Siebel

Global Deployment Guide

June 2021

Siebel
Global Deployment Guide

June 2021

Part Number: F12742-06

Copyright © 2021, Oracle and/or its affiliates.

Authors: Siebel Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display in any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or “commercial computer software documentation” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

Contents

Preface i

1 What's New in This Release 1

What's New in Siebel Global Deployment Guide, Siebel CRM 21.6 Update 1
What's New in Siebel Global Deployment Guide, Siebel CRM 21.3 Update 1
What's New in Siebel Global Deployment Guide, Siebel CRM 21.2 Update 1
What's New in Siebel Global Deployment Guide, Siebel CRM 20.8 Update 2
What's New in Siebel Global Deployment Guide, Siebel CRM 20.1 Update 2
What's New in Siebel Global Deployment Guide, Siebel CRM 19.1 Update 3

2 Overview of Global Deployments 5

Overview of Global Deployments 5
Global Deployment Terminology 5
About Date, Time, Number, and Currency Formatting 15
About Supported Character Sets 16
About Character Set Encodings and Siebel Applications 18
About the Database Collation Sequence 19

3 Planning Global Deployments 21

Planning Global Deployments 21
Understanding Your Company's Global Business Requirements 21
Expected Results of Global Deployment Planning 22
About Ideal Global Deployments 23
About the Siebel Unicode Architecture 24
About Planning Your Global Deployment 26
About Configuring Global Deployments 28
Scenarios for Installing and Deploying Siebel Languages 29
Planning Upgrades for Global Deployments 29

4 Configuring Global Deployments 31

Configuring Global Deployments 31

About Parameters for Language and Locale 31
About Configuring Language Support for Browser Platforms 35
About Unicode Fonts 35
About Integration Considerations 36
About Application-Wide Data 38
Setting Up Global Data 40
Configuring the Siebel Calendar 42
Configuring Cascading Style Sheets to Specify Different Fonts 42
Verifying Bidirectional Capability 45
Configuring Case Insensitivity 46

5 Deploying Siebel CRM with Global Time Zone 47
Deploying Siebel CRM with Global Time Zone 47
About Global Time Zone and Universal Time Coordinated (UTC) 47
Setting UTC System Preferences 52
Setting the Database Server to UTC 53
Enabling Custom Date-time Fields and Columns for UTC 54
Converting Historical Data to UTC 54
Administering Time Zones 63

6 Localizing Global Deployments 67
Localizing Global Deployments 67
About the Localization Process 67
Defining the Scope of the Localization 68
Developing a Glossary for Translatable Terminology 69
About Working with Translators 70
Localizing Lists of Values and Multilingual Lists of Values 71
Localizing an Unshipped Language 72
About Testing Globalized Software 82

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *http://docs.oracle.com/*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an email to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

1 What's New in This Release

What's New in Siebel Global Deployment Guide, Siebel CRM 21.6 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
Creating Language Subdirectories and Copying Language-Specific Files for an Unshipped Language on the Siebel Application Interface	Modified topic. Added a step to add entries in server.xml file for unshipped languages.

What's New in Siebel Global Deployment Guide, Siebel CRM 21.3 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
Configuring Cascading Style Sheets to Specify Different Fonts	Modified topic. Added example fonts for Polish and Russian, for supported UNIX operating systems.

What's New in Siebel Global Deployment Guide, Siebel CRM 21.2 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
Multiple topics	Modified topics. As of Siebel CRM 21.2 Update, the **`applicationcontainer`** directory has been replaced by two directories, as follows:

Topic	Description
	• **applicationcontainer_external** (for Siebel Application Interface) • **applicationcontainer_internal** (for all other Siebel Enterprise components) In the Siebel Application Interface installation, Web artifacts for application configurations, which were formerly located in **applicationcontainer\webapps\siebel**, now map to **applicationcontainer_external\siebelwebroot**. This directory contains subdirectories such as **files**, **fonts**, **htmltemplates**, **images**, **migration**, **scripts**, and **smc**. For more information, see *Siebel Installation Guide* .
Scenarios for Installing and Deploying Siebel Languages	Modified topic. As of Siebel CRM 21.2 Update, installing additional languages no longer requires using silent (unattended) installation. For more information, see *Siebel Installation Guide* .
Creating Language Subdirectories and Copying Language-Specific Files for an Unshipped Language on the Siebel Server	Modified topic. As of Siebel CRM 21.2 Update, directories such as **ENUSIA**, **DEUSIA**, and so on are no longer created under **SIEBSRVR\bin**. Consequentially, a step that no longer applies was removed in this topic.

What's New in Siebel Global Deployment Guide, Siebel CRM 20.8 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
Database Collation for the Local Database	Modified topic. Oracle Database SE2 is now used for the local database, replacing Oracle Database XE. The applicable configuration file section is now [LOCAL_SE] instead of [LOCAL_XE]. For more information, see *Siebel Installation Guide* .

What's New in Siebel Global Deployment Guide, Siebel CRM 20.1 Update

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

What’s New in Siebel Global Deployment Guide, Siebel CRM 19.1 Update

No new features have been added to this guide for this release. This guide has been updated to reflect only product name changes.

2 Overview of Global Deployments

Overview of Global Deployments

This chapter provides overview information about global deployments of Siebel CRM. It includes the following topics:

- *Global Deployment Terminology*
- *About Date, Time, Number, and Currency Formatting*
- *About Supported Character Sets*
- *About Character Set Encodings and Siebel Applications*
- *About the Database Collation Sequence*

Global Deployment Terminology

This guide uses many specialized terms relevant to global deployments of Siebel CRM. This topic contains the following information, which describes some of these terms and related concepts:

- *General Terminology*
- *Platform*
- *Language*
- *Locale*
- *Internationalization*
- *Localization*

General Terminology

This topic is part of *Global Deployment Terminology*.

The following table identifies and provides brief descriptions for some of the terms relevant to global deployments of Siebel CRM. Additional information is provided in subsequent topics.

Term	Meaning
Character set	A group of characters (alphanumeric, text, or special) usually associated with one or more languages or scripts (writing systems). There are many character sets used in the computing industry. Character sets are identified by a character set name, such as 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. See also *About Supported Character Sets*.
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

