



SIEBEL⁷
eBusiness

GLOBAL DEPLOYMENT GUIDE

VERSION 7.5, REV. A

12-BD362V

JANUARY 2003

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2003 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Siebel Systems, Inc.

The full text search capabilities of Siebel eBusiness Applications include technology used under license from Hummingbird Ltd. and are the copyright of Hummingbird Ltd. and/or its licensors.

Siebel, the Siebel logo, TrickleSync, TSQ, Universal Agent, and other Siebel product names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Supportsoft™ is a registered trademark of Supportsoft, Inc. Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are “commercial computer software” as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Introduction

How This Guide Is Organized	8
Additional Resources	9
Revision History	11

Chapter 1. Overview of Global Deployment

What Is Global Deployment?	14
Global Deployment Terminology	15
Formatting	18
Internationalization	18
Locale	20
Localization	21
Multicurrency Support	23
Platform	23
Siebel Language	24

Chapter 2. Supported Character Sets and Collation

Supported Traditional and Unicode Character Sets	26
Traditional Character Sets	26
Unicode Character Set	27
Siebel Character Set Encoding Names and Business Services	29
Special Character Set Encoding Names and Business Services	30
Updating Currency Symbols	30
Collation of the Local and Sample Database	31
Collation Sequences and Sorting Orders	31

Chapter 3. Planning Global Deployments

Planning Your Global Deployment	34
Addressing Data Sharing and Privacy Issues in Europe and APAC	36
Deployment Scenarios for Installing Multiple Languages on Siebel Servers	37
Scenario 1	37
Scenario 2	37
Scenario 3	38
Integration Considerations	39
Transcode Business Service	39
Character Conversion Errors in External Systems	40
Setting Up and Administering Locales	42
About the Active Language	43
Configuring Language Support for Browser Platforms	44
Configuring Language Support for Windows 2000	45
Configuring Language Support for Windows XP	47
Configuring Language Support on Windows ME/98/95 and NT 4.0	49
Configuring Language Support on IBM OS/2 and Sun Solaris 8	49
Online Help	50
Application-Wide Data	51

Chapter 4. Configuring Global Applications

Enabling Global Data	54
Translating a Globally-Enabled Field	56
Using the Global Time Zone Feature	57
Converting Data to UTC	58
Enabling UTC	58
Administering Time Zones	61
Enabling the Global Time Zone Feature in an Upgraded Environment	63
Performing a UTC Delta Upgrade	77

Configuring Bidirectional Capability	79
Installing Right-to-Left Images	79
Setting the Direction to Right-to-Left Mode on the Siebel Client	80
The Localization Process	81
What You Can Localize	82
Localizing an Unshipped Language	82
Localizing a Siebel Handheld Client	86
Supported Microsoft Windows Code Pages	86
Unicode Support for Siebel Handheld Clients	86
Adding Handwriting Recognition Software to a Supported Handheld Device	87

Chapter 5. Global Deployment Scenarios

Siebel Unicode Architecture	90
Global Deployment Scenarios	93
Monolingual Data Support Deployment	94
Multilingual User Interface Deployment	96
Unicode Database Deployment	98
Non-Unicode Database Deployment	99

Index

Introduction

This guide provides an overview of deploying Siebel eBusiness Applications in global environments. It includes overview information about global deployments, as well as a discussion of the Unicode software architecture and global deployment scenarios.

This book will be useful primarily to people whose title or job description matches one of the following:

Database Administrators	Persons who administer the database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.
Siebel Application Administrators	Persons responsible for planning, setting up, and maintaining Siebel applications.
Siebel Application Developers	Persons who plan, implement, and configure Siebel applications, possibly adding new functionality.
Siebel System Administrators	Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.
Translators	Persons involved in the localizing of applications.

How This Guide Is Organized

This guide provides conceptual information necessary to understand, implement, and deploy Siebel applications in more than one locale or language. Because many aspects of a global deployment are interrelated, you should read the entire book to fully understand the impact of any decisions you make during your global deployment. You should refer to this guide for information on the following topics:

- Siebel-supported character sets and encodings
- Deployment scenarios for installing multiple languages on Siebel servers
- Integration guidelines
- Font management
- Configuring applications for global deployments
- Global time zone support
- Configuring bidirectional capability
- The localization process
- The Siebel Unicode architecture
- Global deployment scenarios

Additional Resources

For detailed information on the procedures that you would perform during a global deployment, you should refer to the appropriate guide. For example, information on configuring multilingual picklists is contained in the *Siebel Tools Reference*. [Table 1](#) lists additional documentation on global deployments.

Table 1. Additional Global Deployment Documentation

See...	For information on...
<i>Applications Administration Guide</i>	Adding multilingual language items
	Administering multilingual lists of values (MLOVs)
	Correspondence templates
	Supported currencies
<i>Online Help Development Guide</i>	Localizing Online Help
<i>Overview: Siebel eBusiness Application Integration Volume I</i>	The eAI business service and XML Converter business service
	Unicode support when integrating applications
<i>Product Administration Guide</i>	Multilingual translations for product-related data
<i>Siebel Anywhere Administration Guide</i>	Distributing executables in a global environment
<i>Siebel Communications Server Administration Guide</i>	Configuring Siebel Communications Server in a global environment
<i>Siebel Connector for Oracle Applications</i>	Implementing in Unicode and non-Unicode environments
<i>Siebel eMail Response Administration Guide</i>	eMail Response in a global environment
<i>Security Guide for Siebel eBusiness Applications</i>	Using encryption algorithms in a Unicode environment
<i>Siebel Remote and Replication Manager Administration Guide</i>	Setting locales in the Siebel Mobile Web Client
<i>Siebel Server Installation Guide</i>	Setting locales on Siebel servers

Table 1. Additional Global Deployment Documentation

See...	For information on...
<i>Siebel SmartScript Administration Guide</i>	SmartScripts in a global environment
<i>Siebel Tools Online Help</i>	Unicode support for Visual Basic and eScript
<i>Siebel Tools Reference</i>	Checking in and out locale-specific attributes
	Compiling a Siebel Repository File (.srf) for a specific language
	Configuring with multilingual lists of values (MLOVs)
	Dynamic Toggle applets
	Multilingual Picklists
	Repositories with multiple locales
	Setting the Tools language mode
	Using the Locale Management Utility
<i>Siebel Wireless Administration Guide</i>	Deploying a Wireless application in a global environment

Revision History

Global Deployment Guide, Version 7.5, Rev. A

January 2003 Bookshelf

Table 2. Changes Made in Rev. A for January 2003 Bookshelf

Topic	Revision
Siebel Character Set Encoding Names table.	This information has been moved to <i>System Requirements and Supported Platforms</i> .
“Localizing a Siebel Handheld Client” on page 86	Added this topic.
“Setting Up and Administering Locales” on page 42	Updated information for Siebel Dedicated Clients and removed information on setting database locales and other platforms.
Language-Independent Code Parameter Values table.	This information has been moved to <i>System Requirements and Supported Platforms</i> .
“Addressing Data Sharing and Privacy Issues in Europe and APAC” on page 36	Added this topic.
Change request solutions and customer comments.	Updated content where appropriate.
Recommended Font Settings.	This information has been moved to <i>Siebel Tools Reference and Configuration Guidelines</i> . In addition, an obsolete table on font settings has been removed.

Introduction

Revision History

Overview of Global Deployment

1

The global deployment of a single software application can mean different things to different people. The purpose of this chapter is to clarify the concept of global deployment by providing information on the following subjects:

- [What Is Global Deployment? on page 14](#)
- [Global Deployment Terminology on page 15](#)

What Is Global Deployment?

Siebel eBusiness Applications are designed to meet the needs of customers operating in a global environment. These needs might include:

- Functioning simultaneously in multiple regions, using multiple languages
- Viewing data in local time and number formats
- Managing global time differences (optional)
- Supporting local business models (optional)
- Meeting local regulations
- Allowing each individual to work in a local language environment

A single installation of a Siebel application can run in more than one country simultaneously, even though people in these countries may speak different languages. This is called *concurrent multiple language support*, which means that the text in the user interface and the entries in picklists can be displayed to multiple users in any of the languages defined in your Siebel application. Global deployment encompasses all of the above and generally means a deployment that goes beyond a single instance in a single language.

Global Deployment Terminology

This guide uses many specialized terms. [Table 3](#) provides a list of many of these terms.

Table 3. Global Deployment Terminology

Term	Meaning
Character Set	A group of characters (alphanumeric, text, or special) usually associated with one or more languages, such as the English alphabet or Japanese kanji. There are many character sets used in the computing industry. Character sets are identified by a <i>character set name</i> .
Character Set Encoding	Also known as <i>character encoding</i> . A specific computer representation of a character set. Some character sets can have multiple encodings—for example, Western European or Latin 1 is encoded differently in ISO 8859-1, 8859-15, ANSI 1252 (Microsoft standard), and EBCDIC. Unicode also comes in different encodings, such as UTF-8 or UTF-16. In general, the differences in encodings are between ISO, ANSI, and EBCDIC.
Character Set Encoding Name	A name identifying a <i>character set encoding</i> . For character sets with a single encoding, the <i>character set encoding name</i> and the <i>character set name</i> are usually the same. The notable exception is Unicode. Aside from Unicode, the most prominent example of a character set with multiple encodings is JIS (Japan Industrial Standard). Shift-JIS, EUC, and ISO 2022-JP are all encodings based on JIS. As with character set names, industry standardization is minimal and there are multiple names for the same encoding.
Character Set Name	A name identifying a <i>character set</i> , for example Western European or Latin 1. These names are not well-standardized and many character sets have multiple names. However, you can use formal identifiers to clearly specify a character set when necessary.
Code Page	Also known as <i>codepage</i> . A Microsoft Windows implementation of a <i>character set encoding</i> . Some other vendors use the term code page and even in Microsoft Windows, it is used for ANSI code page (Windows) and OEM code page (DOS), but not for ISO character sets. IBM uses a numbering system which is similar, and often identical, to Microsoft. However, IBM rennumbers extensions while Microsoft does not, which can lead to instructions, such as, “Use IBM 943 with Siebel applications and MS 932; IBM 932 is an older version.”

Table 3. Global Deployment Terminology

Term	Meaning
Code Page Number	A Microsoft Windows designator for a Microsoft Windows code page. This number is used by the Windows operating system to identify code pages and is similar to a <i>character set encoding name</i> .
Coordinated Universal Time (UTC)	Also known as <i>universal time coordinated</i> . A time scale defined and recommended by the International Radio Consultative Committee (CCIR), and maintained by the Bureau International des Poids et Mesures (BIPM). The Global Time Zone feature uses UTC, which does not observe daylight savings time (DST). For more information, see Chapter 4, “Configuring Global Applications.”
Formatting	The formatting of data, such as dates, time, numbers, and currency. For more information, see “Formatting” on page 18 .
Global Deployment	The process of installing, configuring, testing, and deploying Siebel eBusiness Applications in more than one locale and language.
Internationalization	<i>Internationalization</i> (often abbreviated as I18N since there are 18 characters between the initial I and the terminal N) refers to the process of making a software product that can correctly process data for any customer, including data entry and display, through proper locale. For more information, see “Internationalization” on page 18 .
Internet Assigned Numbers Authority (IANA)	An Internet body that managed Internet addresses, domain names and protocol parameters. It has been replaced by ICANN (Internet Corporation for Assigned Names and Numbers), which was formed in 1998. IANA conventions are used for the World Wide Web, email, and XML.
Locale	A <i>locale</i> refers to a set of user preference information related to the user’s language and cultural conventions. For more information, see “Locale” on page 20 .
Localization	<i>Localization</i> (often abbreviated L10N since there are 10 characters between the initial L and the terminal N) is the process of adapting a program for a specific locale. For more information, see “Localization” on page 21 .
Multicurrency Support	A feature that allows automatic currency conversion and currency formatting. For more information, see “Multicurrency Support” on page 23 .

Table 3. Global Deployment Terminology

Term	Meaning
Platform	A <i>platform</i> includes the operating system of the various entities of a Siebel deployment; the database, the Siebel servers, and the clients and the character set used by these entities. For more information, see “Platform” on page 23 .
Siebel Character Set Encoding Name	The name used within Siebel eBusiness Applications to refer to a specific character set encoding.
Siebel Language	The language of the Siebel eBusiness Applications software installed on the system. For more information, see “Siebel Language” on page 24 .
Traditional (Non-Unicode) Character Set	Traditional (Non-Unicode) character sets refer to character sets other than Unicode and imply that the character set supports a restricted set of characters. Typically, a traditional character set supports the alphabet of a single language or collection of languages that use the same or similar alphabets.
Unicode Character Set	A character set defined by the Unicode Consortium that is the union of most of the major character sets used in the computing industry.

Formatting

Siebel eBusiness Applications support formatting of data, such as dates, time, numbers, and currency, based on locale settings. The Siebel Web Client adopts the locale settings from either the regional settings of the operating system on which the Siebel server is running or the locale settings defined for a particular object manager. The Siebel Mobile Web Client and Siebel Dedicated Web Client adopt the locale settings defined in the client operating system's regional settings.

