SIEBEL.7

eBusiness

GLOBAL DEPLOYMENT GUIDE

MIDMARKET EDITION

VERSION 7.5

12-BD362Z

SEPTEMBER 2002

Contents

Introduction

Chapter 1. Overview of Global Deployment

Chapter 2. Supported Character Sets and Collation

Chapter 3.   Planning A Global Deployment

Chapter 4.   Configuring Global Applications

Chapter 5.   Global Deployment Scenarios

Index

Contents

Introduction

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

NOTE: All Siebel MidMarket product names include the phrase MidMarket Edition to distinguish this product from other Siebel eBusiness Applications. However, in the interest of brevity, after the first mention of a MidMarket product in this document, the product name will be given in abbreviated form. For example, after Siebel Call Center, MidMarket Edition, has been mentioned once, it will be referred to simply as Siebel Call Center. Such reference to a product using an abbreviated form should be understood as a specific reference to the associated Siebel MidMarket Edition product, and not any other Siebel Systems offering. When contacting Siebel Systems for technical support, sales, or other issues, note the full name of the product to make sure that it is properly identified and handled.

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

■ 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, MidMarket Edition*. Table 1 lists additional documentation on global deployments.

Table 1.   Additional Global Deployment Documentation

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

Table 1.   Additional Global Deployment Documentation

See...	For information on...
Siebel Server Installation Guide for Microsoft Windows, MidMarket Edition	Setting locales on servers running Windows
Siebel SmartScript Administration Guide, MidMarket Edition	SmartScripts in a global environment
Siebel Tools Online Help, MidMarket Edition	Unicode support for Visual Basic and eScript
Siebel Tools Reference, MidMarket Edition	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
	Working with Visual Editors

Revision History

Global Deployment Guide, MidMarket Edition, Version 7.5

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?

■ Global Deployment Terminology

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. The following table provides a list of many of these terms.

Term Glossary

Term	Meaning
Character Set	A group of characters (alphanumeric, text, or special) usually associated with one or more languages, such as the English alphabet. There are many character sets used in the computing industry. Character sets are identified by a *character set name*.
Character Set Encoding	Also known as *character encoding*. 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 *character set encoding*. For character sets with a single encoding, the *character set encoding name* and the *character set name* are usually the same. The notable exception is Unicode. 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 *character set*, 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.
Codepage	Also known as *code page*. A Microsoft Windows implementation of a *character set encoding*. Some other vendors use the term codepage and even in Microsoft Windows, it is used for ANSI codepage (Windows) and OEM codepage (DOS), but not for ISO character sets. IBM uses a numbering system which is similar, and often identical, to Microsoft. However, IBM renumbers 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."
Codepage Number	A Microsoft Windows designator for a Microsoft Windows codepage. This number is used by the Windows operating system to identify codepages and is similar to a *character set encoding name*.

Term	Meaning
Coordinated Universal Time (UTC)	Also known as *universal time coordinated*. 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 16.
Global Deployment	The process of installing, configuring, testing, and deploying Siebel eBusiness Applications in more than one locale and language.
Internationalization	*Internationalization* (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 16.
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 *locale* refers to a set of user preference information related to the user's language and cultural conventions. For more information, see "Locale" on page 18.
Localization	*Localization* (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 19.
Multicurrency Support	A feature that allows automatic currency conversion and currency formatting. For more information, see "Multicurrency Support" on page 20.
Platform	A *platform* 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 21.
Siebel Character Set Encoding Name	The name used within Siebel eBusiness Applications to refer to a specific character set encoding.

Term	Meaning
Siebel Language	The language of the Siebel eBusiness Applications software installed on the system. For more information, see "Siebel Language" on page 21.
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 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. 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 the system requirements and supported platforms documentation for your Siebel application.

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, an Input Method Editor (IME), 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 codepage of the operating system.

- **System locale.** If you are using a Microsoft Windows operating system, this is a system-wide setting that designates which codepage is used as the default for all users on the system. These codepages 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). 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

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 .

