

# StarOffice 6.0

## A Multi-lingual Office Suite

*Technical Overview*

Dieter Loeschky, *Staff Engineer*

Shanmugam Senthil, *Staff Engineer*



Sun Microsystems, Inc.  
901 San Antonio Road  
Palo Alto, CA 94303  
1 (800) 786.7638  
1.512.434.1511

## Copyrights and Trademarks

Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road, California 94303, U.S.A. All rights reserved.

This documentation is distributed under licenses restricting its use. You may make copies of and redistribute it, but you may not modify or make derivative works of this documentation without prior written authorization of Sun and its licensors, if any.

Sun, Sun Microsystems, the Sun logo, Sun Webtop, StarSuite, StarOffice, the StarOffice logo, Java, JavaBeans, JavaScript, and the Java Coffee Cup are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX ® is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.



Please  
Recycle

# Contents

---

<b>Contents .....</b>	<b>3</b>
<b>Overview .....</b>	<b>4</b>
<b>Architecture .....</b>	<b>5</b>
<b>Enabling Unicode Text.....</b>	<b>6</b>
Text representation .....	6
Multi-lingual Display .....	6
Multi-lingual Input.....	7
Multi-lingual Printing.....	7
UI Message can be in any language.....	8
User Defined Font Set.....	8
<b>Multi-lingual Text Processing .....</b>	<b>9</b>
Break Iterator .....	12
Collation .....	14
Find / Replace.....	16
<b>Additional features for CJK users .....</b>	<b>17</b>
Vertical Writing.....	17
StarOffice Impress: .....	19
Ruby Text Implementation.....	20
Font Effects Tab Page Changes .....	21
Asian Layout Tab Page.....	22
Numbering/Bullets Enhancements for CJK .....	23
<b>Conclusion .....</b>	<b>24</b>
<b>Reference .....</b>	<b>25</b>
<b>Contacts.....</b>	<b>25</b>
<b>Acknowledgements .....</b>	<b>26</b>

# StarOffice 6.0

## A Multi-lingual Office Suite

### Abstract

*Unicode based character representation enables multi-lingual text. The user cannot benefit from this representation unless the application supports multi-lingual text input, display and printing. Typically, the language input and output support available on the platform is tied to a locale, limiting support to only one or a very few languages. StarOffice6.0 uses Unicode to represent multi-lingual text. While StarOffice leverage on platform specific input/output APIs, it has overcome multi-lingual limitations on some platforms (like Linux) by enhancing input method support and creative font handling techniques. Using this capability, users can input/output multi-lingual text across all platforms in a consistent manner. StarOffice i18n framework has added new features to allow the user to apply any language setting to different sub-sections of document enabling multi-lingual collation or word break behavior in a single document. StarOffice provides rich locale specific features, such as vertical writing and ruby text support, which are available with any contemporary office suite like Microsoft Office 2000. With cross- platform multi-lingual support and new features for Asian language users, StarOffice allows the user to author, modify and view documents on any system in any locale.*

## Overview

---

StarOffice® (also known as StarSuite® in Asian countries) is a complete, extensible, cross-platform personal productivity application suite. It comes along with several productivity applications such as:

- StarOffice Writer® for document authoring
- StarOffice Calc ® for spreadsheets
- StarOffice Impress® for presentations
- StarOffice Draw® for image editing.

Sun Microsystems has provided most of the source code of this product through open source licensing. The open source project is called OpenOffice.org.

StarOffice 6.0 is currently supported on Solaris Sparc, Solaris Intel, Microsoft Windows 9x/NT/ME/2000 and Linux. It is completely implemented in C++ language. It provides consistent user interface and functionality across all platforms. The current version uses Unicode in order to support all European, Asian and BiDi languages. Adopting Unicode as basic character representation allows StarOffice to represent multi-lingual text without any additional overhead. Any application that supports Unicode must also support multi-lingual input, display and printing in order to utilize all the benefits of Unicode. Unfortunately, most of these operating systems support only single locale and do not have the capability of implementing multi-lingual text.

This presentation describes StarOffice architecture and the changes made to overcome limitations posed by each platform, thus enabling multi-lingual input, display and printing across all supported platforms. By Leveraging on StarOffice i18n framework, user can perform any locale sensitive operation such as collation in any locale without depending upon platform locale. The detailed design principles behind this i18n framework are discussed in the next sections. Several new features and enhancements such as vertical writing, that are targeted for Asian language users are also described, which enables StarOffice globally across all platforms and locales.

## Architecture

---

StarOffice follows a layered module approach to achieve consistent functionality across all platforms. The Layered module approach defines the modules into distinct components, which separates them strictly through the interface. This approach makes it easy to enhance the module without affecting the rest of the product and thus improves the productivity. StarOffice modules are classified into four distinct layers or tiers as follows:

### **System Abstraction Layer**

This layer encapsulates all system specific APIs and provides a consistent object-oriented API to access all system resources in a platform-independent manner. The modules outside this layer do not use any platform dependent APIs. Porting StarOffice to other platforms involves porting or rewriting the modules in this layer only. Modules in this layer handle platform independent representation of multi-lingual text by using Unicode, multi-lingual input, display and printing.

### **Infrastructure Layer**

A Platform-independent environment for building applications, components and services is provided by the infrastructure layer. It covers most aspects of an object-

oriented API for a complete object-oriented platform including a component model, scripting, compound documents, etc. StarOffice components follow the UNO component model, which is similar to COM but with additional features. UNO is an interface based component model that supports exceptions. Currently UNO binding for C++ and Java are available. All StarOffice components are implemented as UNO components in C++. Since the interaction between UNO components is only through interfaces, implementation of each component can be done independently.

### **Framework Layer**

To allow the reuse of implementations in different applications, the framework layer provides the framework or environment for each application and all shared functionality like common dialogs, file access or configuration management. The StarOffice I18n framework falls into this layer, which is used by all StarOffice components for easy implementation of i18n APIs.

### **Application Layer**

All StarOffice applications such as StarOffice Writer are part of the Application layer. The way these applications interact is governed by Framework and Infrastructure layers.

## **Enabling Unicode Text**

---

Enabling Unicode text in an application requires an abstract mechanism to represent multi-lingual text, input, display and printing. Text IO capability among the platforms is different. Hence this capability is implemented in System Abstract Layer to provide consistent and platform independent functionality to rest of StarOffice components.

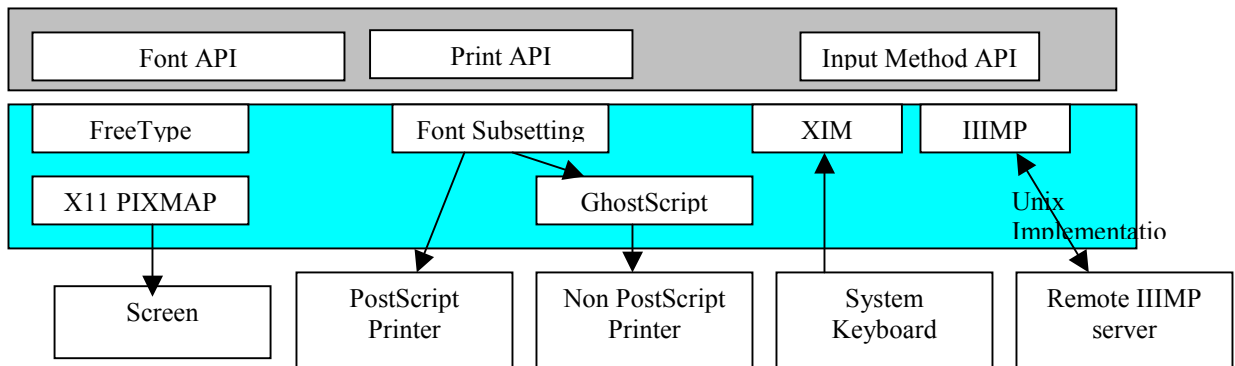
### **Text representation**

The System Abstraction Layer (SAL) abstracts the system level functions required to run StarSuite. The Runtime Library (RTL) module is part of SAL and implements Unicode based string and memory management routines and codeset converters. The RTL module implements UnicodeString as an array of 16 bit integers. At this level the string class is represented as an array of codepoints. The StarOffice components use Character Iterator APIs to find the character boundary in the code point array without assuming one codepoint is one character. The Unicode String implementation provides the application with codepoint and character view for better performance.