Term	Meaning
	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. Aside from Unicode, a prominent example of a character set with multiple encodings is JIS (Japan Industrial Standard). Shift-JIS, EUC, and ISO 2022-JP are all encodings based on JIS. As with character set names, industry standardization is minimal and there are multiple names for the same encoding. The character set encoding is also known as a code page or codepage (one word), which often refers to a vendor implementation of a character set encoding. For Microsoft Windows, the term code page is used for ANSI code page (Windows) and OEM code page (DOS), but not for ISO character sets. IBM uses a numbering system which is similar, and often identical, to Microsoft. However, IBM renumbers extensions while Microsoft does not, which can lead to statements like "You can use IBM 943 with Siebel applications and MS 932; IBM 932 is an older version." See also *About Character Set Encodings and Siebel Applications*.
Code point	A single data value representing a single character in a code page.
Global deployment	The process of installing, configuring, testing, and deploying Siebel CRM in more than one locale and language.
Internationalization	The process of making a software product that can correctly process data for any customer, including data entry and display. Internationalization makes localization possible. For more information, see *Internationalization*.
Internet Corporation for Assigned Names and Numbers (ICANN)	An Internet body that manages Internet addresses, domain names and protocol parameters. ICANN conventions are used for the World Wide Web, email, and XML.
Language	The language or languages of the installed Siebel CRM software or the Siebel database, or that you add to an existing deployment. For more information, see *Language*.
Locale	A set of user preference information related to the user's language, location, and cultural conventions, including the formatting or presentation style of data such as dates, time, numbers, and currency. For more information, see *Locale* and *About Date, Time, Number, and Currency Formatting*.
Localization	The process of readying a product for use in a particular target country through the use of a locale. For more information, see *Localization*.
Multicurrency support	A feature that allows automatic currency conversion and currency formatting. For more information, see *About Date, Time, Number, and Currency Formatting*.
Non-Unicode (traditional) character set	Non-Unicode (traditional) 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 of a collection of languages that use the same or similar alphabets. See also *About Supported Character Sets*.

Term	Meaning
Platform	A platform includes the operating system of the various entities of a Siebel CRM deployment; the database, the Siebel Servers, and the clients and the character set used by these entities. For more information, see *Platform*.
Script	A system of writing that requires graphical symbols to be placed in a certain order to communicate information. The symbols can be based on an alphabet, on pictures of objects in the world around us, or on some other system. The Roman script (sometimes called the Roman alphabet) is a script that uses 26 symbols to represent sounds made by the human mouth, organized into an alphabet. Originally the script used to write Latin (the language of the Romans), it has been extended with diacritics on many of the characters to express other sounds present in Western European languages. (In a Siebel CRM development context, a very different meaning of the term script is a program written using a language such as Siebel eScript.)
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. See also *About Supported Character Sets*. **Note:** Oracle does not support new Siebel CRM deployments on non-Unicode databases. All new Siebel CRM deployments must be Unicode.
Universal Time Coordinated (UTC)	Also known as coordinated universal time. A time scale that is 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 in order to provide a standard internal time for the Siebel applications, which can then be adapted to each user's time zone. UTC allows activities and other events to be scheduled across time zones. For more information, see *Deploying Siebel CRM with Global Time Zone*.

Platform

This topic is part of *Global Deployment Terminology*.

The *platform* determines what data can be processed in a Siebel CRM 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 guide uses the term platform in several places to discuss deployment options as well as specific functionality available in Siebel CRM.

Siebel CRM generally does not support mixed character encoding environments. The reason is that it is not technically possible to manage an environment that uses multiple character set encodings on databases and servers without a genuine risk of losing data in the process.

For example, suppose a database is set up with a Western European character set encoding and a user tries to insert Japanese data through a Siebel Server set up for Japanese. Depending on the actual database, the effect could be that

the user's data would be rejected and not stored in the database, or the data could get converted by the database and stored incorrectly as unreadable characters (substitution characters), resulting in loss of the original Japanese data.

Field Size Issues

In some cases, a user might try to enter data into the application and receive an error message saying that the language of the text entered "is not compatible with the database language" or that the length of the text entered "is bigger than the corresponding length allocated in the database."

Such an error might occur when the character set of the data does not match the character set of the database and cannot be converted without data loss. Or, the error might occur when the data string is too long to fit into the database column.

If you have an existing Siebel deployment and upgrade the database default character set to Unicode (any encoding), then all links to other information technology systems must be examined for compatibility. In many cases, data previously encoded in a traditional codepage will now be larger in terms of bytes, and could overflow the fields that are used for transfers. In such cases, Siebel Unicode data values will be truncated, especially if the data value is large enough to approach the maximum defined size of the field.

You must examine the field sizes in tables used by Siebel EAI or Siebel EIM, in any linking or replication software, and in any extension columns that have been added.

Depending on your existing code page, the languages and specific characters representing your data, and the Unicode encoding that you are migrating to, field sizes might need to be enlarged. In some scenarios, field sizes must be doubled.

Customers can set byte limits on columns in the database. At the user interface level, the input size can be controlled to make sure that this column width is not exceeded. In most cases, there is a 1:1 mapping between bytes and characters.

Field Sizes by Database Platforms

When specifying new field sizes, be careful as to whether they must be given in bytes or in characters. Field size units will vary by the RDBMS vendor:

- **Oracle Database.** Specify field sizes in characters (using character semantics).
- **IBM DB2.** Specify field sizes in bytes.
- **IBM DB2 for z/OS.** Specify field sizes in bytes.
- **Microsoft SQL Server.** Specify field sizes in bytes.

Note: For more information about using IBM DB2 for z/OS, see *Implementing Siebel Business Applications on DB2 for z/OS* and *Siebel Database Upgrade Guide for DB2 for z/OS* .

Considerations for Deployments Using Code Page 932 (Shift-JIS)

The following are special considerations for Japanese-language deployments using Shift-JIS, which is also known as Code Page 932 (or 936 on IBM DB2, or JA16SJIS on Oracle Database). It is assumed that the client computer uses the same code page as the database.

- Some Japanese characters require two bytes per character. However, note that user input can include both single-byte and double-byte chars, in combination. Because there is no direct correlation between length limit and byte limit in the case of double-byte languages, it is impossible to provide direct validation of byte limits. For validation purposes, the assumption should be that all characters typed are double-byte. This may leave unused fields when only single-byte characters are used. Custom validation based on analysis of the input string might be able to solve this problem, if a customer chooses to implement this.

- If your deployment is migrating from Code Page 932 (Shift-JIS) to Unicode, then you must be careful with characters entered on a system where the data is stored in Code Page 932.

  Over 300 characters present in Code Page 932 have only one representation in Unicode, so when these characters are moved to a Unicode system, they are converted permanently to the new value used in Unicode. Because of this conversion, users will see that the character they originally entered has been slightly changed, but the meaning should be the same as before.

  An example is the replacement of the WAVE DASH character by the FULL WIDTH TILDE character, often used in expressing appointment times. There is no correction for this situation, because it is a result of the design of both Code Page 932 and Unicode character sets, which are industry standards.

Limitation for Arabic and Numeric Fields

Although the Arabic language is supported, Arabic digits cannot be used in numeric fields in Siebel CRM.

Language

This topic is part of *Global Deployment Terminology*.

The *language* for a Siebel application can mean multiple things, and might involve different system or application elements. These elements are independent from the language of the data that the user enters in the Siebel database. You must install seed data according to how you want to use languages in the Siebel applications.

Primary Language, Active Language, and Resource Language

The term language has the following major meanings with respect to Siebel CRM:

- The *primary language* (sometimes called the *base language*) is the first language installed for this Siebel product installation, particularly the first language installed in the Siebel database.

  > **Note:** In general, the term primary language refers to the first language installed in the Siebel database. Sometimes this phrase also refers to the language for Siebel Enterprise Server messages and logging, which is specified during installation and initial configuration. For most deployments, these would be the same language. See also *Siebel Installation Guide* .