The features that make the product internationalized are part of the software architecture; they do not require a special version of the product.
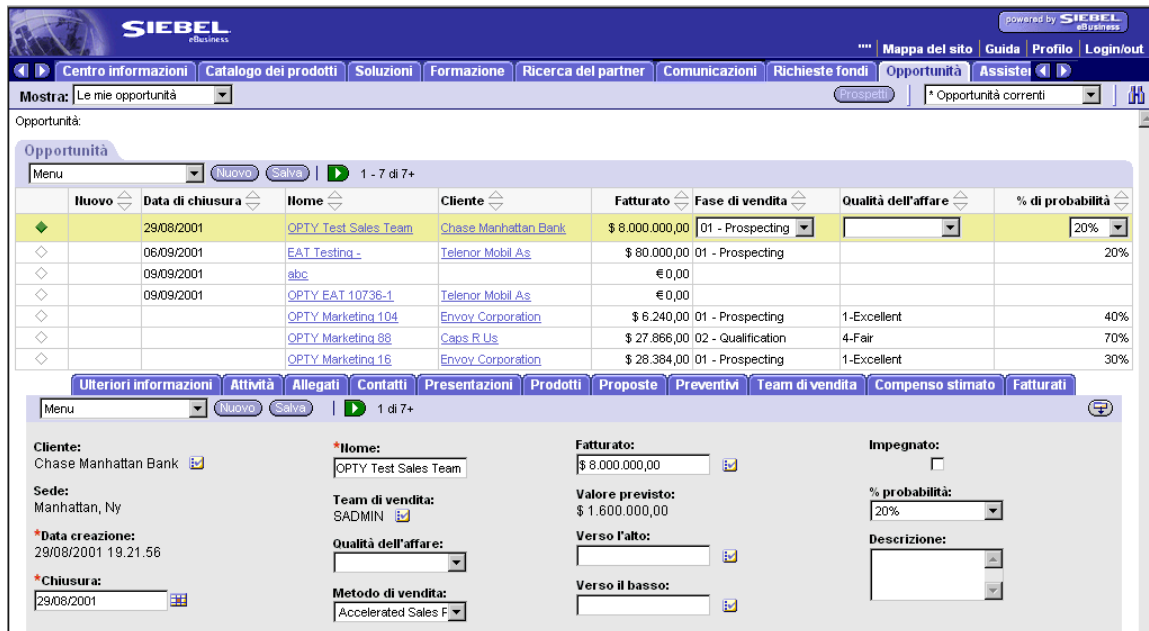
Figure 1.   Localized User Interface

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, MidMarket Edition*. For information on configuring dual-currency display, see *Siebel Tools Reference, MidMarket Edition*.

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.

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 codepage database, the user can enter a contact name in French, because all French characters are representable in the Western European codepage.

Global Deployment Terminology

Supported Character Sets and Collation 2

This section provides information on:

■ Supported Traditional and Unicode Character Sets

■ Updating Currency Symbols

■ Collation of the Local and Sample Database

Supported Traditional and Unicode Character Sets

This section provides information on Siebel-supported traditional (non-Unicode) and Unicode character sets.

Traditional Character Sets

Before the emergence of Unicode, traditional character set encodings, such as codepages, addressed the storage and processing requirements for a set of characters used in specific languages, such as English, or particular regions, such as Western Europe and Latin America. Each codepage could only handle a subset of the requirements for the global market. Table 2 on page 26 shows examples of specific Microsoft Windows codepages available for languages and regions.

In a traditional character set, character data from multiple codepages cannot be processed in the same environment. When processing data from multiple codepages the numeric representation of a character in one codepage may be different from the representation in another codepage, if the character even exists.

For example, the letter a-umlaut (ä) in the 1252 codepage, occupying code point E4 (Hex value), is not an a-umlaut in the 1256 codepage.

The only common characters supported by all codepages are ASCII. This only includes the 7-bit characters in the codepage (values 0-127). These characters are the English language characters, but not the extended ones used by most European languages, for example. Though this mode works well for regional operations, global operations have a need to store and process data from multiple regions at the same time.

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 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.

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.

The UTF-16 standard is supported on the Oracle 9i database.

Siebel Character Set Encoding Names

There are a number of existing codepages to support the needs of users around the world. The Siebel application references these codepages in a standardized way in arguments to business services, environment variables, and other places. Table 2 lists these names as the Siebel Name and gives other references for comparisons to other ways of listing these codepage names. This list does *not* imply Release 7.5 support for the codepages listed in this table for Siebel databases.

Codepages should *not* be considered identical to character sets, for example, Windows codepage 1250 is not identical to charset ISO8859-2. You should examine both the codepage and the character set to determine any differences.

NOTE: Recognition is case-insensitive.