### **Multi-lingual Display**

The Visual Class Library (VCL) module encapsulates all access to the different underlying GUI systems. This module provides abstract interface to render multi-lingual text on the screen. StarOffice components require rich set of font APIs for

text display such as anti-aliasing, vertical glyph, rotation, and kern pair information. Win32 graphic system APIs is sufficient to meet these APIs on Windows. But on the Unix platform, X11 APIs do not allow operating on any of the low-level font operations. To overcome this limitation, StarOffice has integrated a FreeType rasterizer to perform the font level operation. Instead of using X11 Rendering functions such as XDrawString, StarOffice generates a pixmap using the FreeType rasterizer and then uses the X11 Pixmap APIs to display them on the screen. The figure below shows the architecture of VCL. Using this built-in rasterizer, StarOffice can make full use of local fonts installed on the system without relying on X related font paths.



## Multi-lingual Input

The VCL module provides abstract APIs to for multi-lingual input. The abstract input method APIs support *on-the-spot (for Asian languages)* editing at the component. VCL uses platform IME APIs such as XIM on X11, which provide on the spot support. If the platform supports multi-lingual input (for example MS Windows2000), StarOffice makes full use of this capability. Usually input method is tied to a platform locale and limits the number of languages the user can enter. StarOffice bundles with a platform and codeset neutral input method protocol to get input method service from a remote server. The input method protocol is called IIIMP, which is available as open source. StarOffice can use IIIMP to interact with any multi-lingual input method server and user can input multi-lingual data even though the platform does not support it.

## Multi-lingual Printing

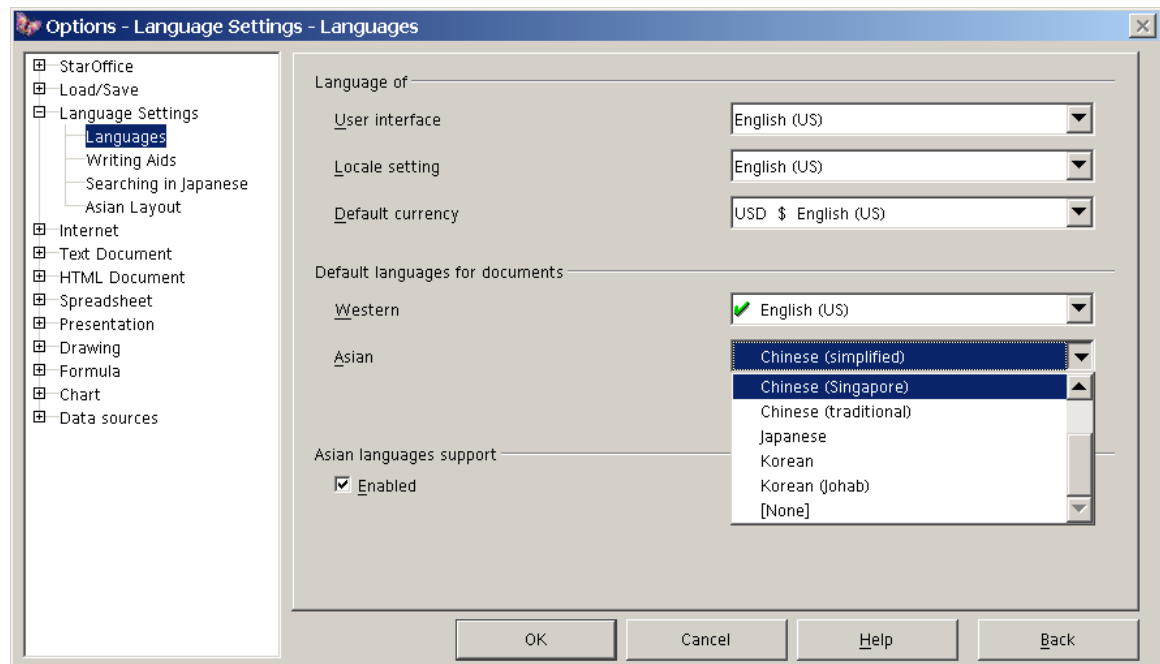
The VCL module uses a platform's printing capability to print documents. VCL uses Win32 API on the Windows platform and generates PostScript output on Solaris and Linux platforms.. The Win32 print engine can print on all types of printers ranging from dot matrix, PCL3 up to PostScript printers. The generated PostScript can be sent directly to a PostScript printer directly. Ghostscript must be used to print the PostScript on a non-PostScript printer. StarOffice does not use Xprint to generate PostScript since Xprint engine assumes that the target printer will have PostScript engine and all necessary fonts to render the page. Asian

language documents often cannot be printed correctly because all the fonts used by StarOffice may not be present on the PostScript printer. To overcome this restriction, VCL performs a font-subsetting method to embed the glyphs used in the document as part of a PostScript document. The PostScript printer does not require the font to be installed if the glyphs are embedded in the document. Font subsetting allows VCL to ensure that the document is printed as seen on the screen (WYSIWIG) without any font support from the printer.

The VCL module has successfully bridged the gap between the graphics APIs of Win32 and Unix flavor to provide consistent APIs, which handle multi-lingual text on all platforms. StarOffice has several new features leveraging on the successful operation of these modules.

### UI Messages can be in any language

StarOffice is bundled with one TrueType font, which has most character glyphs defined in Unicode 2.0. Since StarOffice multi-lingual functionality is not limited by platforms any underlying locale, users can have UI messages in any language irrespective of the platform locale. For example, the user can choose to have Japanese UI message on English locale. To configure language for UI messages, StarOffice has added a new GUI dialog, which is described below.



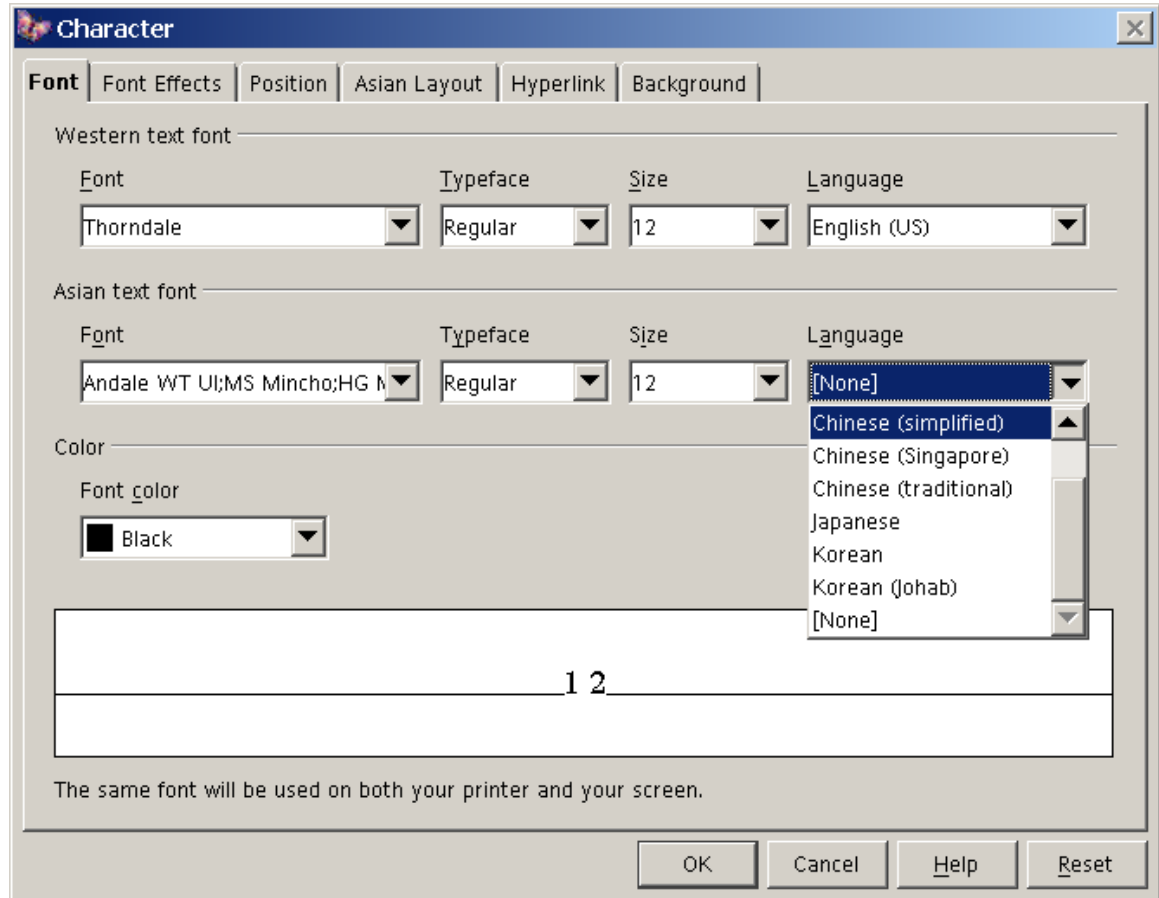
Using this GUI dialog, the user can configure the preferred locale, user interface language and the Asian and Western default languages for documents.