- For the Siebel Mobile Web Client or Developer Web Client, the user can explicitly specify the language to use for an application session by double-clicking the corresponding shortcut, where the language is installed and available.

  - The *active language* is the language in effect for an individual user's session and the language of user interface elements, including multilingual lists of values (MLOVs) that have been enabled. The same language is used for system messages (if the resource language is not separately defined).
  - For a Siebel Web Client session, the language is determined by the Application Object Manager component that is invoked through the specified URL. This language cannot be changed by the user, except by logging in to a different language-specific Application Object Manager where the language is installed and available. The URL includes the language code.

- The *resource language*, if it is defined, is used as the default language for system messages.

Languages and Siebel Installations and Upgrades

Installing Siebel languages on the Siebel Server (or on other Siebel components) installs the language-specific run-time environment: resource libraries such as DLL files, configuration (CFG) files, error messages, help files, and so on. In general, you install the same languages on all of the components in your Siebel Enterprise. For more information, see *Scenarios for Installing and Deploying Siebel Languages*.

When you install the Siebel database for a new installation, language-specific seed data is added for the primary language only. For multilingual deployments, you must add seed data for additional languages separately after your initial Siebel database installation. The Siebel database also includes the Siebel runtime repository.

When you upgrade the Siebel database from a prior Siebel version, all of the existing languages are upgraded at the same time. For more information, see *Siebel Database Upgrade Guide* .

The languages allowed in data are constrained only by the character encoding of the database platform. For example, although a user might be using a U.S. English version of a Siebel application with a Western European code page database, the user can enter or view contact data in French, because all French characters are representable in the Western European code page.

With a Unicode code page, and appropriate fonts locally installed, languages using dissimilar scripts, such as French and Japanese, can be used together.

Language Codes

Each language code used by Siebel CRM uses a three-letter code, such as ENU for U.S. English, FRA for French, THA for Thai, and so on. Using language codes with only two characters does not work and is not supported.

> **Note:** For a list of the languages supported by Siebel CRM, and their language codes, see 1513102.1 (Article ID) on My Oracle Support. They are also listed in the second table in *Configuring Cascading Style Sheets to Specify Different Fonts*. Special requirements apply when you localize an unshipped language, as noted in *Localizing an Unshipped Language*.

Languages and Application Development and Deployment

For application development using Siebel Tools or Siebel Web Tools, you set the language mode to work with object definitions for a particular language. For information about the language mode in Siebel Tools or Siebel Web Tools and how to set it, see *Using Siebel Tools* . The Siebel runtime repository includes language-specific content.

Although one Application Object Manager component can support only one language, multiple Application Object Managers can run at the same time on the same Siebel Server, each configured for a different language.

Additional Information

The following topics contain more information about language deployment for Siebel applications:

- *Scenarios for Installing and Deploying Siebel Languages*
- *About Parameters for Language and Locale*
- *Creating Language and Locale Records*
- *Siebel Installation Guide*
- For Siebel language support, Unicode support, and legacy code page support, see 1513102.1 (Article ID) on My Oracle Support

Locale

This topic is part of *Global Deployment Terminology*.

A *locale* is based on the language, country (territory), and character set applicable to a particular place or region. Siebel CRM cannot control the character set supported by the database and do not directly support the concept of a country, so the locale is primarily based on the language. Organization is sometimes used as a proxy for country.

Locales are defined in the Siebel seed data and associated with Application Object Manager components, as are languages. A Siebel locale includes a collection of user profile inputs, including keyboard layout and the formats used for numbers, dates, currencies, and times. Bidirectionality is also a function of locale, as noted in *Verifying Bidirectional Capability*.

The Siebel Web Client adopts the locale settings in effect for the Application Object Manager component on the Siebel Server.

The Siebel Mobile Web Client and Developer Web Client adopt the locale settings defined in the client operating system's regional settings.

For more information about locales defined in Siebel CRM, see *Siebel Applications Administration Guide* .

Types of Locales

Different types of locales are described as follows:

- **User locale.** The current language and country settings active for this session.

  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.

  Both the Siebel database and the Siebel applications have locale settings, which are independent of the operating system (except for the Siebel Mobile Web Client and Developer Web Client).

- **Input locale.** The current language used for entering data from the keyboard.

  The input locale affects the layout of keys on the keyboard, and for some languages, the way in which those key entries are then processed before the application enters the data into the current form on the screen. The input locale describes the language being entered and the input method, which could be a particular keyboard layout or a speech-to-text converter.

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

  An input method editor (IME) allows you to enter complex characters, such as those in Asian languages, directly from the keyboard. For information about setting the IME mode on applet controls and list columns, see *Configuring Siebel Business Applications* .

- **System locale.** If you are using a Microsoft Windows operating system, then the system locale is a systemwide setting that designates which code page is used as the default for all of the users on the system. If you are using a UNIX operating system, then the settings for formatting and code page locales are *not* systemwide.

These code pages and fonts allow non-Unicode applications to run as they would on a system localized to the language of the system locale.

For more information about specifying the system locale on UNIX, see *Siebel Installation Guide* .

> **Note:** If you are using a Windows operating system, then you *must* restart the system after changing a system locale.

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 February four, year two thousand-and-twenty-one* can appear differently depending on the locale. It might appear as:

- 02/04/2021, 04:30 PM, if the locale used is English American (ENU).
- 04.02.2021, 16:30, if the locale used is German (DEU).

Locales specify thousand separators and decimal symbols for numbers. They determine the position of the currency symbol in relation to the currency amount.

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.

Internationalization

This topic is part of *Global Deployment Terminology*.

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. Internationalization is often abbreviated as *I18N* (there are 18 characters between the initial I and the terminal N).

The software might have to input, display, and print characters, sort text, and recognize numbers and dates in different formats, and display and print text right-to-left as well as left-to right. Therefore, certain engineering features must be incorporated into the code to handle these requirements.

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

Internationalization Features

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 applications have 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 resource library files (such as DLL files) are language-independent, while other such files are language-dependent. In general, language-dependent files are located in language-specific installation directories.
- Euro (€) currency support
- 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

  > **Note:** Oracle does not support new Siebel CRM deployments on non-Unicode databases. All new Siebel CRM deployments must be Unicode.

  For a list of the languages supported by Siebel CRM, and the supported code pages for each database, see 1513102.1 (Article ID) on My Oracle Support.
- The ability to support both left-to-right and right-to-left displays, referred to as *bidirectionality*

Localization

This topic is part of *Global Deployment Terminology*.

Localization is the process of readying a product for use in a particular target country. Localization is often abbreviated as L10N, because there are 10 characters between the initial L and the terminal N. (The product must have been internationalized, or else most localization cannot be performed.)

Localization tasks are described in *Localizing Global Deployments*.

Siebel applications are localized as required by the customer base. Local language releases are translated and elements of the user interface, including buttons, error messages, and log files, are configured to meet local requirements.

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

Customers must perform certain tasks to complete localization. The necessary tasks might vary according to the language requirements. For example, implementing any language that displays using a right-to-left directionality, such as Arabic or Hebrew, requires a particular set of tasks.