Examples of different formats based on locales include:

- Date and Time
 - 03/10/95 (U.S. format as mm/dd/yy)
 - 10.03.95 (German format as dd.mm.yy)
- Number
 - 1,234.34 (U.S. format with a comma as digit grouping symbol and a period for the decimal symbol)
 - 1 234,34 (French format with a space as the digit grouping symbol and a comma for the decimal symbol)
 - 1.234,34 (German format with a period as the digit grouping symbol and a comma for the decimal symbol)
- Phone Number
 - + 33 1-23 42 34 56 (French phone number as shown in U.S. regional settings)
 - (415) 295-5000 (U.S. phone number as shown in U.S. regional settings)

Internationalization

Internationalization includes designing software to handle and display data, such as text, diagrams, and numbers, according to the orthography or rules of the language as used in a particular locale. The software might have to input, display, and print characters, sort text, and recognize numbers and dates in different formats, and display and print text right-to-left as well as left-to right. Therefore, certain engineering features must be incorporated into the code to handle these requirements.

Developing an internationalized program means that the feature and code designs do not make assumptions based on a single language or locale and that the source code base simplifies the creation of different language editions of a program.

Some aspects of internationalization include:

- A base version enabled for international environments
- Localizable items separated from the core functionality on which they are running
- Software that takes advantage of supporting platforms, such as the Windows operating systems and the database platform the software is running on

Your Siebel application has been internationalized. Specific features include:

- A base version, enabled for international environments
- Support for localization built into the data model
- Support for separate language-specific modules (where necessary)

For example, some DLLs are language-independent, while other DLLs are language-dependent. Language-dependent DLLs are located in the language-dependent installation directory that they support.

- Euro currency support and triangulation
- String, number, and date handling
- Support for multilingual user data such as:
 - Multilingual picklists (MLOV seed data)
 - Multilingual data for product- and catalog-related entities
- Support for major Unicode and non-Unicode (Traditional) character sets.

For more information, see *System Requirements and Supported Platforms*.

Locale

Locale is a collection of user inputs, including keyboard layout, sorting order, and the formats used for numbers, dates, currencies, and times.

- **User locale.** You can set a locale to provide data to users in their native format, including the formatting of numeric information, such as numbers, times, dates, and currencies. Typically, user locales contain the symbols for the thousand separator, decimal point, negative number representation, time separator, short data format, long data format, and currency symbols. A country specification is often used to select default values for user locale settings.

NOTE: A user locale is *not* a language setting and has no relationship to input languages, keyboard layouts, character sets, or user interface languages.

- **Input locale.** This locale is used to describe the language being entered and the input method, which could be a particular keyboard layout or a speech-to-text converter.

Keyboard layout is a defined input locale that correlates the keys on the keyboard to their subsequent character definition mapping within the code page of the operating system.

- **System locale.** If you are using a Microsoft Windows operating system, this is a system-wide setting that designates which code page is used as the default for all users on the system. If you are using a UNIX operating system, the settings for formatting and code page locales are *not* systemwide. These code pages and fonts allow non-Unicode applications to run as they would on a system localized to the language of the system locale.

NOTE: If you are using a Windows operating you *must* restart the system after changing a system locale.

- **Sorting order or collation.** The process of ordering units of textual information. Collation is usually specific to a particular language. The Unicode Collation Algorithm defines a complete, unambiguous, and specified ordering for all characters in the Unicode Standard.

Locale Usage

You can use locale rules to vary the appearance of data for different regions of your implementation. Typically, this data would include dates and times, numbers, and currencies.

For example, the date and time *thirty minutes past four in the afternoon on May nine, year two thousand-and-one* can appear differently depending on the locale. It may appear as:

- 05/09/01, 04:30 PM, if the locale used is English American.
- 09.05.2001, 16:30, if the locale used is German.

Locales also specify thousand separators and decimal symbols for numbers. They determine the position of the currency symbol in relation to the currency amount. They also contain rules for how data is sorted (collated). For example, the words OIL, SOIL, and ÖL are sorted differently depending on the locale:

- In German the words sorted alphabetically would be: OIL, ÖL, SOIL. In German, the character O is not distinguished from Ö.
- In Swedish the words sorted alphabetically would be: OIL, SOIL, ÖL. In Swedish, the character Ö is different from O.

Locales also guide what characters are available through the computer keyboard. Users can remap their keyboards through the locale setting to get access to additional characters when typing.

Localization

Localization includes:

- Translating text strings in the user interface
- Modifying the position, height, and width of controls
- Modifying images, including the direction of images (left to right or right to left)

Siebel eBusiness Applications are localized as required by the Siebel customer base. Local language releases are translated and elements of the user interface, including buttons, error messages, reports, online help, and log files, are configured to meet local requirements. An example of this is shown in Figure 1.

The screenshot displays the Siebel eBusiness Applications interface. At the top, there is a navigation bar with tabs for 'Centro informazioni', 'Catalogo dei prodotti', 'Soluzioni', 'Formazione', 'Ricerca del partner', 'Pianificazione CHAMP', 'Richieste fondi', 'Opportunità', and 'Assiste!'. Below this, a search bar shows 'Mostra: Le mie opportunità'. The main content area is titled 'Opportunità' and contains a table with columns: 'Nuovo', 'Data di chiusura', 'Home', 'Cliente', 'Fatturato', 'Fase di vendita', 'Qualità dell'affare', and '% di probabilità'. The table lists several opportunities, with the first one highlighted in yellow. Below the table, there are tabs for 'Ulteriori informazioni', 'Attività', 'Allegati', 'Contatti', 'Presentazioni', 'Prodotti', 'Proposte', 'Preventivi', 'Team di vendita', 'Compenso stimato', and 'Fatturati'. The bottom section of the interface shows detailed information for the selected opportunity, including 'Cliente: Chase Manhattan Bank', 'Sede: Manhattan, Ny', 'Data creazione: 29/08/2001 19.21.56', 'Chiusura: 29/08/2001', 'Home: OPTY Test Sales Team', 'Team di vendita: SADMIN', 'Qualità dell'affare', 'Metodo di vendita: Accelerated Sales F', 'Fatturato: \$ 8.000.000,00', 'Impegnato: [checkbox]', 'Valor e previsto: \$ 1.600.000,00', '% probabilità: 20%', 'Verso l'alto', 'Verso il basso', and 'Descrizione: [text area]'. The interface is in Italian, demonstrating localization.

Figure 1. Localized User Interface

The features that make the product internationalized are part of the software architecture; they do not require a special version of the product.

Multicurrency Support

Siebel eBusiness Applications are designed to handle the multicurrency transaction requirements of multinational businesses. These features include automatic currency conversion and enhanced currency conversion functionality with full Euro support. Other features include:

- Currency Formatting
 - \$32.45 (U.S. format with currency symbol in front of the amount)
 - 99,40 Euro (German format with currency symbol behind the amount)

Within your Siebel application, you can conduct business in multiple currencies, define additional currencies as needed, and automatically import exchange rate information from online services. Currencies are converted as needed within your Siebel application, such as when rolling up forecasts. For information on administering currency conversion, see *Applications Administration Guide*. For information on configuring dual-currency display, see *Siebel Tools Reference*.

Platform

The *platform* determines what data can be processed in a Siebel deployment. The character set encoding and operating system language of a platform will determine what data can be handled correctly and what data will not be handled correctly by the platform.

This document will use the term *platform* in several places to discuss deployment options as well as specific functionality available in Siebel eBusiness Applications. Keep in mind that Siebel eBusiness Applications generally do not support mixed character encoding environments. The reason is that it is not technically possible to manage an environment that uses multiple character set encodings on databases and servers without a genuine risk of losing data in the process.

For example, suppose a database is set up with an English character set encoding and a user tries to insert Japanese data through a Siebel server set up for Japanese. Depending on the actual database, the effect could be that the user would be denied access to the database, or the data could get converted by the database and stored incorrectly as unreadable characters (substitution characters), resulting in loss of the original Japanese data.

Siebel Language

The *Siebel Language* is seen in the Siebel user interface as well as in the messages and seed data installed with the application. The Siebel language is independent from the language of the data that the user enters in the Siebel database. The languages allowed in data are constrained only by the character encoding of the database platform. For example, although a user may be using an English version of a Siebel application installed with a Western European code page database, the user can enter a contact name in French, because all French characters are representable in the Western European code page.

Supported Character Sets and Collation

2

This section provides information on:

- [Supported Traditional and Unicode Character Sets on page 26](#)
- [Collation of the Local and Sample Database on page 31](#)

Supported Traditional and Unicode Character Sets

This section provides information on Siebel-supported traditional (non-Unicode) and Unicode character sets. In this section the terms character set and code page will be used to cover synonymous terms used by the various platform vendors as mentioned in [Chapter 1, “Overview of Global Deployment.”](#)

Traditional Character Sets

Before the emergence of Unicode, traditional character sets were available to address storage and processing requirements for a specific language or group of languages. Examples of traditional character sets are Western European, covering languages spoken in Western European countries as well as in all countries in the Americas, and Japanese covering the Japanese language.

Because of the regional aspect of traditional character sets, character data for languages not part of the character set cannot be processed in the same environment. Therefore, when a need to process data from belonging to multiple character sets arise, customers are forced to provide multiple environments.

Also, as character sets are expressed in code pages, the numeric representation of a character in one code page may be different from the representation in another code page, and often the character does not even exist.

For example, the letter a-umlaut (ä) in the Western European character set does not exist in the Arabic character set. In a Western European code page, like 1252 or ISO 8859-1, the a-umlaut occupies code point E4 (Hex value). In the Arabic code page, like 1256 or ISO 8859-6, the E4 code point is an Arabic character and not the a-umlaut. This means that it is not possible to represent neither the umlaut character on an Arabic system, nor the Arabic character in a Western European system.

There are a set of characters that are common in most generally used traditional character sets and code pages. These characters are known as the ASCII characters. They include the common characters used in the English language and they occupy the first 128 code points (Hex 00-9F) in the traditional code pages.

For more information, see *System Requirements and Supported Platforms*.

Unicode Character Set

To meet the needs of global operations, a number of software and hardware providers started the Unicode Consortium and created a Unicode standard during the 1990's. The repertoire of this international character code for information processing includes characters for the major scripts of the world, as well as technical symbols in common use. Unicode can represent more than 65,000 characters. Unicode character encoding treats alphabetic characters, ideographic characters, such as Kanji, and symbols identically, which means that they can be used in any mixture with equal facility.

A number of competing encodings have emerged during this period. These encodings are encoding formats which do not affect the assignment of glyphs to code points, which is one of the reasons that Unicode is so useful. The two most popular encodings today are UCS-2 and UTF-8.

UCS-2

UCS-2 stands for Universal Character Set - 2 Bytes. In this standard all characters are represented by two bytes no matter the origin.

The UCS-2 standard is supported on the Microsoft (MS) SQL database and on the IBM database, DB2 UDB.

UTF-8

UTF-8 stands for Unicode Transformation Format, 8-bit Encoding. It was initially developed to address the use of Unicode character data in 8-bit UNIX environments. UTF-8 is an encoding of Unicode which is more efficient for storage of English (ASCII), whereas other language data is expanded and can be represented by up to four bytes. For example, English (ASCII) characters are stored using one byte per character, accented European characters use two bytes, and Asian languages are stored using three bytes per character.

The UTF-8 standard is supported on some of the Oracle databases and on the IBM database, DB2 UDB.

NOTE: Although the UTF-8 standard is supported on IBM and Oracle databases, this does *not* imply that Siebel Systems supports deployment on these particular databases. For more information, see *System Requirements and Supported Platforms*.

UTF-16

UTF-16 is analogous to UCS-2. It is a two-byte representation of all characters and may replace the UCS-2 standard in the future. This encoding was developed after it became clear that 16-bits would not be sufficient to represent all of the modern languages. UTF-16 can access 63,000 characters as single Unicode 16-bit units and an additional one million characters through a mechanism known as surrogate pairs. For surrogate pairs, two ranges of Unicode code values are reserved for the high (first) and low (second) values of these pairs. High values are from 0xD800 to 0xDBFF, and low values are from 0xDC00 to 0xDFFF. The number of characters requiring surrogate pairs should be fairly limited, because the most common characters have already been included in the first 64,000 values.

NOTE: Although the UTF-16 standard is supported on IBM and Oracle databases, this does *not* imply that Siebel Systems supports deployment on these particular databases. For more information, see *System Requirements and Supported Platforms*.

Siebel Character Set Encoding Names and Business Services

The Siebel character set encoding names are used in the following places in the Siebel application:

- **Enterprise Database Server Code Page system preference.** This variable is set during the database installation. Typically, no modification is required.
- **SIEBEL_CODEPAGE (UNIX Environment Var).** This variable accepts all supported character set encoding names except UTF-8 and UTF-16.
- **Character conversion argument.** This argument is available in the following business services:
 - **Transcode business service.** Accepts all supported character set encoding names. While this business service is provided, it will normally be used for data validation to prevent data that cannot be converted to the appropriate code page from entering or leaving the Siebel application. Whenever possible, customers should use the EAI business service or the XML Converter business service to convert data.
 - **EAI business service (MQ Series, DLL, File, HTTP, MSMQ).** Accepts all supported character set encoding names.
 - **XML Converter business service.** Accepts all supported character set encoding names.

When business services are invoked from a workflow, the valid set of encodings is controlled by a picklist. If the business services are invoked through scripting or a similar mechanism, the character set name is supplied textually.

Special Character Set Encoding Names and Business Services

In addition to the Character Set Encoding Names listed in *System Requirements and Supported Platforms*, some Siebel business services take Special Character Set Encoding Names as inputs. [Table 4](#) lists these business services, the special supported character set encoding names and whether the Special Character Set Encoding Name is accepted by the business service. These Special Encodings give additional flexibility when configuring data movement from one environment to another.