Table 2.   Siebel Character Set Encoding Names

Siebel Name	IANA Name	Windows Codepage	Description
UTF-8	UTF-8	65001	Unicode (UTF-8)
UTF-16	UTF-16	1200	Unicode (UTF-16)
CP1252	windows-1252	1252	MS Windows codepage 1252 (Western European)

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 Codepage system preference.** This variable is set during the database installation. Typically, no modification is required.

■ **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 can not be converted to the appropriate codepage 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 Table 2 on page 26, some Siebel business services take Special Character Set Encoding Names as inputs. Table 3 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 3.   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 Codepage	Codepage of the local operating system	Yes	Yes	Yes
Database Server Codepage	Codepage 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 codepage 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, MidMarket Edition*.

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.

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 SQL Anywhere allows developers to create their own custom collation sequence. For more information about the character sets that SQL Anywhere supports, see the 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 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.

Collation of the Local and Sample Database

Planning A Global Deployment   3

This chapter provides information on the following:

- Planning Your Global Deployment

- Deployment Scenarios for Installing Multiple Languages on Siebel Servers

- Integration Considerations

- Choosing Fonts

- About the Active Language

- Setting Up and Administering Locales

- Configuring Language Support for Browser Platforms

- Online Help

- Application-Wide Data

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
- Siebel eMail Response templates
- 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.

See *Siebel Server Installation Guide for Microsoft Windows, MidMarket Edition*.

■ 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, MidMarket Edition* and *Siebel Anywhere Administration Guide, MidMarket Edition* 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.

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 SOMs that will not be running 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 intallation 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, MidMarket Edition*.

For a list of third-party software and their Unicode compliance, see the release notes documentation for your Siebel application.

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 Table 2 on page 26 and Table 3 on page 27. For information on how business services handle character conversion errors, see "Character Conversion Error Handling" on page 38.

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 *Value* 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 Table 4 on page 37.
SourceEncoding	No	This argument is required when ValidationMode is set to *Value* 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 Table 4 on page 37.

■ **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 *StringToEncoding*, *EncodingToString*, or *EncodingToEncoding*.
TargetEncoding	Yes	Required for *StringToEncoding* and *EncodingToEncoding* modes. For the proper language-independent code parameter values, see Table 4 on page 37.
SourceEncoding	No	Required for *EncodingToString* mode. For the proper language-independent code parameter values, see Table 4 on page 37.

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

Language-Independent Code Parameter Values

Table 4 gives a list of values for language-independent code parameters that you can use with the Transcode business service.

Table 4. Language-Independent Code Parameter Values

Language-Independent Code	Display Value	Description
None	D_None	D_No character encoding or character conversion
Local	D_Local system Codepage	D_Codepage of the local operating system
Database Server Codepage	D_DB Server Codepage	D_Codepage of the enterprise server database
UTF-8	D_Unicode (UTF-8)	D_UTF-8

Table 4.   Language-Independent Code Parameter Values

Language-Independent Code	Display Value	Description
UTF-16	D_Unicode (UTF-16)	D_UTF-16
CP1252	D_Western European (CP1252)	D_MS Windows codepage 1252 (Western European)

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, which can be a question mark or black box depending on the database, and saved in the database of the external application, overwriting the original content. 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 36.

Choosing Fonts

When deciding which font to use, consider the responsibilities of the user:

■ Users who do not need to display multiple languages on their user interface should use the default fonts that were installed with their operating system.

■ Users who need to display multiple languages should download and install the appropriate fonts.

The font you select is especially important for global deployments because:

■ The font must include the characters and glyphs (letters) for the language being used. Not every font can show every character or support every character set. For information on supported character sets, see Chapter 2, "Supported Character Sets and Collation."

NOTE: Licensed Microsoft Office 2000 users can download a Unicode font, Arial Unicode MS, at http://office.microsoft.com/downloads/2000/aruniupd.aspx.

■ String length expansion is a consideration for some languages.For example, German text is 30% larger, on average, than the equivalent English text.

Recommended Fonts

Table 5 shows recommended fonts for Western European and some Eastern European character sets.