General Activities for Localization

Localization consists of these general activities:

- **Translation.** Taking all of the applicable strings that appear on-screen in the application user interface and translating them into the language used in the target country.
- **Adaptation.** The process of making sure the product is suitable for use in the target country. Example activities are:

- Modifying the user interface to display language-specific elements, for example, hiding or displaying fields or modifying the position, height, and width of controls to accommodate the target language. For example, if a target country does not have a governmental equivalent to a state, then the State field might be hidden for the target country.
- Modifying images used in the application to those appropriate for the target country.
- Ensuring that the default configuration for the target country includes the right date format, currency, address format, salutations, names of provinces or states, and so on. User interface labels and master data might need to be modified.

  For example, a U.S.-specific term like *SSN* (Social Security number) is not translatable, but might be replaced with an equivalent term for the target country, such as *national ID number*.

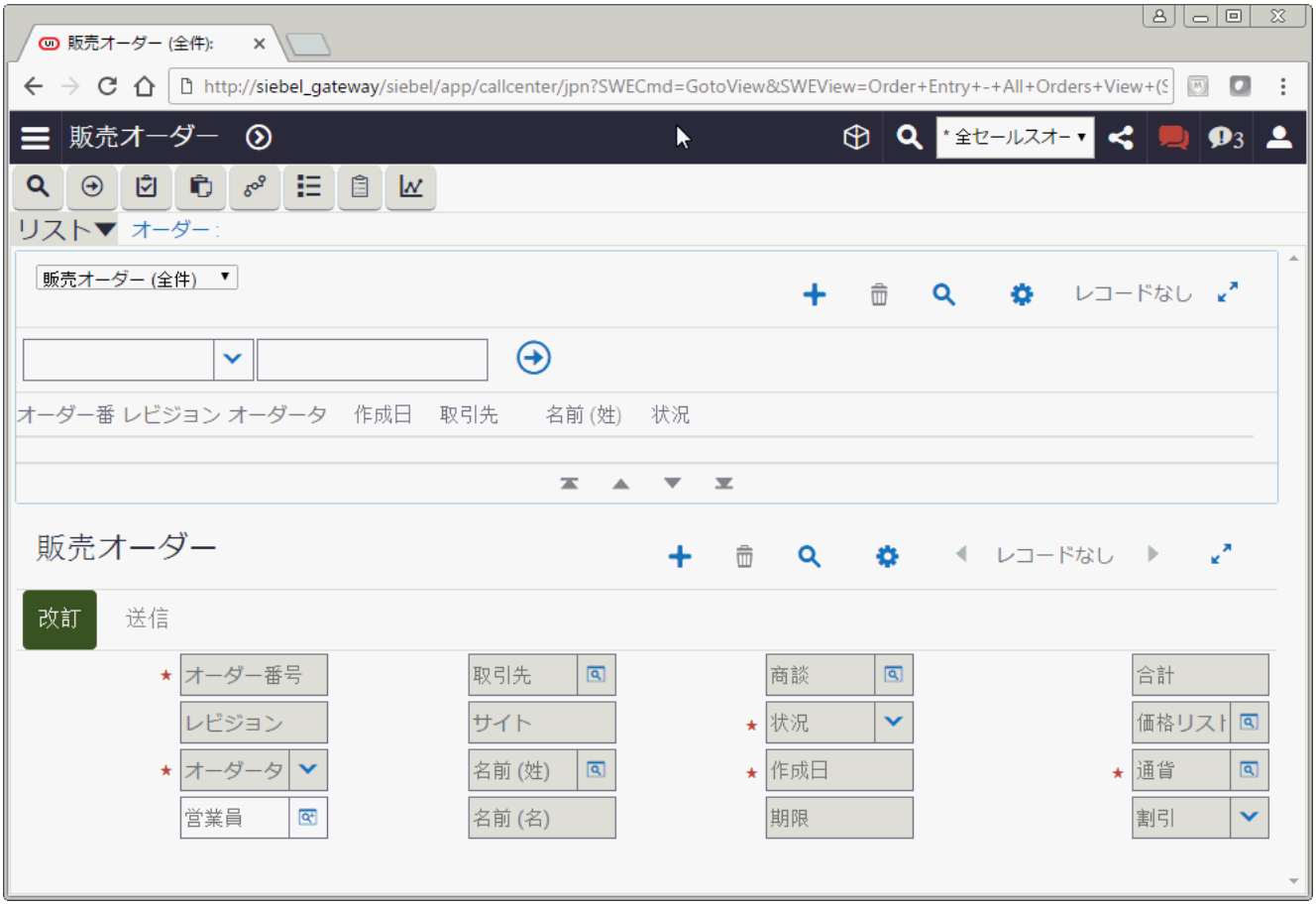
  For another example, the State field is prepopulated with the names of the U.S. states. These values are incorrect in other countries that have states (or equivalent), such as Mexico and Brazil. Where applicable, replace the LOV containing state names with the list of states (or equivalent) for the target country.

  Addresses use a single format for each language, and there are more than 400 address applets across the applications. For each supported language, Siebel CRM predefines the address formats for the target country.

  For example, the address format for France is used with the French (FRA) language. French-speaking users in Canada will find that this is the wrong address format, so you will likely want to change it. Similarly, the U.S. address format, used for the U.S. English (ENU) language, is incorrect for English-speaking users outside of the U.S.
- Changing from a left-to-right display to a right-to-left display. (The ability to support both left-to-right and right-to-left displays, referred to as *bidirectionality*, is an internationalization feature.)
- Defining and implementing access control mechanisms that are appropriate for the users in the target country and the data they work with. Data might need to be visible in multiple countries or visible only in particular countries.

Localization Example

A Siebel CRM localization example for Japanese (JPN) is shown in the following figure. In this example, the localization that was done is part of the standard product.



About Date, Time, Number, and Currency Formatting

Siebel CRM supports formatting of data such as dates and time, numbers, phone numbers, and currency, based on locale settings. More information about formatting for these types of data can be found in *Siebel Applications Administration Guide* .

The following are examples of different formats based on locales:

- Date and time
  - 11/4/2020 or 3/21/2021 (U.S. English format, for November 4, 2020 or March 21, 2021)
  - 04.11.2020 (German format, for 4 November, 2020)
- Number
  - 1,234.34 (U.S. format, with a comma as the 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)
- Currency
    - $32.45 (U.S. format, with U.S. dollar currency symbol in front of the amount)
    - 99.40 kr (Swedish format, with Krona currency symbol behind the amount)

Application handling of multicurrency transactions for multinational businesses includes automatic currency conversion, with full euro support. Siebel CRM allows you to conduct currency transactions using multiple currencies, and to define additional currencies as needed. Currencies are converted as needed within the application, such as when rolling up forecasts.

For information about administering currency conversion, see *Siebel Applications Administration Guide* . For information about configuring dual-currency display, see *Configuring Siebel Business Applications* .

About Supported Character Sets

This topic provides information about the non-Unicode (traditional) and Unicode character sets supported for Siebel CRM. It contains the following information:

- *Non-Unicode (Traditional) Character Sets*
- *Unicode Character Set*

In this guide, the terms character set and code page are used to cover closely related concepts used by the various platform vendors.