### User Defined Font Set

In order to display multi-lingual text, a user needs to have a font that has glyphs for more than one language. It is difficult to find a single TrueType font that has glyphs



for different languages and does not limit the user's ability to display multi-lingual text. To overcome this problem, StarOffice allows the user to define their own font set per character attribute level. The GUI dialog shows how to create personal user font sets.



A StarOffice user can choose one font for latin-1 based languages and another for Asian languages. In addition to the font selection, a user can also select the language it belongs to. For example, the Asian language font belongs to Chinese (Simplified). StarOffice Writer and StarOffice Calc components will use word breaks, spelling checkers of the selected language for all the text in the document with this character attribute.

## Multi-lingual Text Processing

The user should be able to perform locale sensitive operations such as collation on multi-lingual text in the document. StarOffice does not use platform i18n API to perform such locale sensitive operation since the i18n APIs are not consistent across the platform. Instead all StarOffice components use the platform independent i18n APIs provided by the i18n Framework module. This module falls into the framework layer.

The i18n framework does not perform any locale sensitive operation instead it loads locale sensitive modules as UNO objects. Since UNO components can be replaced, added or enhanced without recompiling the rest of the application, new locale support can be added without recompiling the rest of the StarOffice components. This module follows standard UNO service naming convention to locate and load locale sensitive modules. It also implements a fallback approach to load locale sensitive modules as shown in the following table:

Order	Locale
1	<Language>_<country>. <Variant>
2	<Language>_<country>
3	<Language>
4	Default component – the name of the component is directly coded in the stub

Along with the i18n framework, StarOffice provides a Unicode based implementation for collation, character classification and word iteration. While this implementation can be applicable to many locales, it is not sufficient for all locales. This implementation is used as fallback implementation when no locale sensitive implementations can be found. The following figure shows the i18n framework architecture. More details can be found in [1].

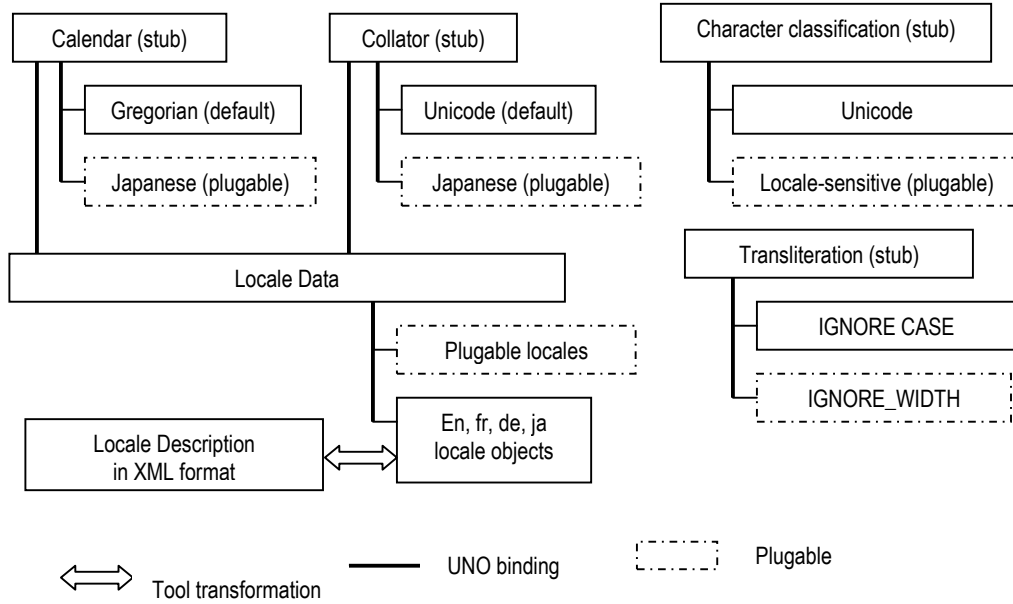


Fig 2: StarSuite I18N Framework Architecture

The key design goal behind this framework is pluggability to decouple the locale support from StarOffice components. Detailed description of each module is as follows