Table 5.   Recommended Fonts for Localized Siebel Applications

Parameter in the .cfg File	Base/English	Western European Languages
LargeFont	MS Sans Serif-10-normal	Arial-8-normal
SmallFont	MS Sans Serif-8-normal	Arial-7-normal
DefaultChartFont	Arial-10-normal	Arial-10-normal
NavBarItemFont	MS Sans Serif-8-bold	Arial-8-bold
NavBarSelectFont	MS Sans Serif-8-bold	Arial-8-bold
NavBarTitleFont	Arial-Black-10-normal	Arial-Black-10-normal
AppletTitleFont	Arial-10-bold	Arial-10-bold
LargeDataFont	N/A	N/A
SmallDataFont	N/A	N/A

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 *Applications Administration Guide, MidMarket Edition*.
Setting Siebel Server locales	For information about setting locales on Siebel servers, see *Siebel Server Installation Guide for Microsoft Windows, MidMarket Edition*.
Setting Siebel Web 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.
Setting database locales	For databases different methods are used to handle the locales. Please refer to your system documentation for detailed information about locales.
Other platforms	Please refer to 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 34.

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

Mobile Web Clients and 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 Mobile Web Client, see *Siebel Remote Administration Guide, MidMarket Edition*.

You can find a list of most of the language codes in the Language Administration view or in Table 7.

Table 7. Examples of Language Codes

Language	Sublanguage	Language Code
English	English (American)	ENU
English	English (British)	ENG

Table 7.   Examples of Language Codes

Language	Sublanguage	Language Code
English	English (Australian)	ENA
English	English (Canadian)	END
English	English (New Zealand)	ENZ
English	English (Irish)	ENI
French	French (Standard)	FRA
French	French (Belgian)	FRB
French	French (Canadian)	FRC
French	French (Swiss)	FRS
French	French (Luxembourg)	FRL
German	German (Standard)	DEU
German	German (Swiss)	DES
German	German (Austrian)	DEA
German	German (Liechtenstein)	DEC
German	German (Luxembourg)	DEL
Italian	Italian (Standard)	ITA
Italian	Italian (Swiss)	ITS
Portuguese	Portuguese (Brazilian)	PTB
Spanish	Spanish (Standard and Traditional)	ESP
Spanish	Mexican	ESM
Spanish	Spanish (Modern)	ESN

NOTE: The file siebel.exe is the same for all language releases.

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 the system requirements and supported platforms documentation for your Siebel application.

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 codepage, 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 Browser Platforms

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. The Arial Unicode MS font is available for downloading at: http://office.microsoft.com/downloads/2000/aruniupd.aspx.

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.

Configuring Language Support for Browser Platforms

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

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
- 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-codepage 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

You can also install language support for Netscape Navigator 6 on IBM OS/2. 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.

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, MidMarket Edition*.

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 Compenstation 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, MidMarket Edition*.

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, MidMarket Edition*.

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, MidMarket Edition*. 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, MidMarket Edition*.

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, MidMarket Edition*.

Application-Wide Data

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

- Using the Global Time Zone Feature

- The Localization Process

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: Navigate to the Product Translation Administration view for sample configurations.

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 60.

2 Set the UTC system preference to True.

For more information, see "Setting UTC System Preferences" on page 60.

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 64.

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.

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.

For more information about configuring database servers, see *Siebel Server Installation Guide for Microsoft Windows, MidMarket Edition*.

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 64.

        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, MidMarket Edition* 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 64.

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, MidMarket Edition*.

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

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.

Field	Description
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.
DST Start Ordinal	Part of the rule that determines when DST starts. For example, if the rule is *the first Sunday in April*, *First* is defined in this field.
DST Start Day	Part of the rule that determines when DST starts. For example, if the rule is *the first Sunday in April*, *Sunday* is defined in this field.
DST Start Month	Part of the rule that determines when DST starts. For example, if the rule is *the first Sunday in April*, *April* 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 59.

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. See *Upgrade Guide for Microsoft Windows, MidMarket Edition* for information about upgrading to Release 7.

2 Prepare your data for conversion to UTC. See "Preparing Your Data for Conversion to UTC" on page 64.

3 Convert your historical data, using the Siebel Software Configuration Utility. See "Updating Existing Data to UTC" on page 70.

4 Review the UTC log file for errors. See "Reviewing the UTC Conversion Log Files" on page 77.

5 Turn on the Global Time Zone Feature. See "Turning On the Global Time Zone Feature" on page 78.

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 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.

Converting Data to UTC

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_communication
- s_evt_act

- s_srv_req
- s_org_ext
- 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
```

Converting Data to UTC

```
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 8 provides a list of the values and regions to use for this setting.