> **Note:** Oracle does not support new Siebel CRM deployments on non-Unicode databases. All new Siebel CRM deployments must be Unicode.

> **Note:** Siebel CRM does not support any character that has been added to a font by mapping it to an open code point that is not within an official character set extension area, such as the Private Use Area (PUA) of Unicode.

Non-Unicode (Traditional) Character Sets

This topic is part of *About Supported Character Sets*.

Before the emergence of Unicode, non-Unicode (traditional) character sets were available to address storage and processing requirements for a specific language or group of languages.

Examples of non-Unicode character sets are Code Page 1252 for languages spoken in Western European countries as well as in the Americas and elsewhere, and Code Page 932 for the Japanese language.

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

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

For example, the letter a-umlaut (ä) in the Western European character set does not exist in the Arabic character set. In a Western European code page, such as 1252 or ISO 8859-1, the a-umlaut occupies code point E4 (Hex value). In an Arabic code page, such as 1256 or ISO 8859-6, the E4 code point is an Arabic character and not the a-umlaut. Thus, you cannot represent the a-umlaut character on an Arabic system, or represent the Arabic character in a Western European system.

There is a set of characters that are common in most generally used non-Unicode character sets and code pages. These characters are known as the ASCII characters. They include the common characters used in the English language and they occupy the first 128 code points (Hex 00-9F) in the non-Unicode code pages. For a list of the languages supported by Siebel CRM, and the supported code pages for each database, see 1513102.1 (Article ID) on My Oracle Support.

> **Note:** It is the customer's responsibility to choose a supported character set that includes the characters required by the customer's business. Because the character set is a property of database configuration performed by the customer, Siebel CRM has no control over this setting. Choosing an inappropriate character set might require database reconfiguration later, and a corresponding need to convert large amounts of transaction data that has built up in the wrong character set. Converting transaction data is generally a time-consuming and costly experience. For help with character set conversion to Unicode, you must engage Oracle's Application Expert Services. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Unicode Character Set

This topic is part of *About Supported Character Sets*.

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 1990s. 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 64 thousand planes of 64 thousand characters each. Unicode character encoding treats alphabetic characters, ideographic characters, such as Kanji, and symbols identically, which means that they can be used in any mixture with equal facility.

The original Unicode standard (1.0) defined a 16-bit entity as the basic unit to represent a character. This standard became the basis of the UCS-2 encoding of Unicode, which specifies 16 bits for each character, regardless of which language it might represent.

However, the UCS-2 standard considered 8 consecutive bits of zero value to be valid data, which has a different meaning to programs written in C, where it means the *end of string*. Because most Web and communications software was written in C at the time the Unicode standard was introduced, an alternative encoding of Unicode called UTF-8 became popular. It encodes exactly the same set of characters, but avoids the *null byte* problem. To do this, it represents data in variable amounts: 1, 2, or 3 bytes in length, depending on the character.

Today the Unicode standard has advanced further, and has defined an extension mechanism to encode more than 16 bits worth of information. This revised standard is now referred to as UTF-16. The UTF-8 standard has remained popular among Web users, and has added a fourth byte in size to address the Unicode extension mechanism. Today there are two forms of Unicode in active use, UTF-16 and UTF-8, and Siebel CRM uses both of them.

For more information about Unicode, see the Web site of the Unicode Consortium:

```
http://www.unicode.org
```

For more information about databases supported by Siebel CRM, see the Certifications tab on My Oracle Support. For a list of the languages supported by Siebel CRM, and the supported code pages for each database, see 1513102.1 (Article ID) on My Oracle Support.

UCS-2

UCS-2 stands for Universal Character Set - 2 Bytes. In this standard, all characters are represented by two bytes (16 bits), no matter the origin.

UTF-8

UTF-8 stands for Unicode Transformation Format, 8-bit Encoding. 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 use one byte for each character, accented European characters use two bytes, and Asian languages use three bytes for each character.

UTF-16

UTF-16 replaces the original UCS-2. 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 is fairly limited, because the most common characters have already been included in the first 64,000 values.

About Character Set Encodings and Siebel Applications

Character set encodings are used in multiple places in Siebel CRM. See also *About Supported Character Sets*.

- **Enterprise DB Server Code Page system preference.** This system preference is set during Siebel Enterprise Server installation and configuration to reflect the character set that the administrator believes has been set up in the database. *This value must not be modified*, because it is used at configuration time to select the correct database schema to be used. (Siebel CRM provides customized schemas to match each database and character set.)

  For more information, see *Siebel Installation Guide* .
- **SIEBEL_CODEPAGE (UNIX environment variable).** This environment variable is created and set for Siebel CRM to indicate the code page that the applications will assume if Siebel configuration files (CFG files, CSS files, and so on) have *not* been saved as Unicode UTF-8, as they would normally be saved. This variable generally does not need to be set explicitly. If you must set it, then the value can be a subset of character set encodings, except UTF-8 and UTF-16.

  For more information, see *Siebel Installation Guide* .
- **Character conversion argument.** This argument is available in the following business services:
  - **Transcode Service business service.** Accepts all of the supported character set encoding names. This business service is normally used for data validation and to prevent data that cannot be converted to the

appropriate code page from entering or leaving the Siebel application. For more information, see *About the Transcode Service Business Service*.

> **Note:** Whenever possible, use EAI business services such as the XML Converter business service to convert data.

- **EAI business services.** These business services accept a variety of character set encodings.

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, then the character set name is supplied textually.

Updating Currency Symbols

In some situations, you might 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 non-Unicode environment, then you must update your currencies to include any currency symbols that you require that were not part of your prior non-Unicode environment.

For information about activating and defining currencies, see *Siebel Applications Administration Guide* .

About the Database Collation Sequence

The *collation sequence* for your database, also called *sort order*, is the ordering relationship, or sequence, between data records. Each database has a collation sequence so that records returned by queries can be returned in a certain order, such as an alphabetic order for text strings. The collation sequence determines the order in which records are displayed in the Siebel client, most noticeably in list views.

A collation sequence is defined when you set up the Siebel database. All of the sorting is done in the Siebel database by the database server. Sorting is not set or performed within the Siebel application and does not depend on the operating system.

> **Note:** For more information about creating and configuring the Siebel database, see *Siebel Installation Guide* . For more information about collation sequences for upgrade environments, see *Siebel Database Upgrade Guide* .

For the collation sequences supported for each supported RDBMS platform for the Siebel database, see the Certifications tab on My Oracle Support. For a list of the languages supported by Siebel CRM, and the supported code pages for each database, see 1513102.1 (Article ID) on My Oracle Support. Also consult your RDBMS vendor documentation.

The collation sequence in effect for a database is determined by one of the following implementation methods:

- Indexes created in the Siebel database provide a default collation sequence. In Oracle Database, indexes always use binary collation sequence.
- Post-query sorting might also be supported for an RDBMS platform. However, this method of sorting yields slower performance and requires all of the records to have been retrieved first. For this reason, it is impractical for Siebel applications, which always perform open-ended queries.

For the development environment, only binary collation sequence is supported. For a production environment, you can specify the collation sequence most suitable for your deployment.