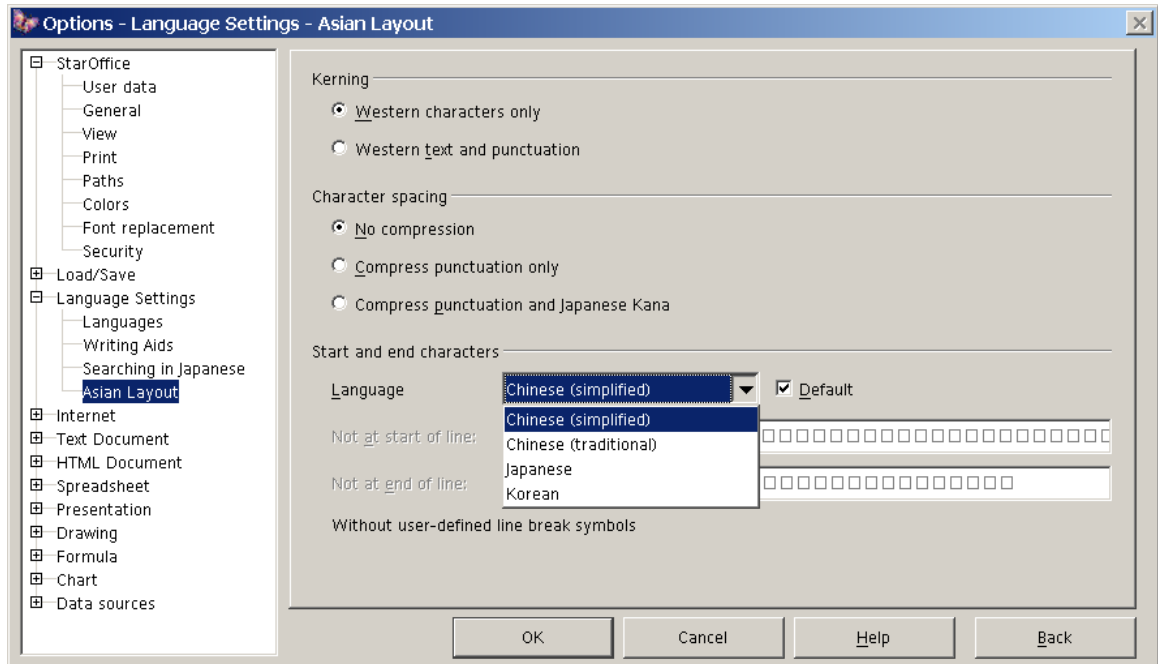
---

## Break Iterator

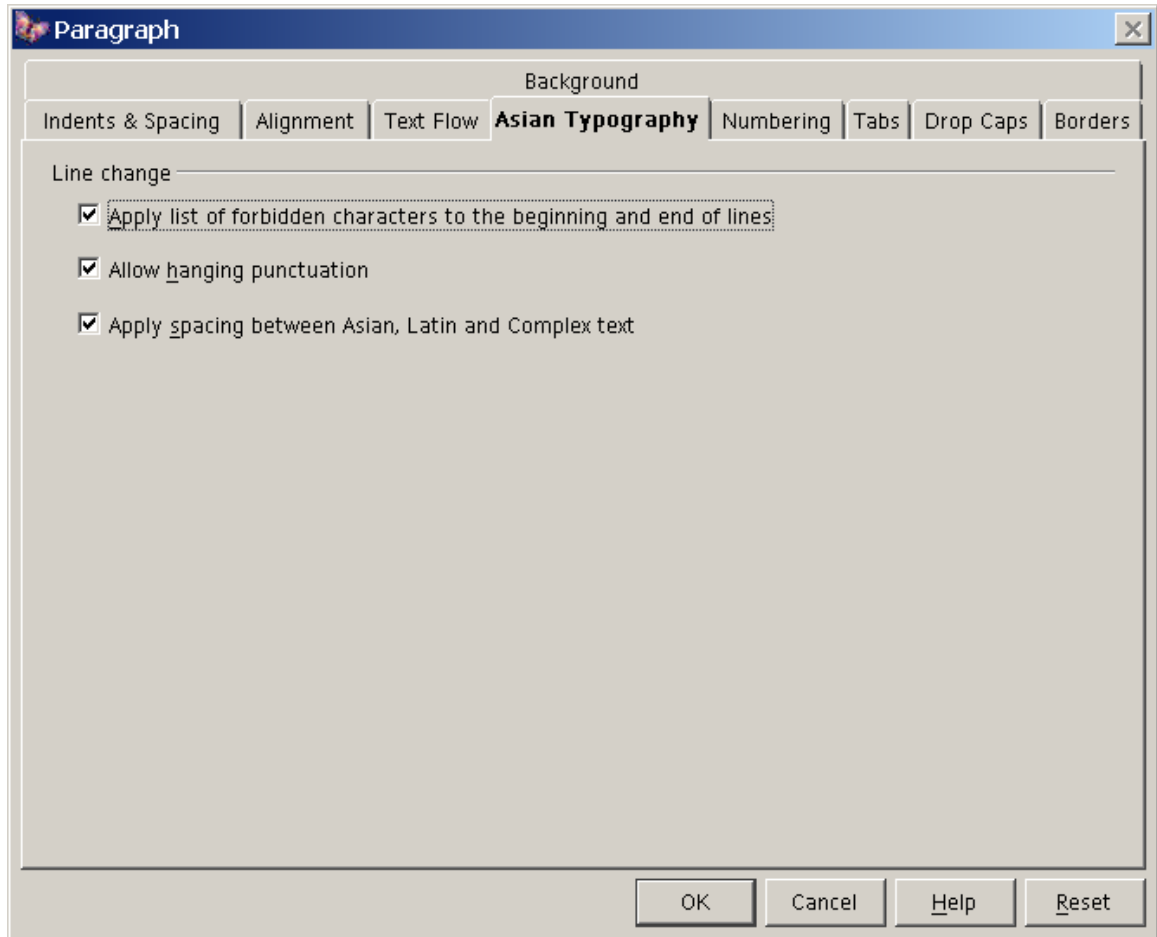
StarSuite does not make the assumption that one character is one code point, which disables the edit control components ability to increment the index by one to calculate the next cursor position. Instead, the StarSuite I18n Framework provides character and word iteration APIs to calculate the next/previous cursor position. Word break algorithms for locales such as Japanese require a dictionary lookup to identify a meaningful word. Line break for edit components of StarSuite can be configured with any one or more of the following options:

- Hyphenate.
- Forbid some set of characters to begin or end the line.
- Allow some punctuation marks to render even outside the margin.

Since StarSuite uses Unicode, it has capabilities to handle multi-lingual documents. If a Japanese document is viewed with StarSuite Writer in an English locale, it does not make any sense to apply English dictionary rules to the Japanese document. In order to associate correct dictionary rules, this module also provides APIs to auto-detect the script of the string. StarSuite currently has a break iterator that works on Unicode as a default object loaded by stub. It also comes with a dictionary based break iterator for Japanese and Chinese locales also.



The figure above shows the new GUI for Asian language users to configure Asian layout. This can be invoked from Tools-> Options -> Language Settings. This dialog lists the forbidden characters to be applied. The default value for this field is retrieved from locale data in XML format. The following dialog shows how to enable page layout rules specific to Asian typography.




---

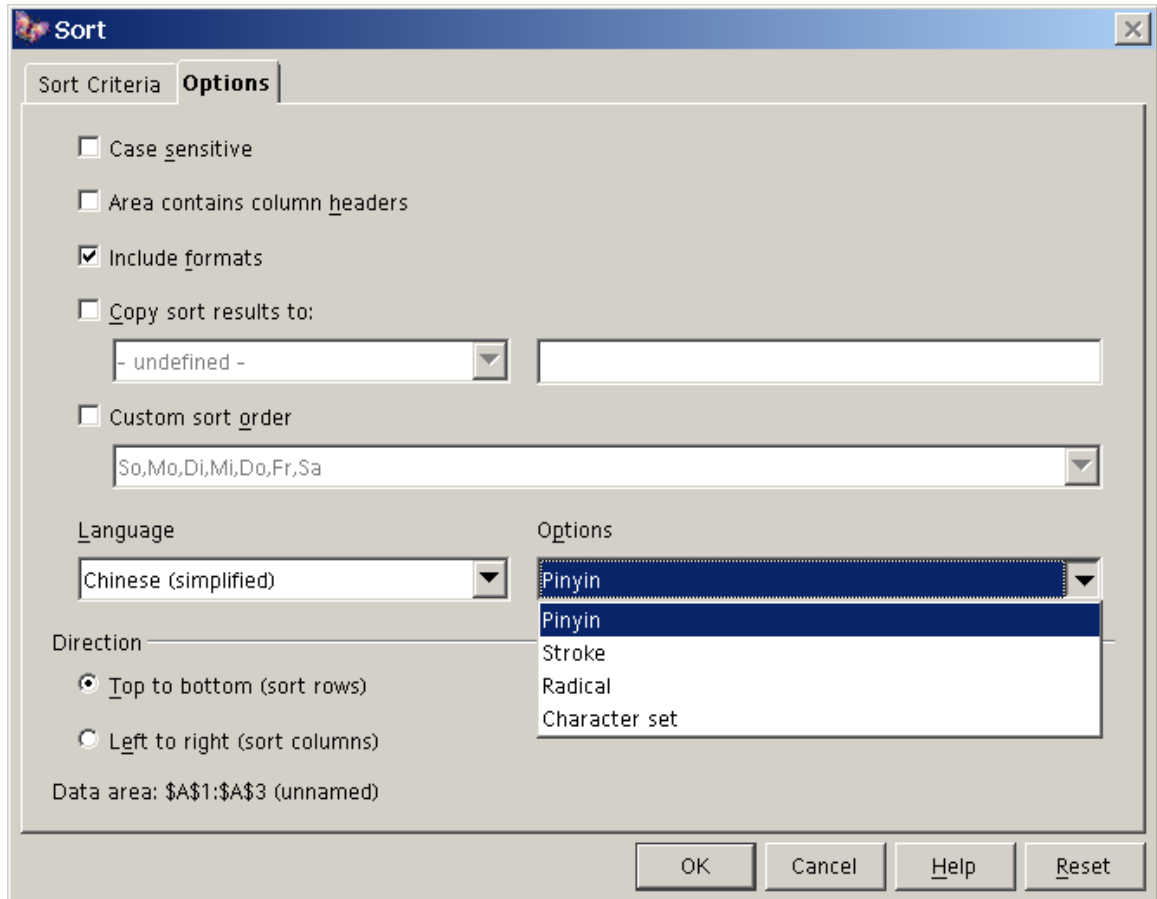
## Collation

Collation is used by many modules in StarSuite. Since collation algorithms are locale-sensitive, each collation algorithm is registered under unique UNO service name. Because of its widespread use by various components, the collation stub needs to provide more sophisticated functionality than just loading localized modules. Collation usage can be classified in two broad categories:

### User-Invoked

In this category, the user selects data from a spreadsheet and invokes sorting through the GUI. The sorting dialog allows users to select the sorting algorithm and the sort options like *case insensitive*. The sorting algorithms available as options to end-user are locale-sensitive, i.e. the German telephone number sorting algorithm is not applicable to the Chinese user. The UNO service name of all collation algorithms applicable to a given locale is listed in XML locale data. The collation stub provides a separate API (*listCollatorAlgorithms(locale)*), which retrieves the list of collator algorithms applicable to a locale from locale data. For example,

Chinese locale can have more than one valid collation algorithm. StarOffice queries using UNO API and lists to user as shown in the following dialog.



In the collation dialog, there is an additional option in the Language Item called *Default*. When the Default option is selected, StarOffice uses Platform's locale to sort the data. This can be useful option to user. For example, a French user collates the data in a table using the default locale option, the data is sorted as per French locale, if Japanese user views the same document, the data is sorted as per Japanese locale. When the user chooses to have a specific behavior in any locale that StarOffice supports the user can override the default with a specific language.

### Application-Invoked

StarSuite invokes collation modules for sorting font names, file names, auto completion, auto correction and so on. Sorting the different data items need not be done according to a general rule. For example, the font names can be sorted insensitive to case in the English-US locale but it can be different for a Japanese locale. These collation options can be mapped into a transliteration module. Since collation options for sorting is locale-sensitive, they are listed in the locale data under an abstract option. StarSuite modules pass the abstract option to the collator stub and the collator stub looks in locale data to find out the actual transliteration modules defined for the abstract name and applies them before invoking the actual

collator algorithm. For example, StarSuite defines an abstract option called NAME\_SORT which is used for sorting font names and file names. The abstract option is mapped into CASE\_IGNORE for the English locale and IGNORE\_WIDTH for the Japanese locale.

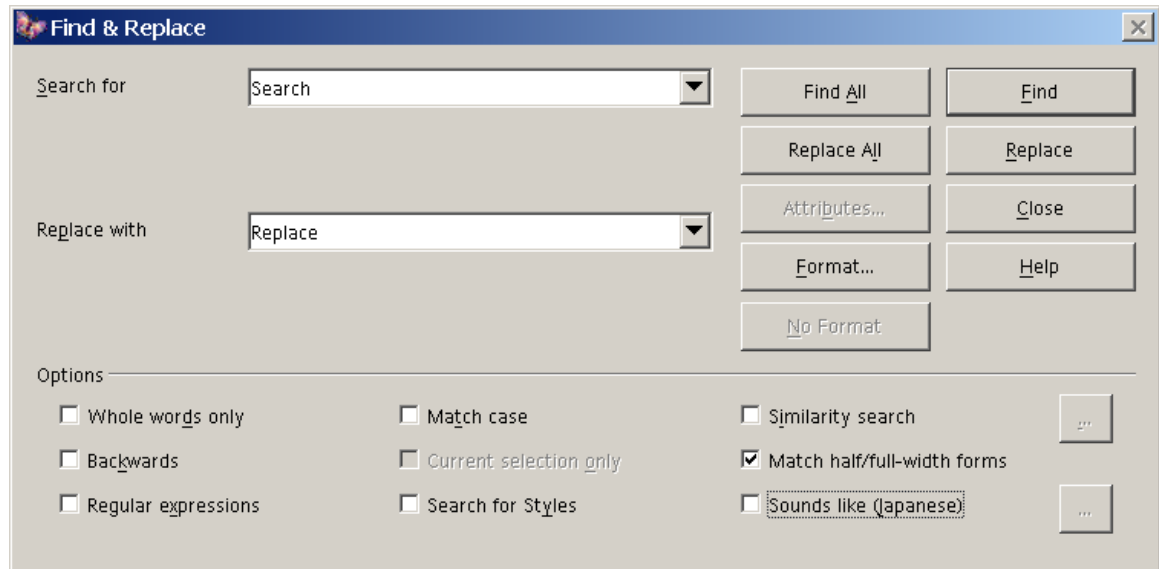
---

## Find / Replace

StarSuite currently supports three types of search algorithm,

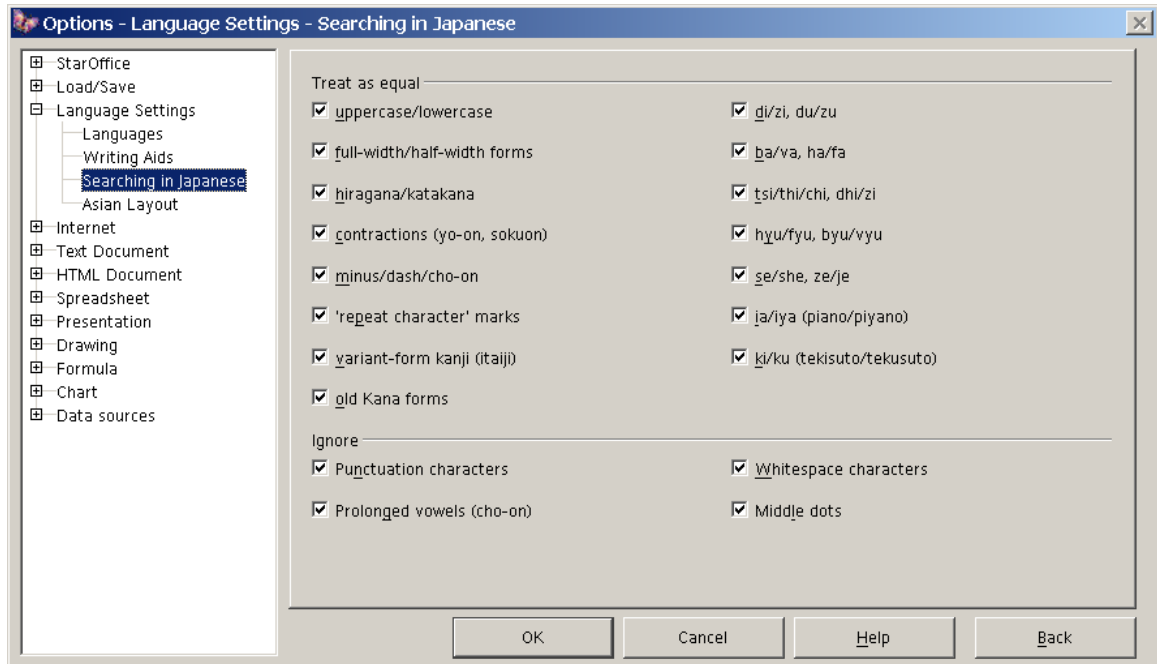
- Absolute search
- Regular expression search
- Approximate search.

The absolute search performs exact matches in a document. The absolute search provides options which can be mapped into transliteration modules. The list of search options applicable for a locale is listed in XML locale data. Using transliteration modules, Find/Replace can work without differentiate Half/Full Width format as shown below.



StarOffice also allows searching for words with sounds like option. The list of options available is shown as follows.