Table 4. Special Character Set Encoding Names and Business Services

Special Encoding Name	Description	Transcode Business Service	EAI Business Service	XML Converter Business Service
None	No character encoding or character conversion	No	Yes	No
Local System Code Page	Code page of the local operating system	Yes	Yes	Yes
Database Server Code Page	Code page of the enterprise server database	Yes	No	No

Updating Currency Symbols

In some situations, you may need to update your currency symbols. For example, if you are operating in a Unicode environment but your currency seed data was originally installed in a code page environment, you need to update your currencies to include currency symbols that can only co-exist in a Unicode environment. For information on performing this, see *Applications Administration Guide*.

Collation of the Local and Sample Database

Collation is the ordering relationship, or sequence, among objects. Databases have collation sequences so that items can be sorted into a certain order (such as in alphabetic order for text strings). The ability to define a collation sequence is important because different locales sort characters in different orders.

For example, letters with accents may or may not be sorted with the same unaccented letters. If you have *a*, *ä*, and *b*, the following sort orders are possible, as shown in [Table 5](#).

Table 5. Example Sort Order With Accented and Unaccented Letters

German	Swedish
a	a
ä	b
b	ä

In the preceding example, German is sorting first by diacritical, and then by character (for example, *a* is followed by *ä*, and then *b*). Swedish, by comparison, is using the *ä* as its own character, so *a* is followed by *b*, and then *ä*.

Collation Sequences and Sorting Orders

In your Siebel eBusiness Applications, the collation sequence that affects how records are sorted in the Siebel Mobile Web Client is determined by the collation sequence on the local database's character set.

A collation sequence is *not* the same as a sorting order. Collation sequences include more functionality and have a greater impact than sorting orders. A collation sequence can also affect the unicity key values, because they affect string comparisons. This also has a bearing on required SQL syntax and could result in broken functionality if the right definitions are not provided.

The collation sequence for local databases is set during the localization process for your Siebel eBusiness Applications. For example, the Authenticated Edition of Sybase SQL Anywhere allows developers to create their own custom collation sequence. For more information about the character sets that Sybase SQL Anywhere supports, see the Sybase SQL Anywhere documentation.

The collation sequence defines more than just the sort order of that character set, it also defines the association between individual characters, such as case comparisons and accented characters. As multiple characters can be associated with each sort position, it is possible to treat an accented character the same as a character without an accent.

It is important to understand the implications of this association. For example, if the characters A and Å are associated by the collation sequence in Sybase SQL Anywhere, when the user does a search for all occurrences of the word AIR, not only would entries of AIR be returned, but also entries of ÅIR.

This chapter provides information on the following:

- [Planning Your Global Deployment on page 34](#)
- [Deployment Scenarios for Installing Multiple Languages on Siebel Servers on page 37](#)
- [Integration Considerations on page 39](#)
- [Setting Up and Administering Locales on page 42](#)
- [About the Active Language on page 43](#)
- [Configuring Language Support for Browser Platforms on page 44](#)
- [Online Help on page 50](#)
- [Application-Wide Data on page 51](#)

Planning Your Global Deployment

As you begin to evaluate your global deployment needs, start with the following steps:

- 1** Determine what your base application language will be. This book assumes a base application of American English (ENU).
- 2** Consider available industry-specific products and the operating system you use.
- 3** Determine what needs to be localized. For example:
 - Menus
 - Picklists
 - Lists of Values
 - View names
 - Strings in applets
 - Reports
 - Correspondence templates
 - Communications templates (for example, for Siebel eMail Response, Send Email, or Send Fax)
 - Forecasts
 - Personalization rules
 - Workflow policies
 - Assignment rules
 - Currency
 - Online Help
 - Reference data (for example, product and catalog data)
 - Sorting

- 4 Perform a gap analysis to determine functionality that needs to be improved or turned off.

You may find that you need to perform additional steps to enable multilingual support. For example, you might want to enable multilingual Lists of Values to support multilingual picklists. In this case, evaluate your Siebel eBusiness Applications' performance for columns used in search specifications.

- 5 Configure your application based on your company's business requirements.

- 6 Deploy the application to a global user base by doing the following:

- Install the necessary Siebel server, Siebel Gateway, Web Server Extensions, and language packs. This will install the language-specific repository (SRF) files, DLLs, and configuration files (CFG).

NOTE: Verify that the NLS_LANG parameter on the Siebel Tools Client is set to AMERICAN_AMERICA.UTF8.

- Install the necessary database language packs. This will write the language-specific seed data into the database.

For more information, see the *Siebel Server Installation Guide* for the operating system you are using.

- Deploy the Siebel Mobile Web Client in the local language, by installing the necessary language packs. This will install the language-specific executables, SRF files, DLLs, and configuration files.

See *Siebel Web Client Administration Guide* and *Siebel Anywhere Administration Guide* for more information.

- Perform application administration in the language of the database. For example, in Switzerland a database could be installed as German, so all application administration seed data is in German.

NOTE: Consider that many tasks for multilingual deployments are more complex and time consuming than they would be in a single-language deployment. This includes tasks such as installing applications, administering data in multiple languages (MLOV and reference data), and training users.

Addressing Data Sharing and Privacy Issues in Europe and APAC

The customer is advised that an international implementation of the Siebel application may be subject to different data protection and privacy laws in different jurisdictions. The customer is advised to consult its legal counsel for specific information and guidance about local law.

Deployment Scenarios for Installing Multiple Languages on Siebel Servers

Consider the following scenarios when you are installing multiple languages on your Siebel servers.

Scenario 1

You are deploying only those languages shipped with Siebel eBusiness Applications, Release 7.5 and you are installing all the languages your enterprise will require during initial installation and configuration of your Siebel servers.

- Install all Siebel servers you require with all the Siebel Language Packs intended to run on each physical machine.
- Configure each server, when prompted.
- Disable any language-specific Siebel application Object Managers (SOMs) that you do not want to run on specific servers. For more information, see the *Siebel Server Installation Guide* for the operating system you are using.

Scenario 2

You are deploying only those languages shipped in Release 7.5. However, you plan to deploy some languages at a later date.

Option 1:

- Install all intended languages at once.
- Follow the previous instructions for [“Scenario 1.”](#)
- Disable any language-specific SOMs that you intend to use in a future deployment (and re-enable them when you are ready to deploy them).

Option 2:

- Install initial rollout languages.
- Follow the procedures in [“Scenario 1.”](#)
- Follow the instructions in [“Scenario 3”](#) when you are ready to deploy additional languages.

Scenario 3

You are deploying languages shipped in a subsequent Siebel eBusiness Applications release to the initial rollout.

Option 1: Add new languages to your existing physical resource allocation.

- Follow the previous instructions for “[Scenario 1.](#)”
- Import the file `omdefs.dat` from the newly installed Siebel Language Packs on the servers where they will be deployed by entering the following KSH command from within the `\SIEBEL_ROOT\binary` directory of your Siebel servers:

```
.\srvrcfg -g SiebelGateway -e SiebelEnterprise -a components -  
i SIEBEL_ROOT\bin\lang\omdefs.dat -c ALL
```

NOTE: Siebel Industry Applications customers should use `omdefs_sia.dat`.

Where:

`SiebelGateway` = the alias of your Siebel Gateway

`SiebelEnterprise` = the alias of your Siebel Enterprise Server

`SIEBEL_ROOT` = your Siebel eBusiness Applications installation directory

- Disable any language-specific SOMs that you do not intend to use on specific servers. For more information, see the *Siebel Server Installation Guide* for the operating system you are using.

Option 2: Add New Physical Resources as Part of Expanding Language Support

- Follow the procedures in “[Scenario 1](#)” while installing new languages on new physical resources.
- Create new language-specific SOMs as part of configuring each new server.
- Disable any language-specific SOMs that you do not intend to use on specific servers.

Integration Considerations

There are many issues to consider when planning your application integration for global deployments. For example, you will need to determine the Unicode or traditional character set encoding supported by the applications you are integrating and how you will validate character conversion and error trapping. For detailed information on application integration for global deployments, see the eBusiness Application Integration documentation, starting with *Overview: Siebel eBusiness Application Integration Volume I*.

For a list of third-party software and their Unicode compliance, see *Release Notes*.

Transcode Business Service

In addition to the improvements to the current Siebel eBusiness Application Integration (eAI) components, Siebel eBusiness Applications offer the Transcode business service that you can use in your workflow processes to validate and convert data from one character set encoding to another. For a list of supported character set encodings, see *System Requirements and Supported Platforms*. For information on how business services handle character conversion errors, see [“Character Conversion Error Handling” on page 41](#).

The Transcode business service supports two methods:

- **Validate.** This method confirms the input property set hierarchy or the < value > of the input property set.

Method Argument	Required	Description
ValidationMode	Yes	If set to <i>Value</i> then only the < value > is validated. Otherwise, the properties are validated for the entire hierarchy.
TargetEncoding	Yes	For the proper language-independent code parameter values, see <i>System Requirements and Supported Platforms</i> .
SourceEncoding	No	This argument is required when ValidationMode is set to <i>Value</i> and the input < value > contains binary data. This implies conversion from binary data in SourceEncoding to binary data in TargetEncoding. For the proper language-independent code parameter values, see <i>System Requirements and Supported Platforms</i> .

- **Convert.** Converts the < value > in the input property set to the target encoding in the output.

Method Argument	Required	Description
ConversionMode	Yes	Values can be <i>StringToEncoding</i> , <i>EncodingToString</i> , or <i>EncodingToEncoding</i> .
TargetEncoding	Yes	Required for <i>StringToEncoding</i> and <i>EncodingToEncoding</i> modes. For the proper language-independent code parameter values, see <i>System Requirements and Supported Platforms</i> .
SourceEncoding	No	Required for <i>EncodingToString</i> mode. For the proper language-independent code parameter values, see <i>System Requirements and Supported Platforms</i> .

NOTE: To ignore conversion errors, specify *IgnoreConversionErrors* in the method argument.

Language-Independent Code Parameter Values

For information on language-independent code parameter values, see *System Requirements and Supported Platforms*.

Character Conversion Errors in External Systems

When Siebel eBusiness Applications send data to an external application using a traditional (non-Unicode) character set encoding, the external application may not be able to accept all of the data if the data does not belong to the character set encoding of the application. If this happens, the unrepresentable character will be converted to a substitution character. The substitution character can be a question mark or upside down question mark (indicating character loss in the database) or a black box (indicating a display issue), depending on the database. This is called *unrepresentable character conversion* and the following section shows how to avoid this situation through character conversion error handling techniques.

Character Conversion Error Handling

The transport and adaptor business services have two character conversion error handling modes:

- Generate an error without sending the message
- Ignore the error, substitute replacement characters wherever possible, and send the message

These error handling modes are specified in an argument to the business service call.

There is second case that occurs when you are using a Unicode communication stream, such as XML to communicate between Siebel eBusiness Applications and an external, non-Unicode application. In this case, the character conversion can be handled by the external application after it receives the message. This could be too late and the external application might not provide the desired error handling functionality.

A solution to this would be use the Transcode business service before sending any data to the external application. You can use the Validate method in the Transcode business service to detect whether the character conversion performed by the external application will fail. If the character conversion will fail, you can chose not to send the data. For more information, see [“Transcode Business Service” on page 39](#).

Setting Up and Administering Locales

During installation, you need to set locales for your Siebel Web Clients, Siebel servers, and database. After installation, you can modify the Siebel server locale or add additional locales. [Table 6](#) shows where you can find detailed information on performing these procedures.

Table 6. Additional Information on Setting Locales

For information on...	See...
Administering Siebel server locales	For information on administering Siebel server locales, see <i>Applications Administration Guide</i>
Setting Siebel server locales	For more information, see the <i>Siebel Server Installation Guide</i> for the operating system you are using.
Setting Siebel Dedicated Client locales	On a Windows PC, locales are administered through settings found on the Control Panel. For other operating systems, see your system documentation for detailed information about locales.

About the Active Language

The active language for Siebel eBusiness Applications is determined by the following:

Siebel Server. For more information, see the *Siebel Server Installation Guide* for the operating system you are using. For deployment scenarios for installing multiple languages on Siebel servers, see [“Deployment Scenarios for Installing Multiple Languages on Siebel Servers”](#) on page 37.

Siebel Web Clients. The active language is determined by the language parameter of the object manager to which the Siebel Web Client is connected. For more information, see *Siebel Web Client Administration Guide*.

Siebel Mobile Web Clients and Siebel Dedicated Web Clients. The active language is determined by the language parameter in the configuration file. There are different configuration files for each language. They are located in a language-specific directory and have the Language parameter set appropriately. For example:

- `c:\siebel\bin\deu\uagent.cfg` (configuration file with Language parameter set to DEU)
- `c:\siebel\bin\enu\uagent.cfg` (configuration file with Language parameter set to ENU)

Users should not modify the Language parameter to change their active language. Rather, users should define the appropriate application configuration file for the language they want to run using the `/c` switch in the command line of the `siebel.exe` icon. For example, to run Call Center in German, you would define the following in the properties of the `siebel.exe` icon:

```
/c "C:\sea700\client\bin\DEU\uagent.cfg"
```

For more information on setting languages for the Siebel Mobile Web Client, see *Siebel Remote and Replication Manager Administration Guide*.

NOTE: Siebel applications use Microsoft codes for Siebel language specifications. For information on Microsoft codes, see your Microsoft documentation.

Configuring Language Support for Browser Platforms

A browser platform includes the following:

- Web browser
- Java runtime environment used by the Web browser (may be built-in or plug-in)
- Operating system where the Web browser runs

For a list of supported Web browsers, operating systems, and other client environment requirements, see *System Requirements and Supported Platforms*.