Note: Changing the collation sequence after the Siebel database has been installed requires rebuilding your indexes. On a fully loaded production database, this task is time-consuming and database resource-intensive. For help with planning a project of this complexity, it is recommended to engage Oracle's Application Expert Services. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Which collation sequence is best for your deployment depends on factors such as RDBMS support, performance requirements, database availability requirements, the code page in use, the needs of your users, and the nature of the data that is to be retrieved by different groups of users.

Binary collation sequence offers the best performance and does not require you to rebuild your indexes for the production environment. This collation sequence works well for users working with English-language data, because the ASCII character set is based on the English alphabet and corresponds to the binary collation sequence. However, sorting might be unsuitable for users and data in languages other than English.

For multilingual deployments using Unicode, a linguistic collation sequence based on the Unicode Collation Algorithm (UCA), which goes by different names for different RDBMS vendors, might be a suitable collation sequence. UCA, also known as ISO 14651, provides reasonably good results with mixed-language data.

Other linguistic, or dictionary, collation sequences might offer optimal sorting results for particular languages or groups of languages. Such collation sequences might be suitable for certain deployments, such as those requiring compatibility with the CP932 (Japanese Shift-JIS) sort order.

Linguistic collation sequences that are not based on UCA might associate multiple characters (such as accented and unaccented versions of a particular letter) so they will be treated the same for sorting purposes, but will also be treated the same in unique indexes. If you are changing to a case- or accent-insensitive collation sequence, then you will need to first clean out any data that is unique only due to a case or accent difference.

Database Collation for the Local Database

Database collation for the Siebel Mobile Web Client, for which the local database uses Oracle Database SE2, is the same as for Oracle Database Enterprise Edition.

Note: As of Siebel CRM 20.8 Update, Oracle Database SE2 replaces Oracle Database XE for the local database for Siebel Mobile Web Client.

3 Planning Global Deployments

Planning Global Deployments

This chapter provides high-level guidelines in how to successfully plan a global deployment of Siebel CRM. It includes the following topics:

- *Understanding Your Company's Global Business Requirements*
- *Expected Results of Global Deployment Planning*
- *About Ideal Global Deployments*
- *About the Siebel Unicode Architecture*
- *About Planning Your Global Deployment*
- *About Configuring Global Deployments*
- *Scenarios for Installing and Deploying Siebel Languages*
- *Planning Upgrades for Global Deployments*

Understanding Your Company's Global Business Requirements

Imagine that your company tells you that it wants to market the software it develops to four other countries and wants the applications to run in the languages of those countries. How do you start? What do you need to consider to make sure that the new product development effort is successful? What do you need to think about when customizing Siebel CRM for this purpose?

Although you must take your company's unique business requirements into consideration in your planning, this guide offers tips and guidelines for undertaking a global product rollout and maintenance in general, as well as configuring the Siebel CRM applications in particular.

By addressing the following questions concerning your organization's global business needs, you will have gone a long way towards planning for your global deployment of Siebel CRM:

- Will your company have one central business location from which all of the business transactions originate, or regional decentralized (distributed) locations for transactions?
  - If your enterprise is decentralized, then does your company need to keep the transactions synchronized, for example, banking transactions?
  - Will remote users synchronize their transactions to a central corporate database or a regional database?
- Are computer telephony integration (CTI) servers, email servers, Oracle Business Intelligence servers, and other servers centralized or regional?
- Which languages does your company headquarters require?
- Which language will be the base, or primary, language with which you begin your development and customization process? The first language installed in the Siebel database is the primary language.

- Have you previously customized any language files from a previous version of Siebel CRM, or configured a new language not yet offered for Siebel CRM?
  Languages previously provided for Siebel CRM can be upgraded. If you previously localized into a language that Siebel CRM now provides directly, then you must either merge your previous use of this language with the Siebel language, or keep them separate.
- Which locales are the languages intended for (for example, French-speaking Canada or France)?
  - Which locale settings will be needed as a result? The answer to this will affect the way currency, numbers, dates, and times are formatted in the software.
  - Because additional Siebel Application Object Managers will be needed for each locale and language combination, determine in advance the implications for memory and performance of the products that you use.
- What character set will you use for your database: a Unicode or a non-Unicode character set?
  This decision has far-reaching implications for the ease with which your organization can deploy globally. If you do not implement a Unicode database, then you might not be able to support all of the languages that your business uses. In that case, you will not be able to roll up data from those countries into your Siebel applications, or into a data warehouse.
- Are there legacy interfaces that you need to consider in your planning and do these have implications for your back-office applications? In which code page is the data of your back-office applications expected? Is there a need to convert between code pages, such as from Unicode to non-Unicode?
- Who will localize your customizations?
- Are there particular legal requirements that you must meet within your global network (for example, as regards European Union Data Protection Directives, Basel II, or others)?

Expected Results of Global Deployment Planning

The answers to the questions in *Understanding Your Company's Global Business Requirements* will help determine your:

- Globalization strategy
- Globalization project timeline
- Network diagram
- Capacity planning
- Call handling strategy for CTI and call center features
- Localization scope, for example, for the List of Views and Applets by language
- Localization budget
- List of modifications that you need for preconfigured functions required by a specific locale
- List of tables and data to be exchanged with other applications, as determined from your analysis of any legacy interfaces

By considering the questions listed in *Understanding Your Company's Global Business Requirements* before you undertake a global deployment, you stand to decrease the total cost of ownership (TCO) your company pays for its global outreach, in the form of:

- Driving down the cost of localization development
- Decreasing global deployment costs in general

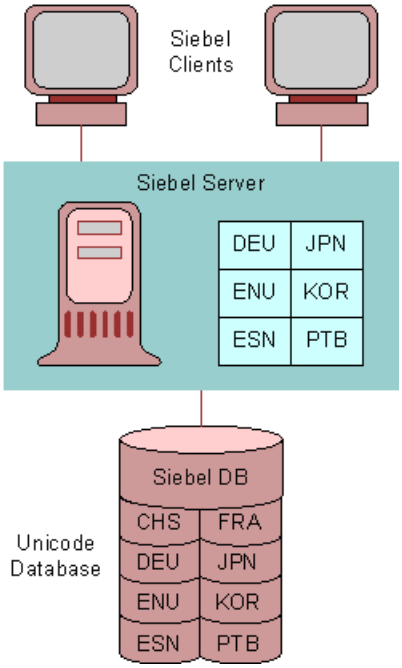
- Shortening the time to market for global deployments

About Ideal Global Deployments

The following figure provides an example of what an ideal global deployment of Siebel CRM might look like. Keep in mind that few companies necessarily meet an ideal. However, with proper planning, the ideal is a goal that can be achieved.

In this example, six Siebel languages have been installed on the Siebel Server. Different Application Object Managers running on this Siebel Server can display the application user interface in these languages.

The Siebel database in this example uses Unicode and therefore supports all of the characters required for the six languages installed on the example Siebel Server. The other two languages shown, Chinese (CHS) and French (FRA), must be supported for customer data in this example deployment, even though these two languages are not used for the Siebel application user interface.



The preceding figure illustrates an ideal centralized global deployment of Siebel applications. Unicode is the character set of the database in this ideal configuration.