## Additional features for CJK users

While StarOffice has multi-lingual capability, it has added several new typographic features commonly used in Asian languages:

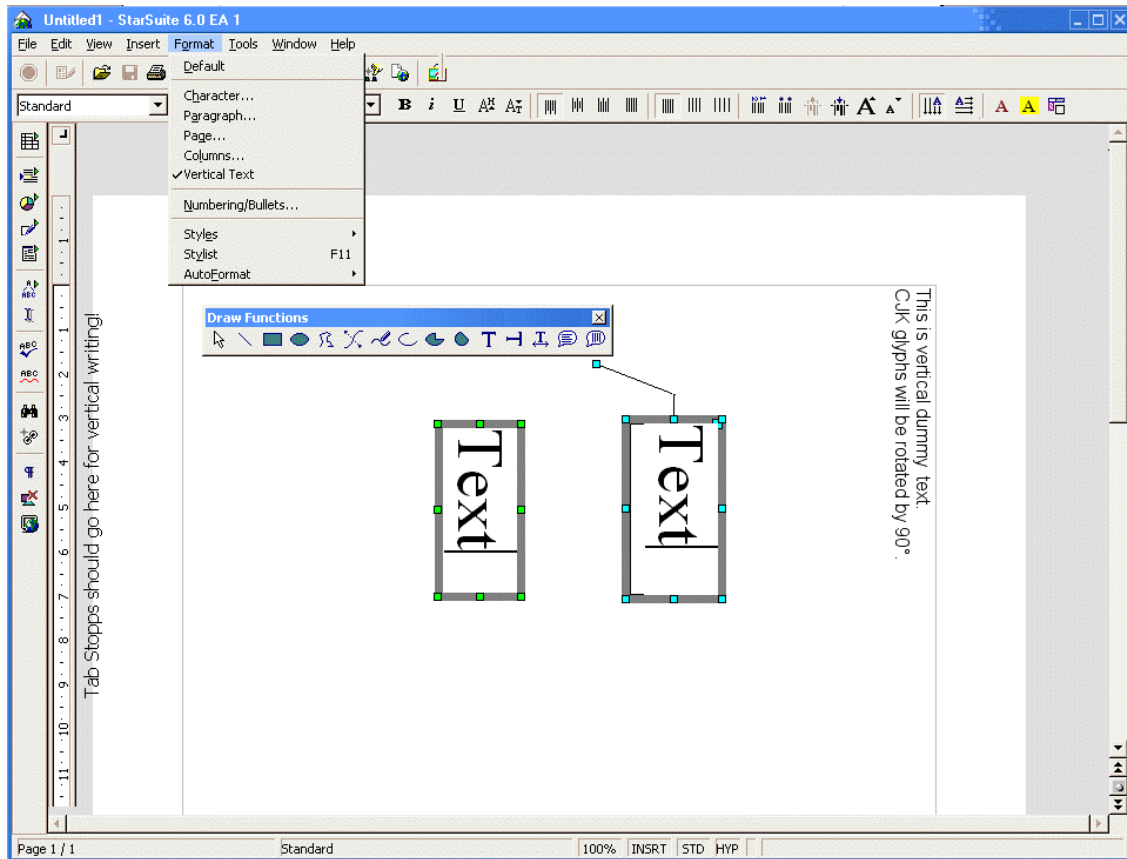
- Vertical writing
- Ruby text.

### Vertical Writing

Vertical writing is the most common way of writing in newspapers, advertisements in Asian countries. StarOffice has enhanced VCL module to handle fonts with vertical glyphs. The following screen shots show how to enable vertical writing mode and all possible text tools active with their formatted counterparts on the workspace. StarOffice Writer and StarOffice Impress have made significant changes to provide easy to use features such as

- Cursor movement in vertical mode
- Copy, paste, selection in vertical mode as in horizontal mode
- Rotating English characters in vertical mode.

StarOffice Writer gives four different ways of inserting text into a Writer document:



#### Normal Text Insertion

For normal text insertion an added set of two toggling buttons for vertical or horizontal writing was needed. Depending on the setting all icons (buttons) should change their looks accordingly.

#### Inserting a Text Frame

If the user inserts a text frame one can always rely on the new text setting possibilities mentioned in “Normal Text Insertion” but will also expect to initially set up a vertical text frame in one go.

#### Inserting Animated Text

It is not necessary to add the vertical writing functionality to the animated text frame.

#### Inserting a Call Out

Basically the same as inserting a Text Frame.

### Vertical Ruler with Tab Stops

When switching in vertical writing mode the horizontal ruler also become a vertical one. This ruler must also contain all set tab stops.

### StarOffice Impress:

StarOffice Impress gives three different ways of inserting text into a Writer document:

#### Inserting a Text Frame

If a text frame is inserted, the new text setting possibilities mentioned in “Normal Text Insertion” will apply, and initially a vertical text frame will be setup at the same time.

#### Inserting a Fit Text to Frame

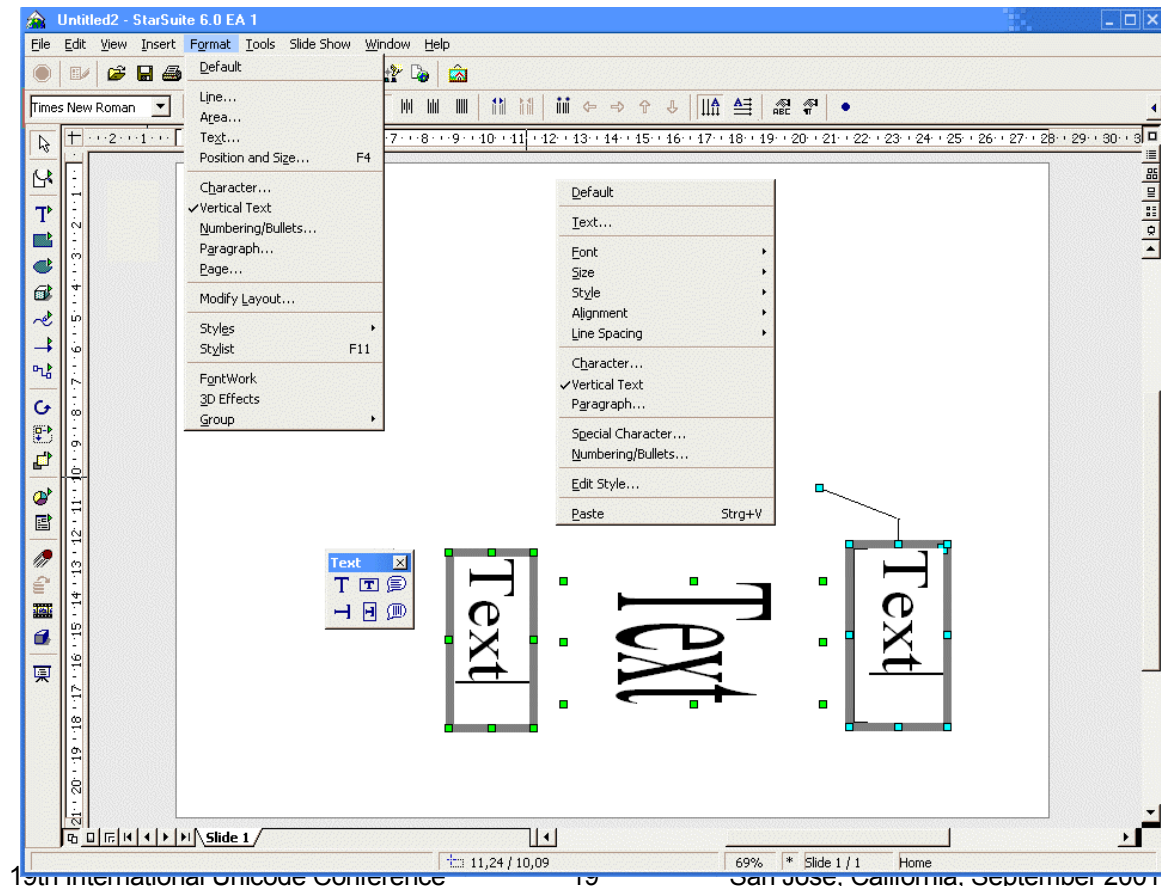
Basically the same as inserting a Text Frame.

#### Inserting a Call Out

Basically the same as inserting a Text Frame.