Current Web browser platform default configurations are not installed with all fonts and other resources necessary to display all Unicode characters outside the default non-Unicode code page, and require some manual setup in order to view all characters correctly in Web pages, including Siebel Web Client screens. More recent versions of operating systems, browsers, and Java runtime environments require less setup than earlier versions.

If characters from some languages appear as hollow or solid boxes or question marks, it may be due to lack of language setup on the browser platform. Question marks can also appear when Unicode characters are mishandled on the server, but box characters almost always indicate a problem at the user interface level.

While the vendor's documentation and support should be definitive on these issues, the following hints can provide a starting point for resolving any issues.

There are several contexts where Web browsers display text, including:

- HTML body text (most text in Siebel client screens)
- HTML field input (all text input in Siebel client screens)
- Tooltips
- Message boxes (alerts)
- Java applets (used in messages, toolbars, popup calculators, and calendars)

Configuring Language Support for Windows 2000

If you are using the Windows 2000 operating system, you need to complete the following procedures to configure language support:

- Enabling text display
- Adding input locales
- Configuring ToolTips and message boxes
- Configuring Java fonts

Installing Language Support for Windows 2000

Complete the following procedure to install language support for Windows 2000. This procedure also allows the correct display of body text.

To install language support for Windows 2000

- 1 Choose Start > Settings > Control Panel.
- 2 Double-click Regional Options and click the General tab.
- 3 Select all of the Language settings for the system boxes and click OK.
- 4 At the prompt, navigate to the Windows install CD or network location to copy Windows install files from.

Allow Windows to restart if requested.

Adding Input Locales for Windows 2000

Complete the following procedure to add input locales for Windows 2000.

To add input locales for Windows 2000

- 1 Choose Start > Settings > Control Panel.
- 2 Double-click Regional Options and click the Input Locales tab.
- 3 In the Installed input locales list, install your desired input locales by selecting them and clicking Add.
- 4 In the Hot keys for input locales list, set up any desired keyboard shortcuts to switch input locales.

After installing input methods, a tray icon for controlling input locale will appear in the lower right area of the Windows taskbar. To switch input locales for each application, you can left-click the tray icon or use the keyboard shortcuts you set up to switch input locales.

Configuring ToolTips and Message Boxes for Windows 2000

Complete the following procedure to configure ToolTips and message boxes in Windows 2000.

To configure ToolTips and message boxes for Windows 2000

- 1 Choose Start > Settings > Control Panel.
- 2 Double-click Display and click the Appearance tab.
- 3 From the Item drop-down list, select ToolTips.
- 4 From the Font drop-down list, select a Unicode font (for example, Arial Unicode MS).
- 5 From the Item drop-down list, select Message Box.
- 6 From the Font drop-down list, select a Unicode font (for example, Arial Unicode MS) and click OK.

NOTE: These are global changes that will affect the appearance of all applications and the Windows desktop.

Configuring Java Fonts for Windows 2000

Complete the following procedure to configure Java fonts for Windows 2000.

To configure Java fonts for Windows 2000

- 1 Install a Unicode font, such as Arial Unicode MS, on Windows 2000.
- 2 Run regedit and create the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Java VM\Font Alias
```
- 3 Create string values with the name of a Java font as name, and the name of a Windows Unicode font as value. Java has 5 font names.

It is also possible to create a .reg file with the following text and run it to install the registry entries.

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Java VM\Font Alias

"Dialog"="Arial Unicode MS"

"DialogInput"="Arial Unicode MS"

"Courier"="Arial Unicode MS"

"Helvetica"="Arial Unicode MS"

"TimesRoman"="Arial Unicode MS"
```

Configuring Language Support for Windows XP

To configure language support for Windows XP, complete the following procedures :

- [Installing Language Support for Windows XP](#)
- [Adding Input Locales for Windows XP on page 48](#)

NOTE: By default, Windows XP should have ToolTips and message boxes properly configured. If they are not, the procedure to configure them is the same as for Windows 2000.

Installing Language Support for Windows XP

Complete the following procedure to install language support for Windows XP. This procedure also allows correct display of body text.

To install language support for Windows XP

- 1 Choose Start > Settings > Control Panel.
- 2 Double-click Regional and Language Options and click the Languages tab.

- 3** In the Supplemental language support list, select the following boxes:
 - Install files for complex script and right-to-left languages
 - Install files for East Asian languages
- 4** Click Apply.
- 5** At the prompt, navigate to the Windows install CD or network location to copy Windows install files from.

Adding Input Locales for Windows XP

Complete the following procedure to add input locales for Windows XP.

To add input locales for Windows XP

- 1** Choose Start > Settings > Control Panel.
- 2** Double-click Regional and Language Options and click the Languages tab.
- 3** Click Details.
- 4** In Installed services, click Add to install the desired input services.
- 5** Click Key Settings to set up any desired keyboard shortcuts to switch input locales.
- 6** Adjust Language Bar settings, if necessary.

After installing input services, a dockable toolbar for controlling input languages appears. To switch input locales for each application, you can click the language bar or use the keyboard shortcuts you set up to switch input locales.

Configuring Language Support on Windows ME/98/95 and NT 4.0

The Windows ME/98/95 and NT 4.0 operating systems do not have cross-code page support, so in this case, Internet Explorer itself can be set up with language support.

To configure language support for Internet Explorer on Windows ME/98/95, and NT 4.0

- 1 Navigate to Windows Update (windowsupdate.microsoft.com).

NOTE: You can also access Windows Update from the Internet Explorer menu bar by choosing Tools > Windows Update.

- 2 Select the desired language support.

It is not necessary to also install a localized user interface for each language.

NOTE: You may also receive a prompt to install language support from Windows Update when you load a Web page containing characters that are not covered by installed language support.

Configuring Language Support on IBM OS/2 and Sun Solaris 8

You can also install language support for Netscape Navigator 6 on IBM OS/2 and Sun Solaris 8. For language support with this Web browser and operating system, IBM recommends using the Unicode font Times New Roman MT 30, which is available as part of the Java package for IBM OS/2. If you are installing language support for Netscape 6 on Solaris 8, you need to install the Sun Solaris locale and Unicode support packages on your Solaris system in order to display and input for Netscape Navigator 6.

Online Help

Your Siebel Client comes with localized online help. Localized online help files are located in the language-specific directories on either the server or the Mobile or Dedicated Web client.

For information on how online help is implemented or how to customize online help, see *Online Help Development Guide*.

Application-Wide Data

This section lists the different types of application-wide data that need to be considered for global deployments.

Sales Cycles

You can only have sales cycles in one language.

Periods

Periods appear in applications such as Incentive Compensation and cannot be localized.

Currency

Make sure that your Siebel eBusiness Applications use the appropriate currency for your users. For information on currency administration, including exchange rates and currency conversion, see *Applications Administration Guide*.

Phone Numbers

You will need to form phone numbers for the countries in which you will be deploying. For information on how to perform this procedure, see *Applications Administration Guide*.

Templates

There are many templates to consider when planning a global deployment.

- Proposal
- Presentation
- Reports
- Email
- Fax
- Short Message Service (SMS)

- Correspondence

Each correspondence template is in a specific language. These templates have been localized. For information on creating new correspondence templates, in any language, see *Applications Administration Guide*. Each user can specify a default correspondence template; users can choose the appropriate template for their locale.

NOTE: You can have the language tagged in the template in the appropriate language/locale field. Because the Object Manager is multilingual, you can write a single workflow rule against one name and then change the search specification on the language or locale of that template.

Literature

The items that are included in the Literature tab can be in any language. For information on adding items to the Literature tab, see *Applications Administration Guide*.

NOTE: You may want to include the language of the literature item in the item's name and description.

SmartScript

SmartScript is a language-independent module. This means that:

- A SmartScript can be run in multiple languages.
- The same logic can be used in a SmartScript, but with a localized message displayed.
- The target language can be manually selected when a script is started.
- The target language can be set automatically when a script is started.

For information on SmartScripts, see *Siebel SmartScript Administration Guide*.

Configuring Global Applications

4

This section contains information for application developers who are configuring Siebel eBusiness Applications for global deployments. The following topics are discussed:

- [Enabling Global Data on page 54](#)
- [Using the Global Time Zone Feature on page 57](#)
- [Configuring Bidirectional Capability on page 79](#)
- [The Localization Process on page 81](#)

Enabling Global Data

The following sample procedure shows how to use the Opportunity Product Business Component to enable the Product Description Field as global data.

To enable global data

- 1** In Siebel Tools, check out and lock the Opportunity Product project.
- 2** In the Object Explorer, navigate to Business Component.
- 3** In the Name field, perform a query for Opportunity Product.
- 4** In the Object Explorer, navigate to Join and add the following record:

Field	Value
Table	S_PROD_INT_LANG
Alias	S_PROD_INT_LANG
Outer Join	Yes

- 5** In the Object Explorer, select Join > Join Specification and add the following record:

Field	Value
Name	Translate
Destination Column	PAR_ROW_ID
Source Field	Product Id

- 6** In the Object Explorer, select Join > Join Constraint and add the following record:

Field	Value
Name	Language
Destination Column	LANG_ID
Value	LOVLanguage()

NOTE: LOVLanguage() is designed to work with the multilingual outbound communication functionality (MOC). You can also use Language(), but this will not work with MOC.

- 7** In the Object Explorer, navigate to Field.
- 8** In the Name field, perform a query for Product Description.
- 9** Change the Join to S_PROD_INT_LANG and the Column mapping to DESC_TEXT.

Translating a Globally-Enabled Field

To translate a globally-enabled field (such as the Product Description field in the preceding sample procedure), complete these tasks in the order in which they appear:

- 1 Expose the view that has the field you want to translate.

In the preceding example, this view should already be exposed since Products is an object that is global-data ready.

- 2 Navigate to the Product Administration screen.
- 3 In the Products list, select a product that you want to translate.
- 4 In the More Info form, click the Show drop-down list and select Translations.
- 5 Enter the translated names and language codes for the product.

NOTE: To view sample configurations, complete [Step 2](#) through [Step 4](#) of the preceding procedure. You will need to select the product whose configuration you want to view in [Step 3](#).

Using the Global Time Zone Feature

Global deployments typically span multiple countries and have users working in several different time zones. The Global Time Zone feature allows you to track dates and times consistently across time zones by using the Coordinated Universal Time (UTC) setting definition.

NOTE: UTC does *not* observe daylight savings time and should not be considered equivalent to Greenwich Mean Time (GMT), which observes daylight savings time.

The Global Time Zone feature allows you to store date and time data in a common format and provides the following key benefits:

- Users can view dates and times in their local time zone regardless of where work items were created.
- Users in different time zones can jointly handle team-based activities, such as managing service requests.
- Customer needs can be met by sales or service assistance based in various locations and time zones. For example, a service request can have fields showing the customer's date and time, which would help agents provide timely customer service.

NOTE: Windows regional settings on the application server or client machines do not affect UTC. It is recommended that you set the database to UTC because some components have time stamps that are used with system dates in the database.

Converting Data to UTC

The Global Time Zone feature converts date and time fields to and from UTC.

- Date and time data that is manually entered by users is converted to UTC based on the time zone that is stored in the user's profile. If the Time Zone field in the user's profile is NULL, the time zone defined by the Default Time Zone system preference is used.
- Date and time data that is entered into the database by the system (date and time stamp) is converted to UTC by the server.
- Date and time data that is displayed to users is converted from UTC to the time zone that is stored in the user profile. If the Time Zone field in the user's profile is NULL, the time zone defined by the Default Time Zone system preference is used.

NOTE: Date and time fields are converted to UTC. Date-only fields or time-only fields are not converted to UTC.

Enabling UTC

Although optional, it is recommended that you enable UTC. Complete the following steps to enable UTC:

- 1 Configure the database server to UTC. See [“Setting the Database to UTC” on page 59](#).
- 2 Set the UTC system preference to True.

For more information, see [“Setting UTC System Preferences” on page 59](#).

NOTE: You need to perform additional conversion steps if you are upgrading your Siebel application. For more information, see [“Enabling the Global Time Zone Feature in an Upgraded Environment” on page 63](#).

Setting the Database to UTC

To enable UTC you need to set the operating system *time* of your database server. The method of setting the system time to UTC on your database server depends on the operating system being used. Many operating systems have a time zone setting called UTC or something similar. Others have time zone options described as Greenwich Mean Time without daylight savings time—this is the equivalent of UTC. It is recommended that you use the system-level time zone unless you need to have a different time zone. In this case, you can set the time zone at the application-level by following the documentation provided by the vendor of your operating system.

NOTE: You should disable Global Time in Siebel eBusiness Applications, Release 7 if you are unable to reset the system clock to UTC on the server running the Siebel database; or if you are planning to migrate to a system environment in which you will be unable to reset the system clock to UTC on the server running the Siebel database, such as on IBM zSeries (OS/390) or IBM iSeries (AS/400).

For more information, see the *Siebel Server Installation Guide* for the operating system you are using.

Setting UTC System Preferences

To enable UTC you must set the UTC system preference to TRUE and define the Default Time Zone.

- **Coordinated Universal Time.** Turns on UTC for the entire system. The UTC system preference should be set to the following:
 - For new installations, set the UTC system preference to TRUE.
 - For upgrades, set the UTC system preference to FALSE. For more information on upgrades and UTC, see [“Enabling the Global Time Zone Feature in an Upgraded Environment”](#) on page 63.

