes impossible with non-Latin character sets.[2]



The International String Class (IString) ToolTip allows the users to see an automatic transliteration of the kanji region labels into English. [3]



[2] Which region is Tokyo and which is Yokohama?

[3] IString provides an automatic transliteration based upon Hepburn phonetics for Japanese into English. Currently IString suports transliteration from Japanese, Korean, Russian, Ukranian. Transliterators for Arabic, Hebrew, Chinese are in development.

3. IString Solutions

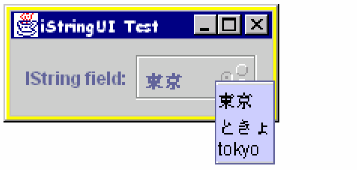
IString solves many other problems encountered in displaying and using multiple human languages in multinational distributed computing environments. The following scenarios are exemplars of these problems and solutions.

Problem 1: A user cannot recognise a label written in a different character set.

An English speaking Tivoli Administrator in New York needs to modify an administrator in the Moskva region named Давид Кумгир.

A help desk operator is assisting a user in the Tokyo region. Using Tivoli Remote Control he has switched the language to English[4], but the region icon labels were created in Japanese. Find the 東京[5] region.

Solution: IString labels contain automatically transliterated information. This information enables viewing of languages who's symbolic representations are not normally understandable by speakers of romance languages.



IString field: 東京

東京
ときょ
tokyo

IString controls enable viewing the transliterations by ToolTips or switching between viewed fields either user activated or program activated.

Problem 2: The characters representing a word have several different readings, so even if you can read the character there is no context to determine what it means.

西 [6]can be read as either West or Spain.

Solution: Contextual data may be stored in the IString associated with the display field.

[4] On the fly language switching is now functional in Presentation Services. Applications may change the language and locale either by user choice or a message from another application.

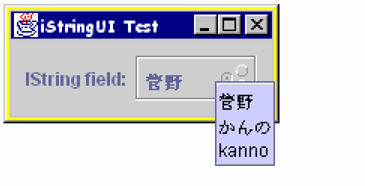
[5] Tokyo /u6771 /u4eac

[6] 西 character is /u897f, is either West or Spain.

Problem 3: Many kanji characters have several pronunciations. It is often not possible to reliably pronounce or sort a Japanese name without previous knowledge of it.

As an operator in a call centre you need to call 苅 野さん but you don't know the pronunciation of the name.

Solution: The sortField can be used to store and display the pronunciation an IString ToolTip or a rotate control can be used to display the information.



IString field: 菅野

菅野
かんの
kanno

Problem 4: You have a list of names and need to alter the collation sequence or create a second order list.

You have a list of users and wish to display them alphabetically, except you want the CEO's name at the top.[7]

Solution: The sortString field of IString to store the order altering information without disturbing the display quality information of the label in the baseString field. This characteristic may also be used for custom collation order or overcoming deficiencies in lexical sort order based on character sets.

Problem 5: Obtaining the correct casing between PC, UNIX and mainframe character sets, or monocasing for performing database searches.

Items inserted into a database must be searched multiple times to find a match. "De Brawer", "de Brawer" and "DE BRAWER" are composed of different characters.

Solution: IString's sortString can be used to store a lexically mapped version of the baseString for database searches.

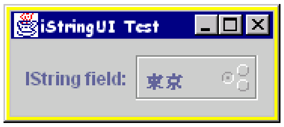
[7] An example was the Tivoli Lotus Notes mailing list where "_Frank Moss" was used as a name using the underscore character to 'float' the name to the top of the list.

4. IString User Interface Controls

IString was initially designed and implemented in Java. There is an IDL class defined to enable the IString data to be stored and used in a CORBA environment. Implementing the class and controls in C or C++ would not be difficult.

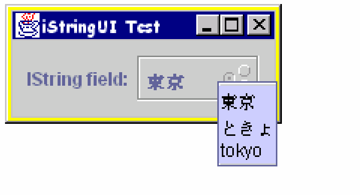
IString data can be used with any AWT or Swing control that implements the getText() and setText() methods. It is very easy to use the custom controls designed for use with IString data.

The State control (also known as a rotate control) offers a representation of the available states of the data contained. A series of radio buttons are used to show the position of the control and provide a visual cue to the currently displayed field.