#### Normal Text Insertion

For normal text insertion an added set of two toggling buttons for vertical or horizontal writing was needed. Depending on the setting all icons (buttons) should change their looks accordingly.



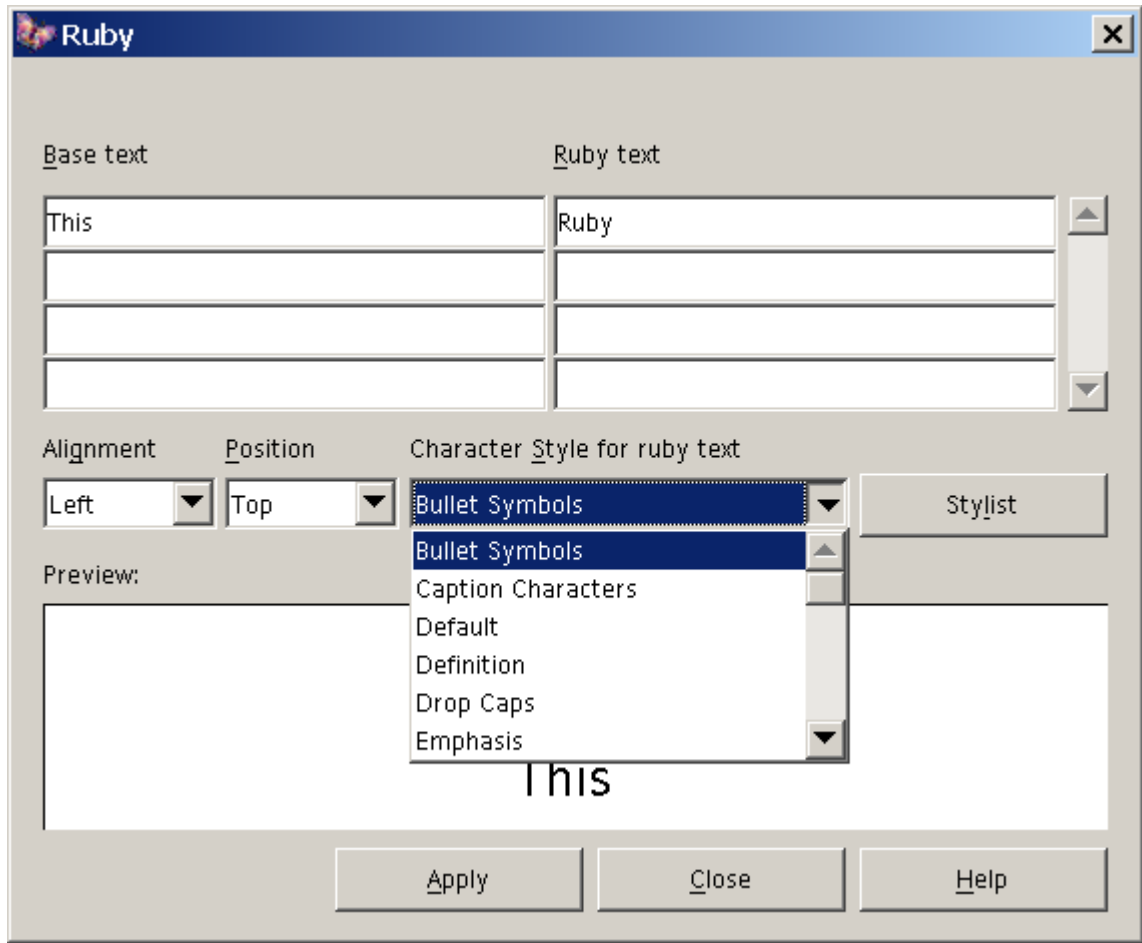
## Ruby Text Implementation

Ruby text is pronunciation aid for specific to Japanese language. StarOffice allows user to compose the Ruby text along with Japanese text composition. The Ruby dialogue is made as modeless dialogue and it floats atop of the document allowing the user to actively work on other features without closing the ruby dialog

A list of base text and ruby text will be updated as soon as the user selects new (via CTRL+mouse key combination) text. The updates are made once the user changes focus back to the Ruby-Dialogue from the text.

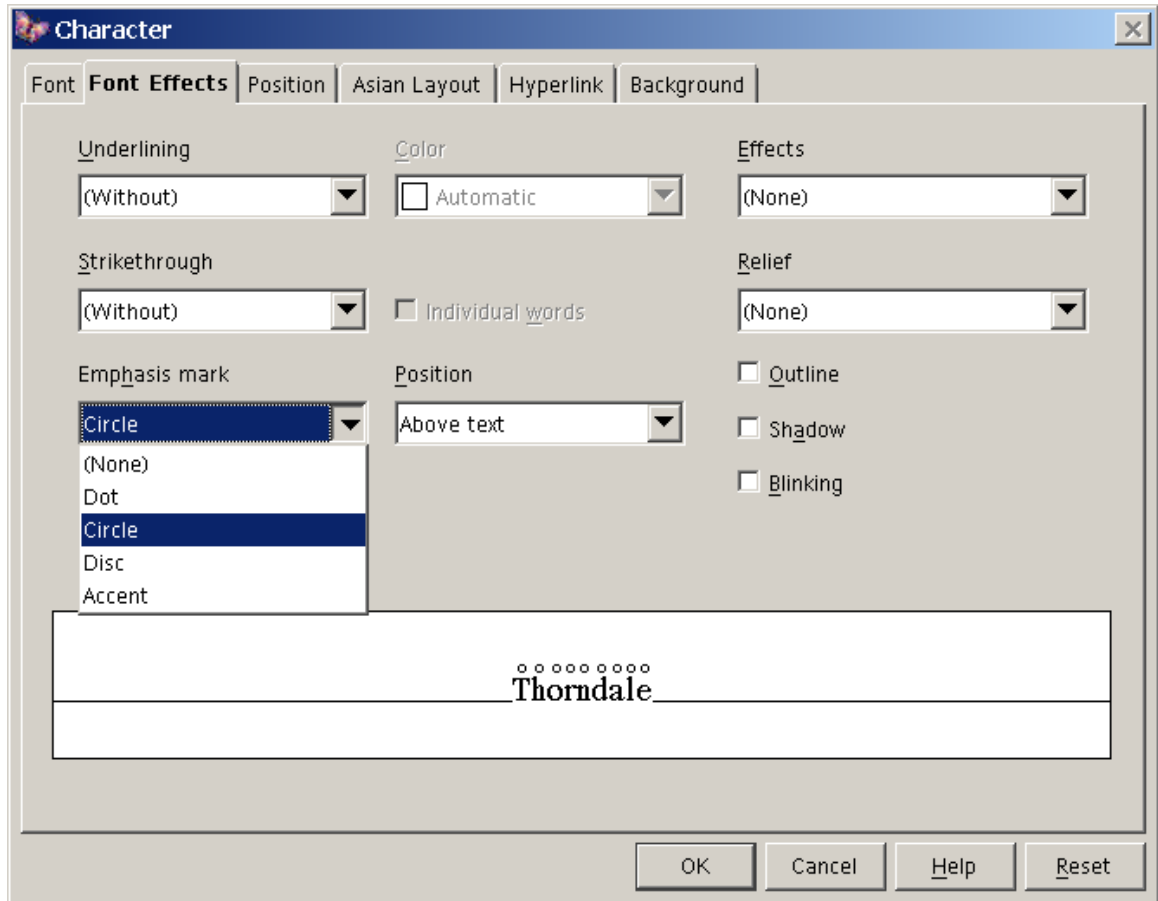
The character style of ruby text allows the user to choose a template for formatting the ruby font and its layout.

Clicking a button opens the Stylist and gives the user the direct access for creating new templates for use within the ruby text.



### Font Effects Tab Page Changes

A new emphasis mark feature for CJK requirements is provided for the CJK version. Due to massive design changes on the first tab page (“Font”) a re-arrangement has been done leaving the “Font Effects” on their own tab page. Also the new feature of having a different color assigned to lining than to the belonging fonts was introduced into the new UI.