NOTE: The UTC system preference is shipped with a default value of TRUE. The Siebel upgrade utility sets the UTC system preference to FALSE when it completes the standard Siebel upgrade process.

- **Default Time Zone.** Defines the default time zone. The default time zone is used when the Default Time Zone Field in the current user's profile is NULL.

You can access the System Preference from the application-level menu, by choosing View > Site Map > Application Administration > System Preferences.

See *Applications Administration Guide* for more information about setting system preferences.

For more information about upgrading historical data to UTC, see [“Enabling the Global Time Zone Feature in an Upgraded Environment”](#) on page 63.

Configuring Fields for UTC

Date and time fields in Siebel eBusiness Applications are enabled for UTC. This includes fields that users manually populate by entering date and time data and fields that the system populates by generating a date and time stamp. However, if you create custom date and time fields, it is important to enable them for UTC so that data entered in these fields is consistent with data entered in other date and time fields.

You enable fields for UTC by setting the Physical Type property for the column and the Type property of the field to which the column is mapped.

- Set the Physical Type property of the column to UTC Date Time.
- Set the Type property of the Field object to DTYPE_UTCDATETIME.

After you compile your changes and if the UTC system preference is set to True, the fields will be enabled for UTC.

NOTE: There are some exceptions where date and time fields are not enabled for UTC.

For more information about configuring fields, see *Siebel Tools Reference*.

Administering Time Zones

The time zones records that are shipped with Siebel eBusiness Applications as seed data include the 74 world time zones. These are the same time zones used by Microsoft in its operating systems. This includes the time zone name, its offset from UTC, and daylight savings information. Tasks for administering time zones include:

- [Modifying Time Zones](#)
- [Maintaining Translations of Time Zones on page 62](#)

Modifying Time Zones

You will rarely need to modify a time zone's regional settings. However, you may want to modify other time zone settings such as the display name or daylight savings rules, as these occasionally change.

To modify time zone settings

- 1 From the application-level menu, choose View > Site Map > Application Administration > Time Zone Administration.
- 2 In the Time Zone list, select the time zone record you want to modify.
- 3 Modify the fields as necessary.

Some fields are described in the following table.

Field	Description
Name	Name of time zone.
Standard Abbreviation	The abbreviation for the time zone.
UTC Offset	The time difference between local time and UTC, allowing for all time zones, including the off-by-30-minutes found in India and Adelaide (Australia).
DST Abbreviation	The abbreviations for daylight savings time.
DST Bias	The difference in time, if any, DST makes. For example, Indiana (a state in the U.S.) does not use DST, so Indiana is in the Eastern Time Zone (GMT + 5) during DST and in the Central Time Zone (GMT + 6) during Standard time. Therefore, Indiana would be set with GMT + 6 with a 0-bias. Another example is Germany, which uses the Offset -60 (UTC-1) and Bias 60 for DST.

Field	Description
DST Start Ordinal	Part of the rule that determines when DST starts. For example, if the rule is <i>the first Sunday in April</i> , <i>First</i> is defined in this field.
DST Start Day	Part of the rule that determines when DST starts. For example, if the rule is <i>the first Sunday in April</i> , <i>Sunday</i> is defined in this field.
DST Start Month	Part of the rule that determines when DST starts. For example, if the rule is <i>the first Sunday in April</i> , <i>April</i> is defined in this field.
DST Start Time	Start time for DST.

Maintaining Translations of Time Zones

You maintain translations of the text-based fields for each time zone using the Time Zone Administration view.

To maintain translations of time zones

- 1 From the application-level menu, choose View > Site Map > Application Administration > Time Zone Administration.
- 2 In the Time Zone list, select the time zone you want to translate.
- 3 In the Time Zone Translation list, select the menu button, and then choose New Record.

The Time Zone Translation form appears.

- 4 In the Time Zone Translation form that appears, enter the translated version of text fields.

Enabling the Global Time Zone Feature in an Upgraded Environment

When enabling the Global Time Zone feature for an upgraded Siebel environment, you should convert your historical data to make all existing date and time values consistent with the Global Time Zone feature logic. The Global Time Zone feature stores date and time values in the database adjusted to UTC time. For more information about how the Global Time Zone feature processes data and time data in UTC, see [“Converting Data to UTC” on page 58](#).

Although enabling your environment for the Global Time Zone feature is optional, it is recommended that you perform the following procedure immediately after upgrading to Release 7.

Converting historical data into a environment using the Global Time Zone feature requires performing the following tasks:

- 1** Upgrade your development and production environments to Release 7. For more information, see the *Upgrade Guide* for the operating system you are using.
- 2** Prepare your data for conversion to UTC. See [“Preparing Your Data for Conversion to UTC” on page 63](#).
- 3** Convert your historical data, using the Siebel Software Configuration Utility. See [“Updating Existing Data to UTC” on page 69](#).
- 4** Review the UTC log file for errors. See [“Reviewing the UTC Conversion Log Files” on page 75](#).
- 5** Turn on the Global Time Zone Feature. See [“Turning On the Global Time Zone Feature” on page 77](#).

Preparing Your Data for Conversion to UTC

Before you convert your data to UTC, you need to prepare your historical data.

- Configure fields and columns (if necessary) for UTC.
- Edit the `driver_utc.ucf` file for UTC conversion.
- Set the default time zone preference for each of your users.
- Save a report of your user time zones.

- Partition large tables to assure sufficient log space.
- Allocate maximum database transaction log space.

NOTE: Log space is controlled through transaction logs. On the DB2 UDB database platform, this is called transaction log. On the Oracle database platform, this is called rollback segment. On the MS SQL database platform, this is called log file.

Configure Fields for UTC

Date and time fields in Siebel eBusiness Applications are enabled for UTC based on their object types in the Siebel Repository. This includes fields that users manually populate by entering date and time data and fields that the system populates by generating a date and time stamp. However, if you create custom date and time fields, it is important to enable them for UTC so that data entered in these fields is consistent with data entered in other date and time fields.

Enable fields for UTC by setting the Physical Type property for the column and the Type property of the field to which the column is mapped.

- Set the Physical Type property of the column to UTC Date Time.
- Set the Type property of the Field object to DTYPE_UTCDATETIME.

If you configured additional columns for UTC date time, you need to add these columns to the `utc_columns.inp` file. This input file tells the UTC conversion utility which columns need to be converted to UTC.

Set Time Zones for Users

Set the default time zone preference in the table `S_CONTACT` for each of your users.

For users who do not have a default time zone set in their user profile, the time zone specified in the `server_time.inp` file will be used for UTC conversions by the conversion utility. The `server_time.inp` file is located in the `siebsrvr/bin` directory.

Edit the driver_utc.ucf File for UTC Conversion

If you create additional input files, you need to add them to the driver_utc.ucf file. There is a default set of input files, but you may need to create an additional file to specify more columns to convert.

An example from the driver_utc.ucf file appears below:

```
[File Execute Entry 7]

Type = FileExecute

File Name = $$SiebelRoot\bin\utcupgd

Check Return Code = 1

Return Code Compliance = 0

16 Bit App = 0

Command Line = /u $UserName /p $Password /c "$ODBCDataSource" /d
$DatabaseOwner /n "$RepositoryName" /g $Language /x
$DatabasePlatform /j $SiebelRoot/bin/s_camp_con_01.inp /l
$SiebelRoot/log/s_camp_con_01.log /s $SiebelRoot/bin/
server_time.inp

Number of 10 Second Wait Loops = 2000

Prompt User For Status = 0

Parallelizable Item = 0

Title Message Num = 0

Estimated Disk Space = 0

Backup Db = 0
```

To edit the driver_utc.ucf file parameters to accommodate additional input files

- 1 Create a new input file in the driver_utc.ucf file by copying a complete step from the driver_utc.ucf file and pasting the copied step immediately after the step which you duplicated.

- 2** In the new step, change the input file parameter (which follows `/j` in the command line) to the name of your new input file.

Using the example `driver_utc.ucf` file above, you need to change:

```
/j $SiebelRoot/bin/s_camp_con_01.inp
```

to

```
/j $SiebelRoot/bin/new_file_name.inp
```

- 3** In the new step, change the log file parameter (which follows `/l` in the command line) to the name of the log file which corresponds to your new input file.

Using the example `driver_utc.ucf` file above, you need to change:

```
/l $SiebelRoot/log/s_camp_con_01.log
```

to

```
/l $SiebelRoot/log/new_file_name.log
```

- 4** Repeat [Step 1 on page 65](#) through [Step 3](#) for each new input file.

- 5** Renumber the file execute entry numbers for your new step and for each subsequent step, in order.

Using the example `driver_utc.ucf` file above (and assuming no preceding steps have changed) you need to change:

```
[File Execute Entry 7]
```

to

```
[File Execute Entry 8]
```

and so on, for each subsequent step.

Save a Report of User Time Zones

Prior to running the UTC conversion utility, save a report of your user time zones as a record of the input data used during the conversion.

Partitioning Prerequisite for the Oracle Database Platform

By default, tables are partitioned by making use of month (CREATED). Make sure that the user running the UTC conversion utility has execute privilege on month. To give execute permission, connect to the database server as the tableowner and execute the following command:

```
grant execute on month to SSE_ROLE
```

Partition Large Tables to Prevent Insufficient Database Log Space

The UTC conversion utility uses input files to control the conversion of your date and time data to UTC. The conversion utility typically processes each table as a whole. However, tables with very large record counts may encounter errors due to constraints on log space at the database level.

The UTC conversion utility prevents errors which may occur due to insufficient log space at the database level by using multiple input files to partition large tables into subsets of records for processing. The UTC conversion utility updates each record set individually to convert all rows in a partitioned table.

The utility uses partition keys to control how a table is divided into record sets. For example, large tables may be divided based on the calendar month in which each record was created, resulting in twelve approximately equal-sized partitions.

Partition keys are supplied for tables that are typically very large and that generally use a lot of log space if updated as a single input file. The tables which are delivered with partition keys are:

- s_camp_con
- s_srv_req
- s_communication
- s_org_ext
- s_evt_act
- s_contact

NOTE: If you require a different partitioning method, or if you want reduced partitioning in order to optimize performance, contact Siebel Technical Support and Expert Services for assistance.

If you create additional partitioned files, you may decrease performance.

The default value for each key is customizable. You determine the way that your tables are partitioned, and you can partition your own tables that you know to have large record counts by adding or modifying the input files.

The input file includes a WHERE clause, which defines the parameters that will be used as partition keys to divide large tables into appropriately sized sections. This WHERE clause represents standard SQL that will be used to filter which records are to be updated by each input file. Verify that you are using the correct SQL syntax.

The input files are located in your siebsrvr/bin directory. Each input file represents a single database transaction.

The following example is from the s_evt_act_00.inp file. This particular file is used to define one partition of the Activities table that includes all records created in the month of January (month = 1) or February (month = 2).

```
[S_EVT_ACT]
Clause = where month(CREATED) = 1 or month(CREATED) = 2
Column = APPT_START_DT, CREATED_BY
Column = TODO_ACTL_END_DT, CREATED_BY
Column = TODO_ACTL_START_DT, CREATED_BY
Column = TODO_AFTER_DT, CREATED_BY
Column = TODO_DUE_DT, CREATED_BY
Column = TODO_PLAN_END_DT, CREATED_BY
Column = TODO_PLAN_START_DT, CREATED_BY
```

NOTE: If the *where* clause is blank, then the table will not be partitioned. It will have all records processed at once.

Allocate Maximum Log Space

Prior to running the UTC conversion utility, set the log space parameters on the database server to the maximum. The UTC conversion utility requires a large amount of log space in order to run properly.

For the Oracle database platform, allocate a single large rollback segment and take other rollback segments offline to make sure that large transactions succeed.

Updating Existing Data to UTC

To update existing date and time data in your data tables to UTC, you need to run the UTC conversion utility from within the Siebel Software Configuration Utility. This utility will help you define the required parameters for UTC conversion, and set the UTC system preference to TRUE.

The UTC conversion takes an input file containing rows of data and updates appropriate database columns to the UTC format. Each entry in the input file contains the table name, a where clause, and a list of columns with their conversion methods. The conversion method defines how to link each record to the user record from which the default time zone is derived. The value specified as the conversion method corresponds to a column whose value identifies a unique user record.

For example:

```
[TABLE_NAME]
Clause =
Column = CREATED, CREATED_BY
```

where:

- The first column value (in this example, CREATED) indicates the column name
- The second column value (in this example, CREATED_BY) indicates the conversion method

In this case, the utility will look up the profile record of the user who created the record being converted, and derive the corresponding time zone from that profile.

NOTE: For date and time values that are not associated with a user, the server conversion method should be used (conversion method = SERVER_TIME). The server conversion method uses the time zone specified in the server_time.inp file.

Server_time.inp File Syntax

You must use the correct syntax in the server_time.inp file to implement UTC. If the exact values are not specified, the UTC Conversion Utility will not work. Use the Time Zone setting on the Microsoft Windows operating system to determine the correct values for this syntax.

Enter the value and region (in quotation marks), on the second line of the `server_time.inp` file. Here is an example for United States Central Standard Time (CST) -06:00:

```
[GLOBAL] SERVER_TIME: "(GMT-06:00) Central Time (US & Canada)";
```

[Table 7](#) provides a list of the values and regions to use for this setting.