Table 8.   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)

Table 8.   Time Zone Settings for server_time.inp File

Value	Region
(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
(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

Table 8.   Time Zone Settings for server_time.inp File

Value	Region
(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
(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

Table 8.   Time Zone Settings for server_time.inp File

Value	Region
(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 selecting Start > Programs > Siebel Servers 7.0 > Configure DB Server.

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 9 on page 75 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.

3 Review the log file generated by the UTC conversion, and resolve errors as necessary. See "Reviewing the UTC Conversion Log Files" on page 77.

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

Table 9.   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

Table 9.   Siebel Software Configuration Utility

At this prompt	Enter or select the following
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`.)
Installation and Configuration Parameters: Database User Name	Database User Name (User name of the Siebel administrator, for example, `sadmin`) 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.

1. 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 listed below. 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:

```
siebupg.exe /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 59 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) 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 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 siebupg?" 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.

The Localization Process

Localizing an application includes translating the user interface and modifying other attributes to meet locale-specific requirements. This process uses the Locale Management Utility (LMU) in Siebel Tools. For information on the Locale Management Utility, see *Siebel Tools Reference, MidMarket Edition*.

NOTE: 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 Can You 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

The following procedure shows how you can localize an unshipped language in your Siebel application. In this example, 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.

To localize 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.

4 Start the application in the base language and navigate to Server Administration > Enterprise Configuration > Component Parameters.

5 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.

6 Query in the parameter's list for *Language*.

7 Change the Language Code to XXX and the Object Manager Resource Language Code to YYY.

8 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.

9 Add complementary values for your new XXX language.

> **TIP:** Do not forget the availability of Unicode characters.

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

11 Select the < siebserver root > \webmaster\ZZZ and copy it to the XXX directory.

12 Select the < siebserver root > \webmaster\ < build number > \scripts\ZZZ and copy it to the XXX directory.

13 Open this directory and locate the \swemessages_zzz.js file.

14 Rename this file to \swemessages_xxx.js file.

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

15 Select the < siebserver root > \public\ZZZ and copy it to the XXX directory.

16 Select the < sweapps root > \public\ZZZ and copy it to the XXX directory.

17 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.

18 Restart the server and the World Wide Web publishing service to have the new settings take effect.

19 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: Make sure that the multilingual list of values (MLOV) column you are using appears in with the appropriate values for language XXX.