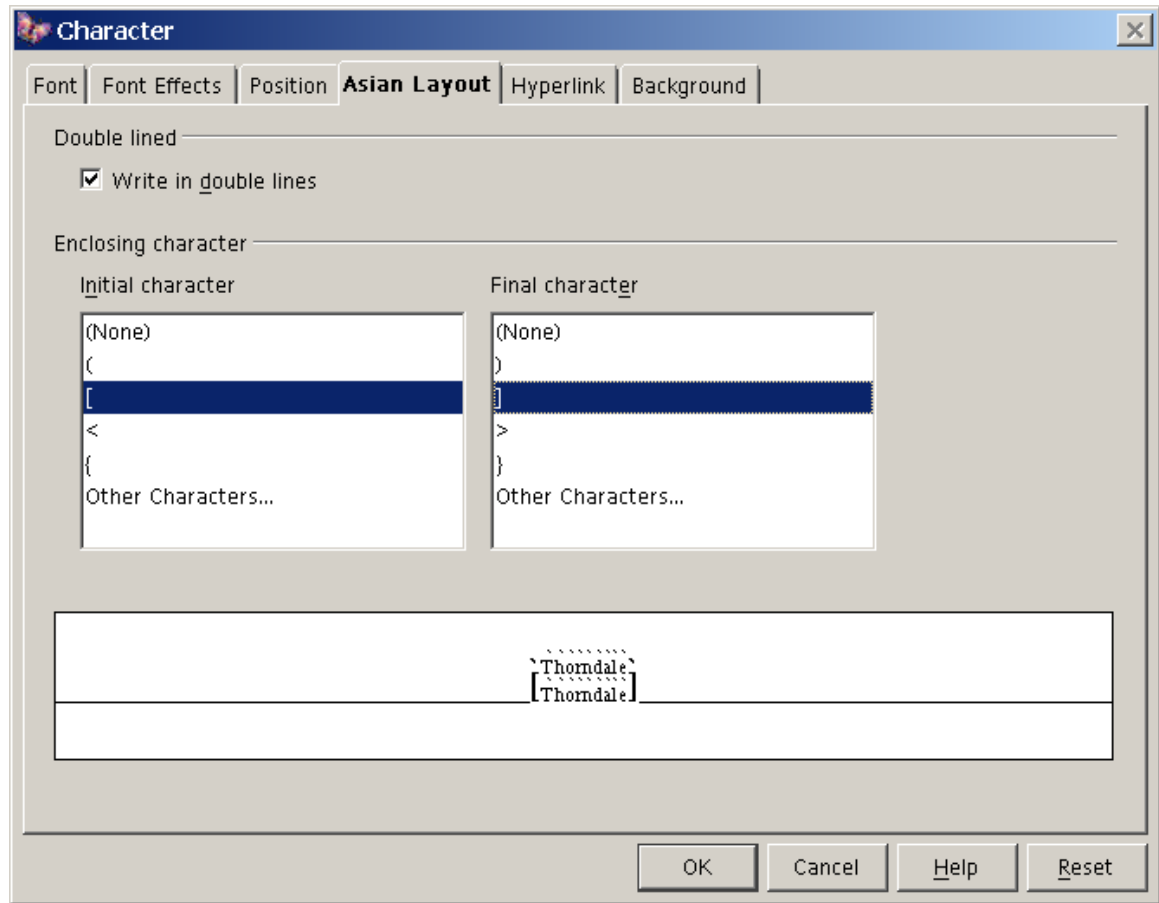


### Asian Layout Tab Page

CJK countries need special formatting to create double lined Latin text in one line. This format encloses the text section with brackets and other characters. The all-new Asian Layout dialogue provides a format for the user to write double lined text

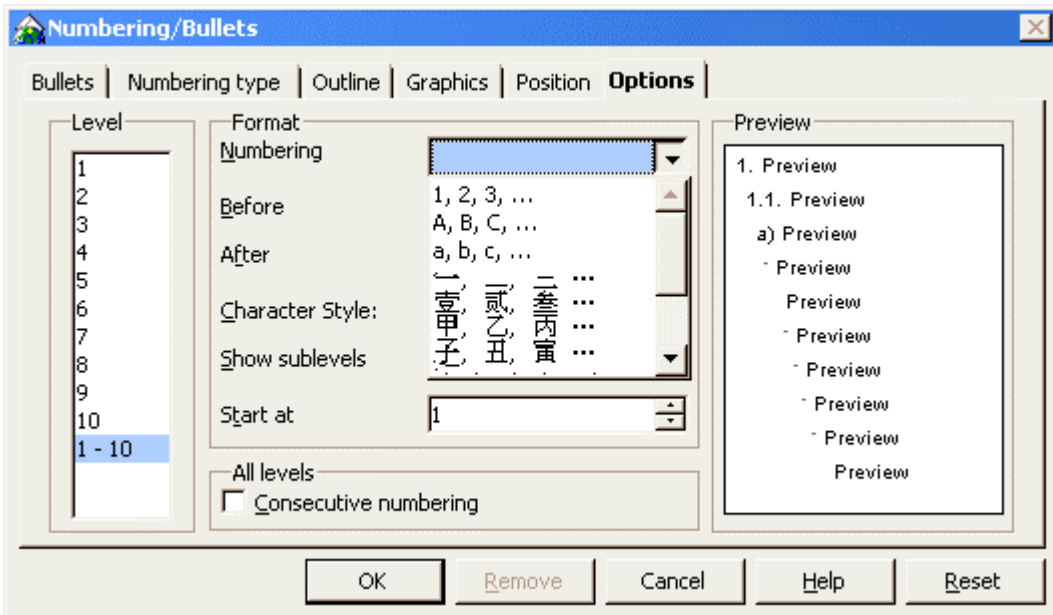
within

one.



## Numbering/Bullets Enhancements for CJK

Minor changes have been made to Numbering and Bullets dialogues. Primarily, the localization of the number formats and the creation of their equivalent preview bitmaps necessary. All available bullets and numberings are loaded from locale XML data files.



## Conclusion

StarOffice has provided Unicode support across variants of platforms by overcoming the limitations posed by the underlying operating system. It has



leveraged technologies like FreeType rasterizer, IIIMP protocol and solved key problem of multi-lingual input, display and printing across various platforms. Using this infrastructure, it has provided many nice features to manage a multi-lingual document. It has also added new features such as vertical writing for Asian users too. This new version of StarOffice will allow both document author and the audience to view multi-lingual documents irrespective of locale and platform.

## Reference

---

- [1] 18th International Unicode Conference. Shanmugam Senthil, Dieter Loeschky [Universal I18N Framework for Office Applications](#)
- [2] Sun Microsystems announces availability of StarOffice[tm] source code on OpenOffice.org  
<http://www.sun.com/smi/Press/sunflash/2000-10/sunflash.20001016.4.html>  
<http://www.sun.com/staroffice/openoffice/>
- [3] OpenOffice.org home page  
<http://www.openoffice.org>
- [4] OpenOffice.org localization and internationalization project  
<http://l10n.openoffice.org/>
- [5] UNO home page  
<http://udk.openoffice.org>
- [6] Introduction to UNO  
<http://udk.openoffice.org/common/man/concept/unointro.html>
- [7] I18n API  
<http://api.openoffice.org/source/browse/api/offapi/com/sun/star/i18n/>
- [8] OpenOffice.org Technical Overview  
[http://www.openoffice.org/white\\_papers/tech\\_overview/tech\\_overview.html](http://www.openoffice.org/white_papers/tech_overview/tech_overview.html)

## Contacts

---

Dieter Loeschky  
[Dieter.Loeschky@sun.com](mailto:Dieter.Loeschky@sun.com)

Shanmugam Senthil  
[Shanmugam.Senthil@sun.com](mailto:Shanmugam.Senthil@sun.com)

# Acknowledgements

---

The i18n framework architecture, design and APIs are teamwork done by a number of developers. The authors of this presentation would like to thank Thomas Hosemann, and Falko Tesch for their invaluable help.