Table 7. Time Zone Settings for `server_time.inp` File

Value	Region
(GMT-12:00)	Eniwetok, Kwajalein
(GMT-11:00)	Midway Island, Samoa
(GMT-10:00)	Hawaii
(GMT-09:00)	Alaska
(GMT-08:00)	Pacific Time (US & Canada); Tijuana
(GMT-07:00)	Arizona
(GMT-07:00)	Mountain Time (US & Canada)
(GMT-06:00)	Central Time (US & Canada)
(GMT-06:00)	Mexico City, Tegucigalpa
(GMT-06:00)	Saskatchewan
(GMT-05:00)	Bogota, Lima, Quito
(GMT-05:00)	Eastern Time (US & Canada)
(GMT-05:00)	Indiana (East)
(GMT-04:00)	Atlantic Time (Canada)
(GMT-04:00)	Caracas, La Paz
(GMT-03:30)	Newfoundland
(GMT-03:00)	Brasilia
(GMT-03:00)	Buenos Aires, Georgetown
(GMT-02:00)	Mid-Atlantic

Table 7. Time Zone Settings for server_time.inp File

Value	Region
(GMT-01:00)	Azores, Cape Verde Is.
(GMT)	Casablanca, Monrovia
(GMT)	Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London
(GMT + 01:00)	Amsterdam, Copenhagen, Madrid, Paris, Vilnius
(GMT + 01:00)	Belgrade, Sarajevo, Skopje, Sofija, Zagreb
(GMT + 01:00)	Bratislava, Budapest, Ljubljana, Prague, Warsaw
(GMT + 01:00)	Brussels, Berlin, Bern, Rome, Stockholm, Vienna
(GMT + 02:00)	Athens, Istanbul, Minsk
(GMT + 02:00)	Bucharest
(GMT + 02:00)	Cairo
(GMT + 02:00)	Harare, Pretoria
(GMT + 02:00)	Helsinki, Riga, Tallinn
(GMT + 02:00)	Israel
(GMT + 03:00)	Baghdad, Kuwait, Riyadh
(GMT + 03:00)	Moscow, St. Petersburg, Volgograd
(GMT + 03:00)	Nairobi
(GMT + 03:30)	Tehran
(GMT + 04:00)	Abu Dhabi, Muscat
(GMT + 04:00)	Baku, Tbilisi
(GMT + 04:30)	Kabul
(GMT + 05:00)	Ekaterinburg
(GMT + 05:00)	Islamabad, Karachi, Tashkent
(GMT + 05:30)	Bombay, Calcutta, Madras, New Delhi
(GMT + 05:45)	Kathmandu

Table 7. Time Zone Settings for server_time.inp File

Value	Region
(GMT + 06:00)	Almaty, Dhaka
(GMT + 06:00)	Colombo
(GMT + 07:00)	Bangkok, Hanoi, Jakarta
(GMT + 08:00)	Beijing, Chongqing, Hong Kong, Urumqi
(GMT + 08:00)	Perth
(GMT + 08:00)	Singapore
(GMT + 08:00)	Taipei
(GMT + 09:00)	Osaka, Sapporo, Tokyo
(GMT + 09:00)	Seoul
(GMT + 09:00)	Yakutsk
(GMT + 09:30)	Adelaide
(GMT + 09:30)	Darwin
(GMT + 10:00)	Brisbane
(GMT + 10:00)	Canberra, Melbourne, Sydney
(GMT + 10:00)	Guam, Port Moresby
(GMT + 10:00)	Hobart
(GMT + 10:00)	Vladivostok
(GMT + 11:00)	Magadan, Solomon Is., New Caledonia
(GMT + 12:00)	Auckland, Wellington
(GMT + 12:00)	Fiji, Kamchatka, Marshall Is.

Running the UTC Conversion Utility

The following procedure will convert your date and time data to UTC.

To run the UTC conversion utility from the Siebel Software Configuration Utility

- 1 Launch the Siebel Software Configuration Utility by one of the following methods:
 - From the Microsoft Windows platform, launch the utility by selecting Start > Programs > Siebel Servers 7.0 > Configure DB Server.
 - On the UNIX platform, launch the utility by first setting the environment variables:

setenv LANGUAGE ENU (where ENU represents your display language)

setenv SIEBEL_ROOT SIEBEL_ROOT (where SIEBEL_ROOT is the name of your Siebel root directory.)

Then type the command:

```
dbsrvr_config.ksh
```

The Siebel Software Configuration Utility appears.

NOTE: The values you enter into the Siebel Software Configuration Utility are case-sensitive. You may not type spaces in parameter values; use underscores (_) instead.

- 2 Enter the required parameters to run the UTC conversion utility.

See [Table 8 on page 74](#) for a list of dialog boxes, options, and required values.

The UTC conversion utility updates your existing data. For columns configured for UTC, the UTC conversion utility adjusts the historical date and time values to their UTC equivalent. After successfully converting the UTC-enabled date and time fields, the conversion utility sets the value for the UTC system preference to TRUE.

The Siebel Software Configuration Utility checks for errors, and writes any errors to a log file.

- Review the log file generated by the UTC conversion, and resolve errors as necessary. See [“Reviewing the UTC Conversion Log Files” on page 75](#).

NOTE: If the UTC conversion fails for any reason, you must review the log files and resolve any errors encountered. Then you need to rerun the UTC conversion utility.

UTC Conversion Parameters

Table 8 shows the UTC conversion parameters to use with the Siebel Configuration Utility.

Table 8. Siebel Software Configuration Utility

At this prompt	Enter or select the following
Siebel Enterprise Parameters: Siebel Gateway Address	Siebel Gateway Address (This is typically the machine name.) Enterprise Server Address
Installation and Configuration Parameters: Siebel Server Directory	Siebel Server Directory (This is the SIEBSRVR_ROOT directory, for example, D:\sea7xx\siebsrvr.)
Installation and Configuration Parameters: Siebel Database Server Directory	Database Server Directory
Database Server Options: Siebel Database Operation	Run Database Utilities
Database Utilities: Database Utility Selection	Coordinated Universal Time (UTC) Conversion
Installation and Configuration Parameters: Language Selection	Base language of your Siebel application (This is the primary language of your prior environment.)
Installation and Configuration Parameters: RDBMS Platform	RDBMS Platform
Installation and Configuration Parameters: ODBC Data Source Name	ODBC data source ¹ (This was created automatically by the Siebel server installation, using the format SiebSrvr_EnterpriseName.)

Table 8. Siebel Software Configuration Utility

At this prompt	Enter or select the following
Installation and Configuration Parameters: Database User Name	Database User Name (User name of the Siebel administrator, for example, admin) Database Password
Installation and Configuration Parameters: Table Owner	Table Owner Name (Note for Microsoft SQL: this is the login for the owner of the database, not necessarily the default owner of the database in DBO.) Table Owner Password
UTC Parameters: Repository Name	Repository Name
Configuration Parameter Review	Review the parameters you have defined and then click Finish.

- To find the name of the ODBC data source, go to Start > Settings > Control Panel > ODBC data source. Click the System DSN tab to find the name of the ODBC data source.

Reviewing the UTC Conversion Log Files

After the UTC conversion utility runs, if errors are encountered, the utility records those errors to log files. The log files are located in the siebsrvr\LOG directory. The default log files are described in the following list. Some of the errors listed in the log file are acceptable, but others will require resolution. An example of an acceptable error is when the utc_drop_temp_tab.log file generates the error ORA-00942: table or view does not exist, when initially dropping the S_CONTACT_TMP and S_TIMEZONE_TMP tables. You may have additional log files if you edited the input file. Carefully review the log files for errors.

```

utc_drop_temp_tab.log
null_timezone.log
utc_insert_to_tmp_tab.log
utc_create_tmp_ind.log
utc_run_stats.log
s_camp_con_00.log
s_camp_con_01.log
s_camp_con_02.log
s_camp_con_03.log
s_communication_00.log

```

```
s_communication_01.log
s_communication_02.log
s_communication_03.log
s_evt_act_00.log
s_evt_act_01.log
s_evt_act_02.log
s_evt_act_03.log
s_evt_act_04.log
s_evt_act_05.log
s_contact.log
s_org_ext.log
s_srv_req.log
utc_columns.log
denorm.log
utc_drop_temp_tab2.log
```

Review the log files that were generated by the UTC conversion utility, including any custom log files that you may have created, and resolve errors as necessary. If the UTC conversion utility was interrupted after it encountered an error, you need to manually re-launch the UTC conversion utility.

Manually Launching the UTC Conversion Utility

If the UTC conversion fails for any reason, you need to review the log files and resolve any errors encountered. Then you need to rerun the UTC conversion utility.

To manually launch the utility

- Type the following command from the SIEBEL_ROOT\siebsrvr_root\BIN directory:
 - On Microsoft Windows, type: `siebupg.exe /m master_utc.ucf`
 - On UNIX, type: `srvrupgwiz /m master_utc.ucf`

Turning On the Global Time Zone Feature

After you have confirmed that your UTC conversion was successful, turn on the Global Time Zone feature by completing the following procedure.

To activate the Global Time Zone feature

- 1 Set the database server to UTC.
- 2 Make sure that the UTC system preference is set to TRUE. If not, then set the UTC system preference to TRUE.

See [“Enabling UTC” on page 58](#) for more information.

Performing a UTC Delta Upgrade

You may want to perform a UTC Delta upgrade if you have upgraded to Siebel eBusiness Applications, Release 7.0.x, performed the UTC upgrade, and then upgraded to Siebel eBusiness Applications, Release 7.5. Completing this will upgrade the data in columns that changed from non-UTC to UTC in the Siebel Repository.

To perform the UTC Delta Upgrade

- 1 Open `$Dbsrvr\[your database platform]\master_utc.ucf`.
- 2 Change the File Name to point to `driver_utc.delta.ucf`.

NOTE: You should replace `driver_utc.ucf` with `driver_utc.delta.ucf`.

- 3 Run UTC Configuration (`ssincfgw` or `dbsrvr_config.ksh` for UNIX) to run the UTC Upgrade process.

Running the UTC Upgrade

After adding any necessary UTC columns and creating any necessary new input files, you are ready to run the UTC Upgrade program to convert your data.

To run the UTC Upgrade on a UNIX operating system (Solaris or AIX)

- 1** Navigate to the SIEBEL_ROOT/Bin directory and run `dbsrvr_config.ksh`.
- 2** At the Siebel Database Operation screen, select Option 5, Run Database Utilities.
- 3** At the “Would you like to run `srvrupgwiz?`” prompt, enter Y to start the UTC Upgrade process.

To run the UTC Upgrade on a Windows operating system

- 1** Navigate to the SIEBEL_ROOT/Bin directory and run `ssincfgw`.
- 2** At the Siebel Database Operation screen, select Option 5, Run Database Utilities.
- 3** On the following screen, select option 2, Universal Time Code Conversion and continue with the configuration until done.
- 4** At the “Would you like to run `siebug?`” prompt, click OK to start the UTC Upgrade process.

NOTE: The time to run the UTC Upgrade process depends on the number of records in your tables.

Configuring Bidirectional Capability

Bidirectional capability allows a user interface to present data in both directions, left-to-right and right-to-left. You can accomplish this by installing a localized HEB or Arabic version of the software, or you can manually configure the software for this capability.









To manually configure this capability in Siebel clients you change the alignment in <SWEApps Root> /public/enu/files/main.css by replacing “text-align:left” with “text-align:right”.

Installing Right-to-Left Images




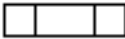
- To install the right-to-left version of certain images, you need to first install the Hebrew (HEB) language pack and then copy the files from <SWEApps Root or Tools Root> /PUBLIC/HEB/images to <SWEApps Root or Tools Root> /PUBLIC/XXX/images. Copy the ENU folder and manually alter the names of the images to swap them when there is a right/left or open/close designation. For example, change arrw_left.gif to arrw_right.gif. This method requires more user interface (UI) configuration and testing.

You need to do this because the user interface is using hundreds of images. An image on the screen is often made up of several separate pieces, with each piece coming from a separate image file. For example, most buttons on the screen are made up of a left, middle, and right piece. This creates a particular issue for bidirectional capability, because the browser must mirror the image within each individual bitmap when the direction changes to right-to-left.

To solve this issue, your Siebel eBusiness Applications come with separate mirrored images for bitmaps.

In Left-to-Right	opencap.gif	middle.gif	closecap.gif	
Button image:				
In Right-to-Left	closecap.gif	middle.gif	opencap.gif	
Button image:				

You can manually perform this by exchanging and renaming the image files, as shown in the following example:

To fix Right-to-Left	closecap.gif	middle.gif	opencap.gif	
Button image:				

NOTE: Approximately 50-100 directional images are affected.

Setting the Direction to Right-to-Left Mode on the Siebel Client

When you are configuring an application for bidirectional capability, you must set the direction to Right-to-Left mode on the Siebel Client. The tasks for performing this are described in the following section for the Siebel Mobile Web Client and Siebel Web Client.

- **Siebel Mobile Web Client.** Add the following line to the [InfraObjMgr] section in the CFG file:

```
UIDirectionality = RTL
```

NOTE: If the [InfraObjMgr] section does not already exist, you must create it.

- **Siebel Web Client.** Set UIDirectionality = RTL as a server parameter in the Object Manager. Start Srvmgr and enter the following command:

```
change param UIDirectionality = RTL for Server <Siebsrvr_Name >
```


The Localization Process

NOTE: Localizing an application includes translating the user interface and modifying other attributes to meet locale-specific requirements. Siebel Systems recommends that you complete as much configuration as possible before you begin localizing the application. There are a number of reasons for this, for example, configuration potentially alters what needs to be translated, so it is better to have a stable application. Also, the check-in and check-out mechanism in Siebel Tools is meant to be performed in only one language so if you do localization on the client during configuration, there is a potential for data loss. The localization process uses the Locale Management Utility (LMU) in Siebel Tools. For information on the Locale Management Utility, see *Siebel Tools Reference*. LMU files are UTF-8 with Byte Order Markers (BOMs).