The Localization Process

Global Deployment Scenarios   5

This chapter provides information on the following:

■ Siebel Unicode Architecture

■ Monolingual Data Support Deployment

■ Multilingual User Interface Deployment

■ Unicode Database Deployment

■ Non-Unicode Database Deployment

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 87 provides a graphical representation of the architecture.
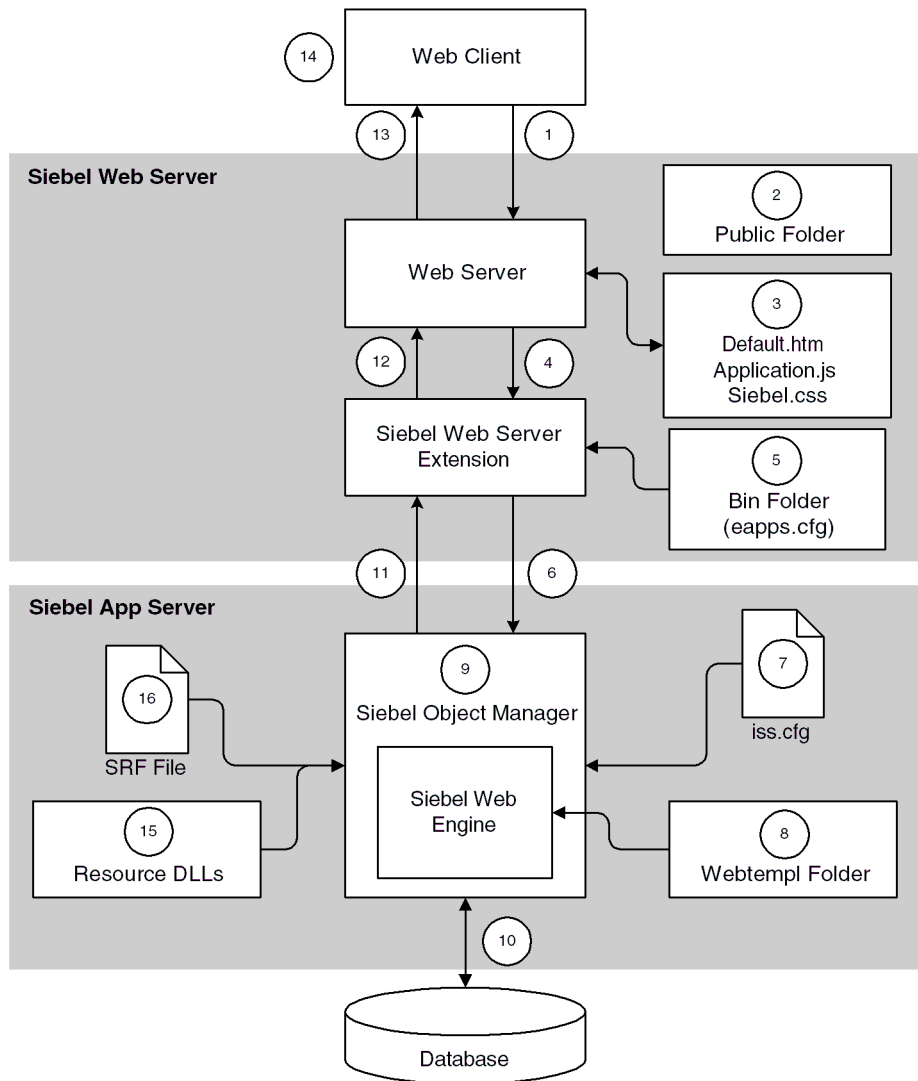
Figure 2.   Siebel Unicode Architecture

Number	Text Encodings
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 codepage.
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, SQL Anywhere, is only supported in a Unicode environment, regardless of the server database codepage.

For more information on supported databases, see the system requirements and supported platforms documentation for your Siebel application.

Global Deployment Scenarios

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 shows an example of a monolingual data support scenario.
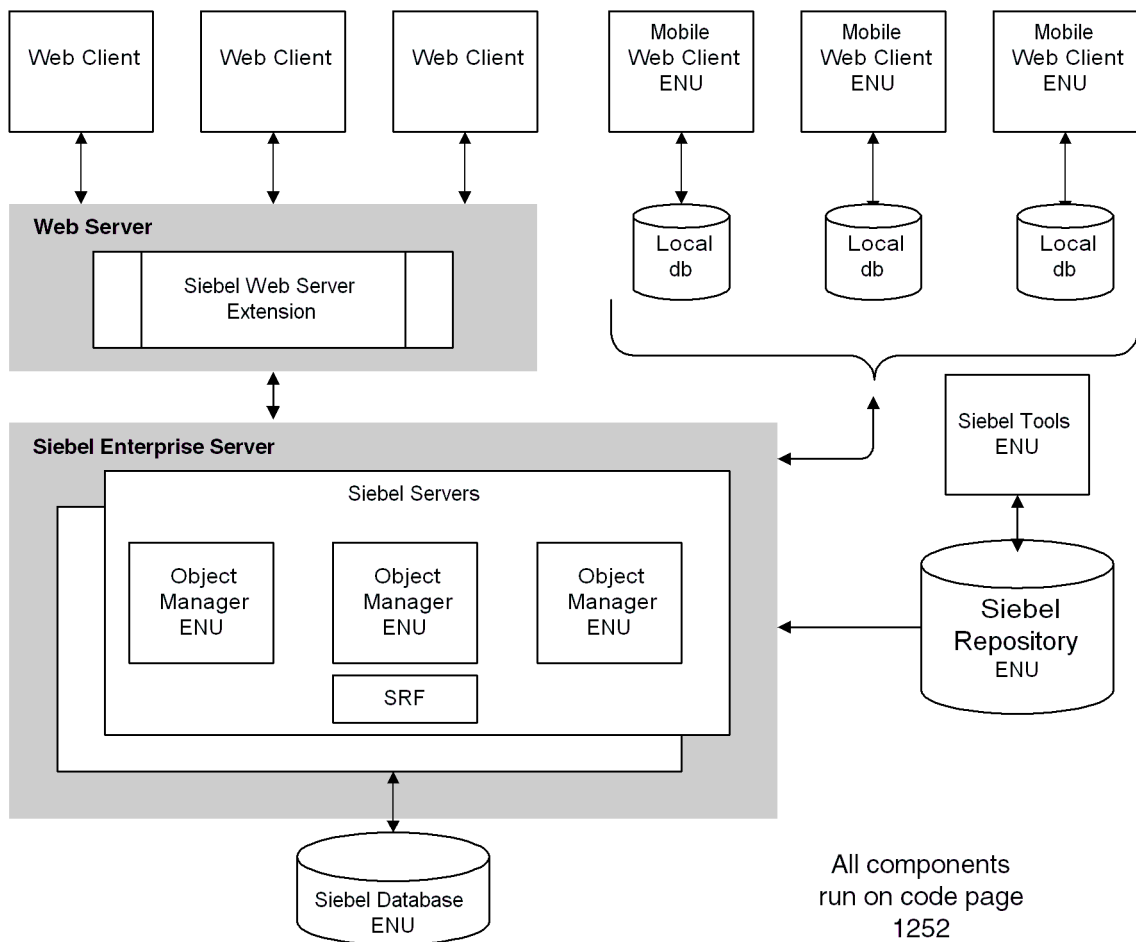
Figure 3.   Example of a Monolingual Data Support Deployment

Multilingual User Interface Deployment

In a multilingual use interface scenario, multiple languages are used across the deployment. Figure 4 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.
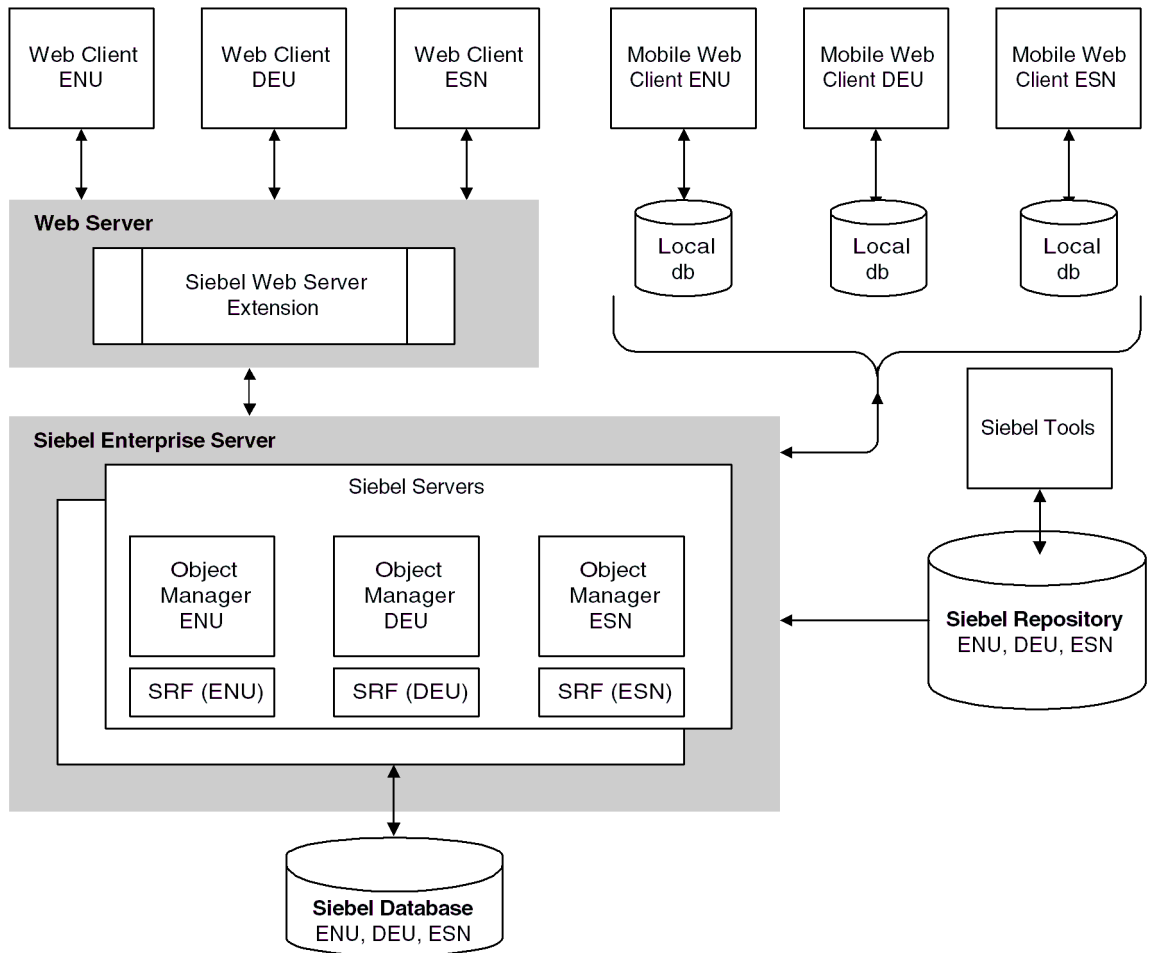
Figure 4.   Example of a Multilingual User Interface Deployment

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.
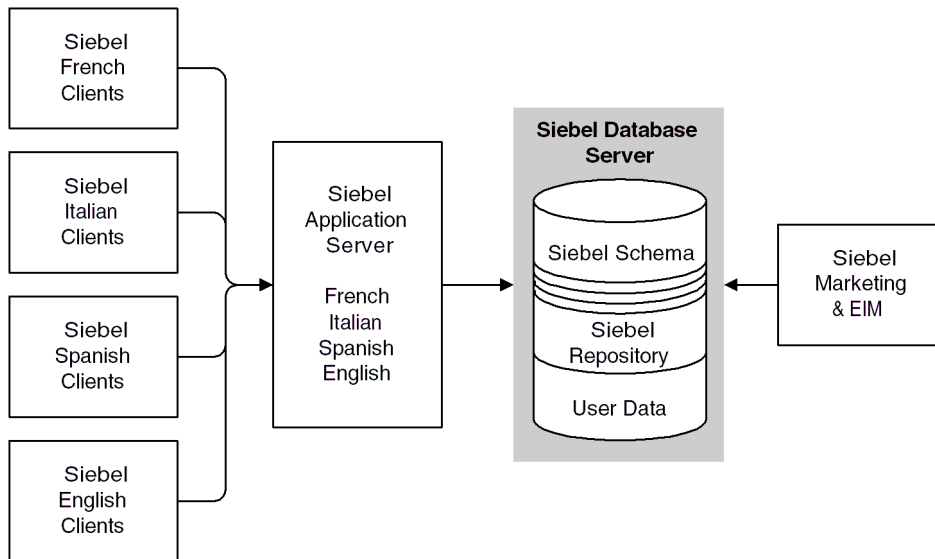
Figure 5.   Example of a Unicode Database Deployment

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 90.

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 codepage. For Siebel eBusiness Applications version 7.5 non-Unicode database deployments, you must choose a database encoding of:

■ **Codepage 1252.** This codepage includes all of the characters needed for English, French, German, Italian, Portuguese (Brazilian), and Spanish.

■ **Codepage 932.** This codepage includes all of the characters needed for Japanese, including ASCII, which is also enough to represent English. You should be aware that this codepage is *not* supported on MSSQL databases.

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

Global Deployment Scenarios

Index

A

active language
  Siebel Clients, about determining 43
application-wide data
  correspondence templates, about 53
  literature, about including in Literature
      tab 53
  sales cycles, about 52
  SmartScript, about and campaigns 53
ASCII characters 24

C

code pages
  ASCII characters 24
  character data conflicts 24
  Unicode 24
collation, definition and example 29
concurrent multiple language support,
      definition 12
configuration
  fields for global time zones 65
correspondence templates, application-wide
      data 53
currencies
  conversion, about 20, 21

D

data
  formatting based on locales 16
  historical, converting to global time zone-
      enabled 70
  historical, preparing for conversion to
      global time zone 64

historical, upgrading for global time
      zone 64
database, about collation sequences and
      example 29
date formats, examples of
      internationalization 16
DB2/UDB 25
dedicated Web clients
  active language, about determining 43
deployment, global. *See* global deployment.
DOS command prompt
  Siebel Upgrade Wizard (development
      environment), manually
      launching 77
driver_utc.ucf file
  editing 66
  parameters, editing 67

E

errors
  UTC, in conversion files 77

F

fonts
  localized Siebel application,
      recommended fonts (table) 41

G

global deployment
  development strategy 32, 33
  in Siebel applications 12
Global Time Zone
  See also Universal Time Zone
  about 58
  upgrading, about and example 64, 70

Index