For a production environment, you can specify the collation sequence most suitable for your deployment. For the development environment, only binary collation sequence is supported. For more information, see *About the Database Collation Sequence*.

Where necessary, character set conversions take place to and from all of the Siebel (Unicode) data sources and other data sources.

Siebel clients run in this deployment on localized operating systems with Unicode fonts installed for selected users. The user interface appears in the language of the user's choice. Even languages such as Japanese (JPN) and Arabic (ARA) are encoded properly.

For more information about installing languages, see *Siebel Installation Guide* .

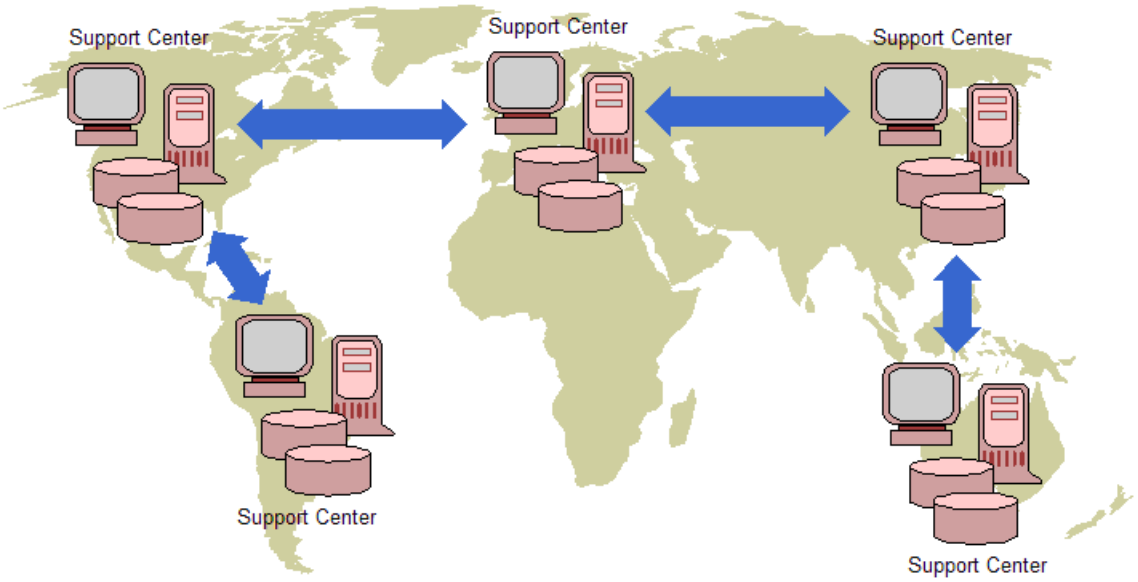
It is strongly recommended that you enable Universal Time Coordinated (UTC) for your global deployment. For more information, see *Deploying Siebel CRM with Global Time Zone*.

Also enable multilingual lists of values (MLOVs) for your global deployment. For more information about configuring LOVs and MLOVs, see *Localizing Lists of Values and Multilingual Lists of Values*.

The ideal global configuration allows deployment of Siebel CRM to satisfy the requirements for either a centralized or a decentralized (distributed) global enterprise.

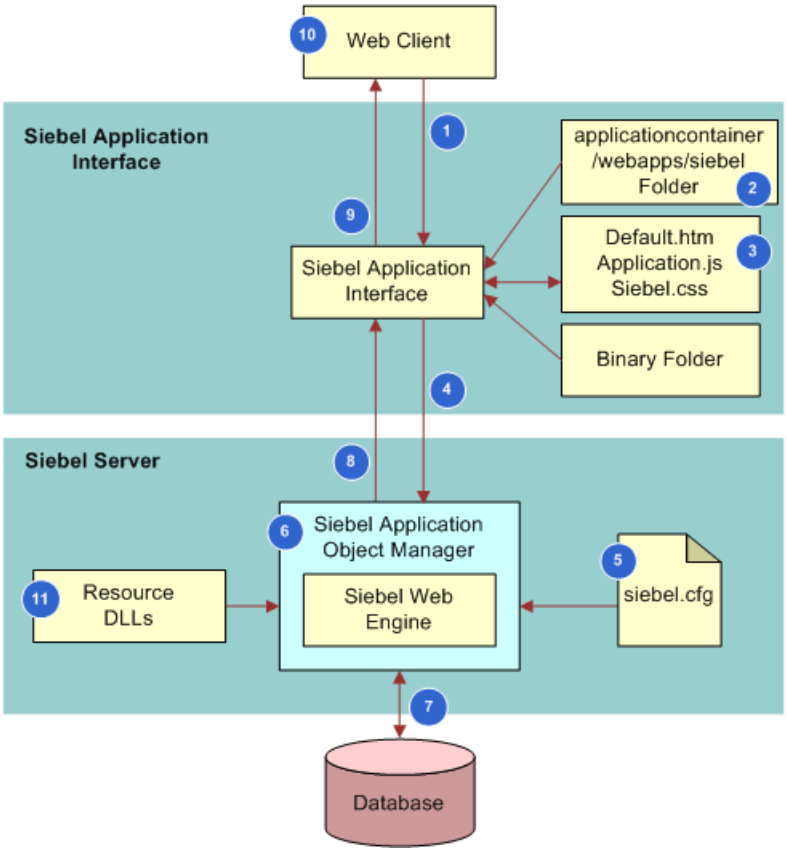
A centralized global deployment might require only a single Siebel CRM installation, and consolidates all of the customer-related information in one data store, creating a single, global view of customers transactions. A centralized deployment is easier and cheaper to maintain and might be satisfactory, depending on your business needs.

A decentralized (distributed) global deployment might have Siebel CRM instances in the Americas, Europe, and Asia, as shown in the following figure. Data is replicated between these sites using Siebel Replication Manager. For more information, see *Siebel Deployment Planning Guide* , *Siebel Remote and Replication Manager Administration Guide* , and other applicable documentation.



About the Siebel Unicode Architecture

The Siebel CRM software architecture uses Unicode, including for internal processing. There are many benefits to using Unicode, such as the ability to show multilingual data in the same session. The following figure provides a graphical representation of the architecture, while the table that follows describes the flow of data through the Siebel Unicode architecture.



The following table describes the flow of data through the Siebel Unicode architecture. For more information about supported databases, see the Certifications tab on My Oracle Support.

Step Number	Text Encoding
1	User input is URL-encoded by JavaScript or the browser.
2	JavaScript code and style sheets are in ASCII. **Note:** Style sheet files can be saved as UTF-8 in order to be able to specify font names that contain non-ASCII characters.
3	HTML pages are in ASCII, or UTF-8 if hard-coded non-ASCII content is required.
4	The Application Interface sends input (still in URL encoding) to the Siebel Server. The Siebel Server decodes input immediately to the Siebel internal encoding (UTF-16).
5	The application configuration file (CFG) is UTF-8. (Some parameter values might specify non-ASCII text for elements such as font names or splash screen text.)
6	Most of the C++ code uses the Siebel internal encoding (UTF-16).