The localization process typically includes the following steps:

- 1** Identify the applications or projects that you want to localize.
- 2** Export strings and other localizable attributes using the Locale Management Utility.

The LMU exports the objects to a flat file.
- 3** Translate strings by modifying the flat file directly or by importing the file into a separate localization development environment, modifying the locale specific attributes, and then exporting the localization result to another flat file.
- 4** Import modified string and modified object definitions into the repository using the LMU.
- 5** If necessary, search for strings or locale specific attributes that have been modified since the last export, and update the string translation or attributes localization for these changes objects.
- 6** Compile the modified projects into a repository file (SRF).
- 7** Distribute the repository file to the appropriate Siebel servers and clients.

What You Can Localize

- **Localizable Elements.** You can localize the following elements:
 - Strings
 - Seed Data
 - Bitmaps
 - Reports
 - On-Line Help
- **Non-Localizable Elements.** You cannot localize the following elements:
 - DLLs
 - Message Files
 - LOG files

Localizing an Unshipped Language

This section describes the process for localizing an unshipped language in your Siebel application. An administrator will need to perform following tasks:

- [Create New Language Subdirectories on page 83](#)
- [Configure the Object Manager on page 83](#)
- [Copy the .js File on page 84](#)
- [Test the Unshipped Language on page 85](#)

In the procedures for performing these tasks, the following variables are used:

Variable	Example
XXX	The unshipped language you are localizing; for example, Esperanto.
YYY	The base language of the application.
ZZZ	Any additional language installed in the enterprise that is not being used as XXX or YYY.

Create New Language Subdirectories

When you are localizing an unshipped language, you must first create new language subdirectories for the unshipped language.

To create language subdirectories for an unshipped language

- 1 Create a language subdirectory under < siebserver root > \objects\XXX.
- 2 Copy the SRF file from language ZZZ for the target language to the < siebserver root > /object/XXX folder if necessary.

Be aware that the user interface will display data in this language.

- 3 Create a new language subdirectory under < siebserver root > \bin\XXX and copy the CFG file used by the original Object Manager into that directory.

Configure the Object Manager

Next, you must configure the Object Manager you are going to use for testing the unshipped language.

To configure the Object Manager for an unshipped language

- 1 Start the application in the base language and navigate to Server Administration > Enterprise Configuration > Component Parameters.
- 2 Select the Object Manager you are going to use for the test. (For example, Call Center Object Manager (YYY).

NOTE: It is easier to use an existing Object Manager definition and point it elsewhere, instead of creating a new Object Manager definition.

- 3 Query in the parameter's list for *Language*.
- 4 Change the Language Code to XXX and the Object Manager Resource Language Code to YYY.
- 5 Navigate to the Application Administration > List of Values Administration and select a column that can be used as a multilingual list of values (MLOV).

For example, SR_STATUS.

- 6 Add complementary values for your new XXX language.

TIP: Do not forget the availability of Unicode characters.

- 7 Edit the CFG file for the Object Manager you chose to use in the [Siebel] section and change the Language = XXX.

Copy the .js File

When you have finished configuring the Object Manager, you must copy the .js file for the unshipped language.

To copy the .js file

- 1 Select the < siebserver root > \webmaster\ZZZ and copy it to the XXX directory.
- 2 Select the < siebserver root > \webmaster\ < build number > \scripts\ZZZ and copy it to the XXX directory.
- 3 Open the XXX directory and locate the \swemessages_zzz.js file.
- 4 Rename this file to \swemessages_xxx.js file.

CAUTION: Only rename the file, do not edit it.

- 5 Select the < siebserver root > \public\ZZZ and copy it to the XXX directory.
- 6 Select the < sweapps root > \public\ZZZ and copy it to the XXX directory.
- 7 Go to the Web server administration utility and edit the properties of the virtual directory you are using. (For example, callcenter_yyy.)

NOTE: It is easier to use an existing virtual directory and point it elsewhere, instead of creating a new virtual directory.

Test the Unshipped Language

The last procedure you need to perform is testing the unshipped language.

To test the unshipped language

- 1** Restart the server and the World Wide Web publishing service to have the new settings take effect.
- 2** Start the Object Manager and test the application.

The application user interface should appear in ZZZ language, even though the application splash screen is in YYY language.

NOTE: If you are using multilingual list of values (MLOV) columns, make sure they are available in the appropriate values for language XXX.

Localizing a Siebel Handheld Client

All Siebel Handheld Clients for 7.5 are localized in the following languages:

- Spanish (ESN)
- French (FRA)
- Italian (ITA)
- Portuguese Brazilian (PTB)
- German (DEU)
- Dutch (NTL)
- Danish (DAN)
- Swedish (SWE)
- Japanese (JPN)
- Simplified Chinese (CHS)

The user interface (for example, screens, labels, and messages) and supporting text files and list of values (LOVs) are fully translated. Some areas, such as the Calendar Date acronyms (for example, MTWTFSS) and error messages cannot be localized by the customer and require assistance from Siebel Global Services. This means that customers requiring a localization in a different language cannot fully localize their applications on their own.

Supported Microsoft Windows Code Pages

For information on supported Microsoft Windows code pages for the Siebel Handheld Client, see *Siebel System Requirements and Supported Platforms*.

Unicode Support for Siebel Handheld Clients

The Siebel Handheld Client is double-byte and enabled for Unicode. Therefore, the software application supports Unicode as long as the handheld device does. For more information on supported handheld devices and operating systems, see *Siebel System Requirements and Supported Platforms*.

Adding Handwriting Recognition Software to a Supported Handheld Device

It is not recommended or supported for customers to add third-party software to a handheld device in order to allow data input in characters not supported by default on the device. The virtual memory that is available on a handheld device is significantly smaller than what is available on a desktop or laptop computer. Adding this type of software to a handheld device can result in the Siebel application being unable to load properly on the device.

This chapter provides information on the following:

- [Siebel Unicode Architecture on page 90](#)
- [Monolingual Data Support Deployment on page 94](#)
- [Multilingual User Interface Deployment on page 96](#)
- [Unicode Database Deployment on page 98](#)
- [Non-Unicode Database Deployment on page 99](#)

Siebel Unicode Architecture

The Siebel software architecture has been designed to fully support Unicode, including internal processing. There are many benefits to using Unicode, such as the ability to show multilingual data in the same session. [Figure 2 on page 91](#) provides a graphical representation of the architecture, while [Table 9 on page 92](#) describes the flow of data through the Siebel Unicode architecture.

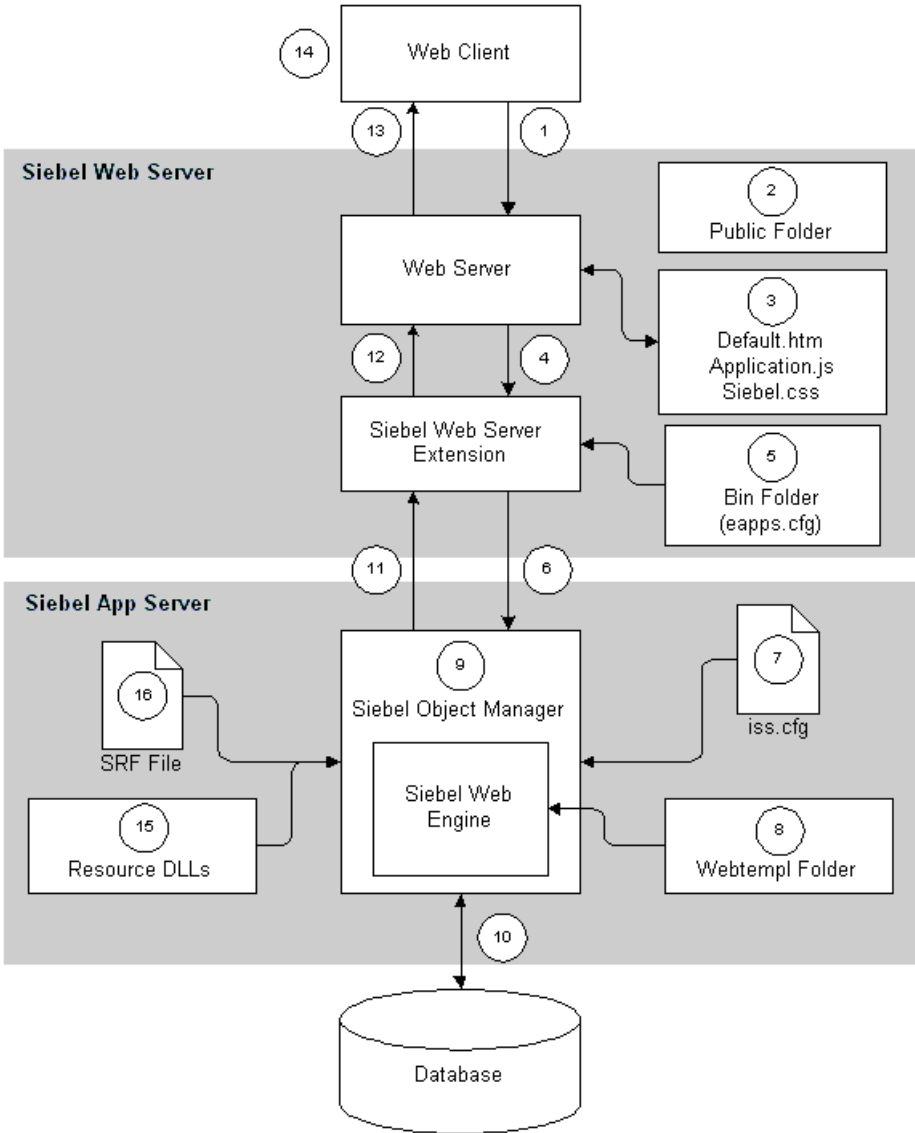


Figure 2. Siebel Unicode Architecture

Table 9. Siebel Unicode Architecture Data Flow

Step Number	Text Encoding
1	User input is URL-encoded by JavaScript or browser.
2	JavaScript code and style sheets are in ASCII.
3	HTML pages are in ASCII, or UTF-8 if hard-coded non-ASCII content is required.
4	Web server passes URL-encoded input to Siebel Web Server Extension (SWSE).
5	Application configuration file (eapps.cfg) is in ASCII.
6	SWSE sends input (still in URL encoding) to the Siebel server. The Siebel server decodes input immediately to the Siebel internal encoding (UTF-16).
7	Configuration file (CFG) is 8-bit, presumably in ASCII.
8	Web templates are in ASCII and are language independent.
9	Most C + + code uses the Siebel internal encoding (UTF-16).
10	Object Manager uses UTF-16 to communicate with the database. The internal encoding of the database can be UTF-8, UTF-16, or a code page.
11	Siebel Web Engine (SWE) sends the HTML page (in UTF-16) to SWSE.
12	SWSE transcodes, or converts, the HTML page to UTF-8 for the World Wide Web.
13	Web server passes the HTML page (in UTF-8) to the browser.
14	Browser code (JavaScript or Java) reads the HTML page in UTF-16.
15	Resource dynamic-link libraries (DLLs) store content in UTF-16.
16	Siebel repository file (SRF) stores content in UTF-16.

Global Deployment Scenarios

When planning to deploy Siebel eBusiness Applications, review your business requirements and decide what type of deployment is appropriate. Some issues to consider are:

- Will your applications be using a Unicode or non-Unicode database?
- Do you require monolingual data support?
- Do you require support for a multilingual user interface?
- Will your implementation include mobile users?

NOTE: The local database, Sybase SQL Anywhere, is only supported in a Unicode environment, regardless of the server database code page.

For more information on supported databases, see *System Requirements and Supported Platforms*.

Monolingual Data Support Deployment

In a monolingual data support scenario, a single language is used across all Siebel deployments. This scenario simplifies data analysis and is relatively simple to configure and deploy. [Figure 3 on page 95](#) shows an example of a monolingual data support scenario.