IString field: 東京

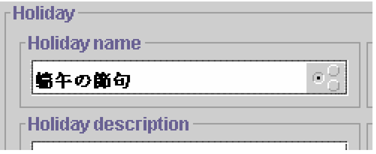
IString controls also display ToolTips differently. All fields of the IString data are loaded into the ToolTip for display as shown below. 'Ordinary' AWT and Swing controls may also display IString ToolTips when they are associated with an IString Controller, which uses Java introspection. The IString Technical Reference discusses use in more detail.



IString field: 東京

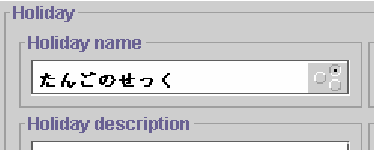
東京
ときょ
tokyo

Rotate controls may also be editable as shown below in a series of examples taken from the Javelin international calendar holiday dialogues.



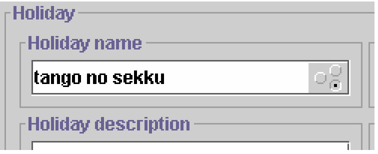
Holiday name: 端午の節句

Holiday description:



Holiday name: たんごのせっく

Holiday description:



Holiday name: tango no sekku

Holiday description:

5. How IString Works

The IString class is a datatype with associated methods and a set of user interface controls. The IString datatype is composed of the following fields:

Field	Type	Purpose / contents
baseString	String	The user entered data, serves as the basis for transliteration.
sortString	String	An intermediate representation used for sorting and collation.
altString	String	Final transliteration, an English transliteration.
sourceLocale	String	ISO code for source locale.
sourceLanguage	String	ISO language code for source language.
sourceVariant	String	Source variant code.
targetLocale	String	ISO code for target locale.
targetLanguage	String	ISO language code for target language.
targetVariant	String	Target variant code.

Below is an example of the data stored in the IString data type for the example problem number 3, Japanese family names. The baseString field stores the final representation of the name in kanji characters, the sortString field contains an intermediate representation that can be used for Japanese collation and pronunciation. The altString field contains a transliterated English version of the name.

Field	Type	User data
baseString	Java String	苅 野
sortString	Java String	かんの
altString	Java String	kanno
sourceLocale	Java String	JP
sourceLanguage	Java String	ja
targetLocale	Java String	US
targetLanguage	Java String	en

Data entered into an IString, either by programmatically invoking a constructor or by a user entering data into an IString field control is transliterated and stored based upon a combination of the source and target languages and locales.

In the above example the data is Japanese so input is gathered from the IME and fed into the different fields. If the language was Russian to English (where there is no IME) the transliteration is lookup table based. More detailed information on the workings of IString can be found in the IString Technical Reference White Paper.

6. Appendix A – Further Reference

Tivoli White Paper: ***IString Technical Reference***, David Kumhyr, 1999

6.1 Patent Applications Filed with the US Patent Office

IBM Docket	Title
AT9-1998-0108	Method, System, and Computer Program Product for Providing a User Interface with alternative Display Choices
AT9-1998-0158	Method, System, and Computer Program Product for Capturing Language Translation and Sorting Information into a Text String Class
AT9-1998-0159	Method, System, and Computer Program Product for Sorting Text Strings
AT9-1998-0160	Method, System, and Computer Program Product for Storing Transliteration and/or Phonetic Spelling Information in a Test String Class
AT9-1998-0408	Method, System, and Computer Program Product for Rotating Through a Sequence of Display States in a Multi-field Text Class in a Graphical User Interface
AT9-1998-0409	Method, System, and Computer Program Product for Controlling the Graphical Display of Multi-field Text String Objects
AT9-1998-0410	Method, System, and Computer Program Product for Displaying the Contents of all Fields in a Multi-field Text String Object
AT9-1998-0411	Method, System, and Computer Program Product for Dynamic Language Switching in a Multi_field Text String Object via Messaging
AT9-1998-0413	Method, System, and Computer Program Product for Case and Character-Encoding Insensitive Searching of International Databases
AT9-1998-0578	Method, System, and Computer Program Product for Automatic Character Transliteration in a Text String Object
AT9-1998-0843	Method, System, and Computer Program Product for Transferring Human Language Data Across System Boundaries
AT9-1998-0844	Method, System, and Computer Program Product for Storing Bidirectional Language Data in a Text String Object for Display on on Non-Bidirectional Operating Systems