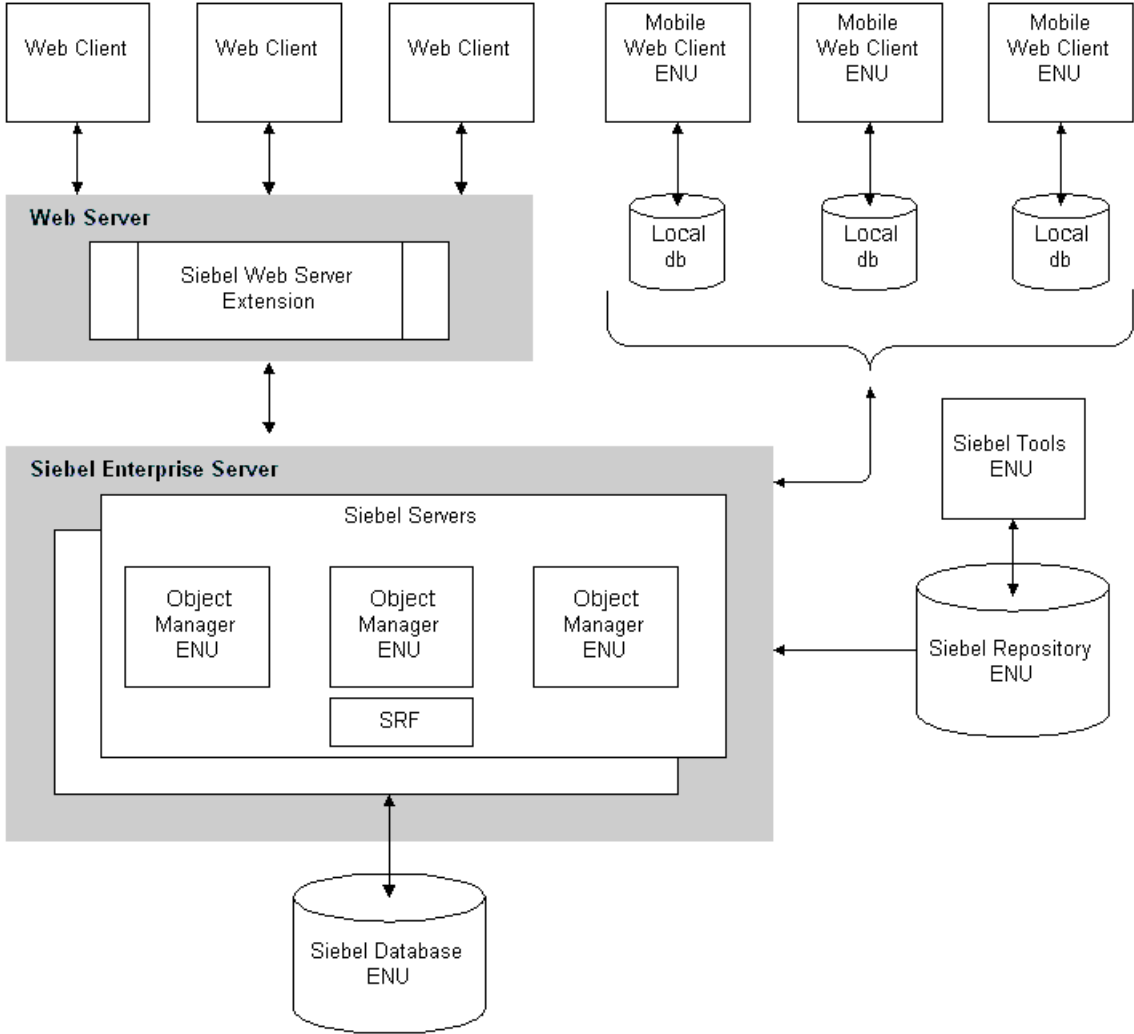


Figure 3. Example of a Monolingual Data Support Deployment

Multilingual User Interface Deployment

In a multilingual use interface scenario, multiple languages are used across the deployment. [Figure 4 on page 97](#) shows an example of a multilingual user interface deployment with users being able to choose the language they want to appear on their user interface.

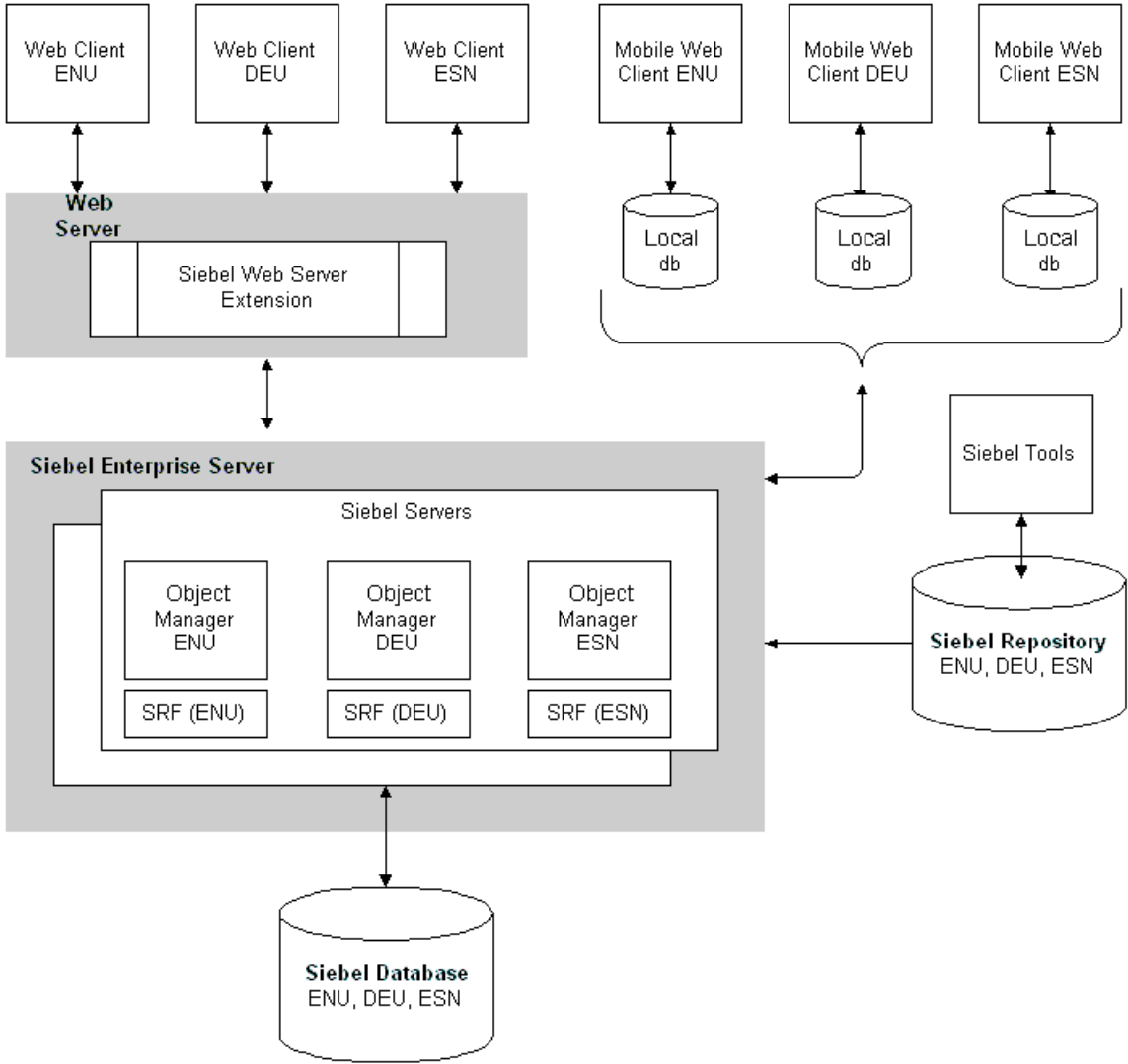


Figure 4. Example of a Multilingual User Interface Deployment

Unicode Database Deployment

When you deploy Siebel eBusiness Applications with a Unicode database, the Siebel environment is set up to process and display data and the user interface in any desired language or format. This model is designed for deployments with users who require access to the same data, but in multiple locales and language. For Release 7.5 Unicode database deployments, you must choose a database encoding of UTF-8, UTF-16, or UCS-2.

Figure 5 shows an example of a Unicode database deployment.

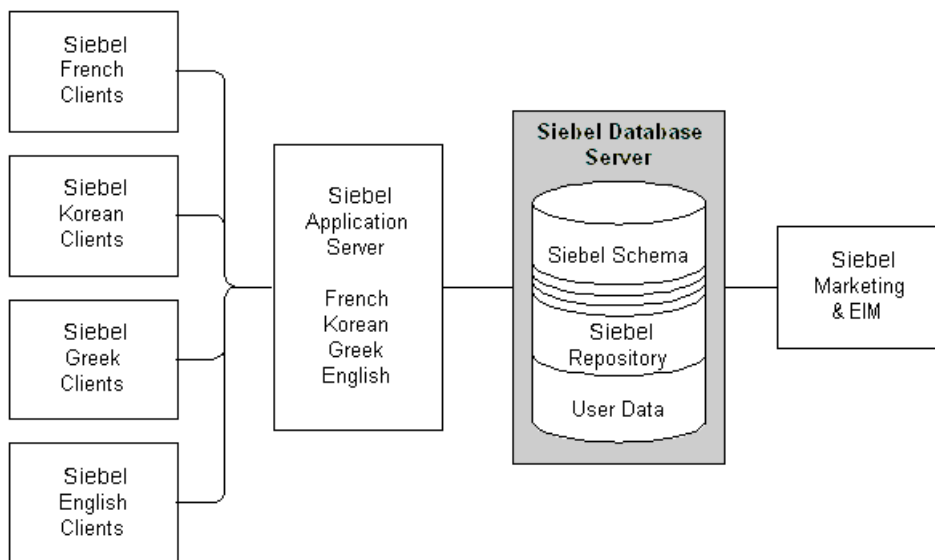


Figure 5. Example of a Unicode Database Deployment

Non-Unicode Database Deployment

If you do not have multiple users in different locales that require access to the same data, you can use a non-Unicode database in your deployment. This would be similar to the monolingual data support scenario, as shown in [Figure 3 on page 95](#).

This type of deployment is simpler to set up, configure, and deploy. However, the disadvantage to this scenario is that data can only be represented in a limited set of languages.

The decision on whether to use a Unicode or non-Unicode database should be based on whether all required languages can be represented by one code page. For Siebel eBusiness Applications version 7.5 non-Unicode database deployments, you must choose a database encoding of:

- **Code Page 1252.** This code page includes all of the characters needed for English, French, German, Italian, Portuguese, Icelandic, Dutch, Danish, Swedish, Norwegian, and Spanish.
- **Code Page 932.** This code page includes all of the characters needed for Japanese, including ASCII, which is also enough to represent English. You should be aware that while Microsoft supports code page 932, Siebel Systems does *not* support using this code page on MSSQL databases.

NOTE: Any broader set of required characters requires using a Unicode database.

Global Deployment Scenarios

Global Deployment Scenarios

Index

A

- active language
 - Siebel Clients, about determining 43
- application-wide data
 - correspondence templates, about 52
 - literature, about including in Literature tab 52
 - sales cycles, about 51
 - SmartScript, about and campaigns 52

C

- collation, definition and example 31, 32
- concurrent multiple language support,
 - definition 14
- configuration
 - fields for global time zones 64
- correspondence templates, application-wide data 52
- currencies
 - conversion, about 23

D

- data
 - formatting based on locales 18
 - historical, converting to global time zone-enabled 69
 - historical, preparing for conversion to global time zone 63
 - historical, upgrading for global time zone 63
- database, about collation sequences and example 31, 32
- date formats, examples of
 - internationalization 18
- DB2/UDB 27, 28

- dedicated Web clients
 - active language, about determining 43
 - deployment, global. *See* global deployment.
- DOS command prompt
 - Siebel Upgrade Wizard (development environment), manually launching 76
- driver_utc.ucf file
 - editing 65
 - parameters, editing 65

E

- errors
 - UTC, in conversion files 75

G

- global deployment
 - development strategy 34, 35
 - in Siebel applications 14
 - terms defined 23
- Global Time Zone
 - See also* Universal Time Zone
 - about 57
 - upgrading, about and example 63, 68

H

- help
 - online help, location of 50

I

- I18N. *See* internationalization
- internationalization
 - correspondence templates, about using 52
 - date formats 18

- defined and use 18
 - features of 19
 - locale settings, formatting of 18
 - multi-currency support 23
 - number formats 18
 - phone number formats, examples 18
- L**
- Language parameter
 - active language, about using to change 43
 - languages
 - correspondence templates, about using 52
 - multiple, working with 14
 - Siebel Language, about and example 24
 - literature, about including in Literature tab 52
 - local database, about collation sequences and example 31, 32
 - localization
 - correspondence templates, about using 52
 - defined 21, 22
 - log space, allocating 68
- M**
- mobile Web clients
 - active language, about determining 43
 - global deployment tip 35
 - MS SQL 27
- N**
- number formats 18
- O**
- online help, location of 50
 - Oracle
 - UTF8 standard 28
- P**
- phone number formats, examples 18
 - platform, defined 23
- R**
- repository
 - Siebel Upgrade Wizard, launching if conversion fails 76
 - Siebel Upgrade Wizard, manually launching 76
- S**
- sales cycles, about 51
 - sample database, about collation sequences and example 31, 32
 - Siebel Client
 - active language, about determining 43
 - Siebel Language, about and example 24
 - Siebel mobile Web client. *See* mobile Web clients
 - Siebel Repository. *See* repository
 - Siebel Upgrade Wizard, manually launching development environment 76
 - SmartScript, about and campaigns 52
 - sort orders, collation definition and example 31, 32
- T**
- tables, partitioning for UTC 67
 - templates
 - correspondence templates, about using 52
 - time zones
 - global, about 57
 - records, default 61
 - setting for users 64
 - settings, modifying 61
 - user report, saving 66
 - translations
 - localization, role of 21, 22
 - time zones, maintaining 62
- U**
- UCS2 Unicode standard 27
 - Unicode
 - standards, history of 27
 - UNIX environments (8-bit) 27
 - UTF8 standard 27

- Unicode Transformation Format. *See* UTF8
 - Unicode standard
 - Universal Character Set - 2 Bytes. *See* UCS2
 - Unicode standard
 - Universal Time Zone
 - See also* Global Time Zone
 - Global Time Zone, upgrading 63, 68
 - UNIX Unicode (8-bit) 27
 - UTC conversion utility
 - conversion log files, reviewing 75
 - conversion parameters, table 74
 - data conversion to and from 58, 60
 - failure of 74
 - launching 76
 - running 73
 - turning on 77
 - UTF8 Unicode standard 27
- W**
- Web clients
 - active language, about determining for 43