Step Number	Text Encoding
7	The Application Object Manager uses UTF-16 to communicate with the database. The internal encoding of the database can be UTF-8, UTF-16, or a code page. The database includes the Siebel runtime repository and the Web templates.
8	The Siebel Web Engine (SWE), which is part of the Application Object Manager, sends the HTML page (in UTF-16) to the Application Interface.
9	The Application Interface transcodes, or converts, the HTML page to UTF-8 for the World Wide Web and passes the HTML page (in UTF-8) to the browser.
10	The browser code (JavaScript) reads the HTML page in UTF-16.
11	The resource libraries (such as DLL files) store content in UTF-16.

About Planning Your Global Deployment

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

1. Determine what your base application language will be, for example, U.S. English (ENU).
2. Consider the available industry-specific products and the operating systems that you use.
3. Install and configure the Siebel CRM software for a development environment. For details, see *Siebel Installation Guide* , and see the last step in this task (which is for a production environment). For more information about the development environment, see *Using Siebel Tools* .

   > **Note:** The specific installation and configuration tasks that you must perform, as well as the sequence that you perform them in, vary depending on your installation case. For details, see *Siebel Installation Guide* . In general, the information presented here assumes the new installation case.

4. Determine which application elements need to be localized. For example:
   - Menus
   - Picklists
   - Lists of values
   - View names
   - Strings in applets
   - Correspondence templates
   - Communications templates (for example, for Siebel Email Response, Send Email, or Send Fax)
   - Forecasts
   - Personalization rules
   - Workflow policies
   - Assignment rules

- Currency
- Master data (for example, product and catalog data)
- iHelp

  For more information about creating and administering iHelp, see *Siebel Applications Administration Guide* .

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

   You might determine 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 CRM applications' performance for columns used in search specifications.
6. Configure your Siebel CRM software based on your company's business requirements. For details on application configuration tasks, see *Configuring Siebel Business Applications* and other applicable documentation.
7. Test and validate your localized applications. Correct any issues before deployment.
8. For a production environment, deploy the Siebel CRM applications to a global user base. For details, see *Siebel Installation Guide* , *Siebel Database Upgrade Guide* , and other books on the *Siebel Bookshelf* . Do the following:

   - Install the Siebel Gateway, Siebel Server, Siebel Application Interface, and other modules, with the Siebel languages that you require, for your production system. The Siebel languages include the language-specific run-time environment: resource libraries such as DLL files, configuration (CFG) files, error messages, help files, and so on. (You will also deploy files that you updated during the application configuration process.) For more information about this task, see *Siebel Installation Guide* .
   - Install the Siebel database for the production environment. This task installs seed data and repository data for the primary language into the Siebel database. For more information about this task, see *Siebel Installation Guide* . For more information about the production environment, see *Using Siebel Tools* .
   - Install language-specific seed data for all of the other deployed languages into the Siebel database.
   - Install language-specific repository data for all of the other deployed languages into the Siebel database.
   - Using Siebel Management Console, configure the Siebel Gateway, Siebel Enterprise, Siebel Server, Siebel Application Interface, and other modules. For more information about this task, see *Siebel Installation Guide* .
   - Install the Siebel Mobile Web Client and deploy it in the local language by including the necessary Siebel languages.

     For more information, see *Siebel Installation Guide* , *Siebel Remote and Replication Manager Administration Guide* , and *Siebel Anywhere Administration Guide* .

Note: Many tasks for multilingual deployments are more complex and time-consuming than they are for a single-language deployment. Such tasks include installing applications, localizing custom configurations, administering data in multiple languages (MLOV and master data), testing localized applications in each deployed language, training users, and so on. Depending on your requirements, additional hardware might be needed to support deploying multilingual applications globally.

About Configuring Global Deployments

By answering the following questions, you will be underway to planning for configuration of your global network applications. For detailed information about configuration tasks, see *Configuring Global Deployments*.

- Which address applet layouts do you want to modify for your target countries?
- Which Name layouts do you want to modify for your target country and languages?
- Do you want to add *preferred language* to contact records? The preferred language is the language in which a customer prefers to receive communications such as email messages, brochures, and so on.
- Do you want to separate prospect lists by country and language?
- Do you want to add a language code or country code to attachment names? Doing so can help you to identify attachments that are intended only for an audience using a particular language or page size, for example.
- Do you want to extend product masters, training course descriptions, and other master data to have local language versions? Doing so can help you provide such information to the right audience.
- Do you want to mark selected data records with a country or language tag? Doing so can help you roll up sales records originating in a particular country or region, for example.
- Have you converted LOVs to MLOVs (multilingual LOVs), where appropriate? Have you updated LOVs or MLOVs with international content, for example, lists of countries, offices, and other data? Have you added organizations by country or region? Organizations can be defined as countries. See also *Localizing Lists of Values and Multilingual Lists of Values*.
- Have you created localized response templates for email?
- Have you created multilingual PDQs? This term refers to PDQs that are constrained by language or by country, in order for the query names to display using the appropriate language.
- Have you modified custom search criteria to match non-English records?
- Have you created localized seed and demo data?
- Have you modified workflows to perform LOV value lookup?

By addressing the foregoing questions, you can achieve the following outcomes:

- Address and name applets are formatted to your target countries.
- LOVs or MLOVs, and PDQs are configured.
- Email servers and other third-party servers are configured.
- Seed and demo data are localized.
- Workflows are modified for language and locale.
- Global data, SmartScripts, iHelp, and so on are localized.

Scenarios for Installing and Deploying Siebel Languages

This topic summarizes deployment scenarios when you plan to deploy multiple Siebel languages for your Siebel CRM software. It is strongly recommended to include all of the languages that you might require when you first install the Siebel CRM software.

> **Note:** For detailed information about these scenarios, including prerequisites, limitations, installation and configuration instructions, and associated tasks performed for the Siebel database, see *Siebel Installation Guide* .

Consider the following scenarios for installing and deploying multiple Siebel languages:

- **Install and deploy all of the languages you require once.** For customers installing and deploying languages shipped with the current version of Siebel CRM, and who *do not* plan to deploy other languages at a later date. This option is easiest but might not be suitable for phased language deployments.
- **Install all of the languages you will require, but deploy some languages later.** For customers installing and deploying languages shipped with the current version of Siebel CRM, but who plan to deploy some of the installed languages at a later date. This option might be suitable for phased language deployments.
- **Install and deploy additional languages in an existing deployment.** For customers installing and deploying languages shipped with the current version of Siebel CRM, and who plan to install and deploy additional languages at a later date. This option might be suitable for some phased language deployments.

Planning Upgrades for Global Deployments

This topic describes considerations for planning a successful upgrade of Siebel CRM within a global deployment.

For detailed information about the upgrade process, see *Siebel Database Upgrade Guide* . See also *Siebel Installation Guide* .

Planning for the Upgrade

Addressing the questions in the following list will help you prepare for a successful upgrade of Siebel CRM within a global deployment.

- What database version do you use?

  Where applicable, you must upgrade your RDBMS to a supported version for the current version of Siebel CRM. If you are moving to Unicode, then the new database version might also make this possible.
- What version of Siebel CRM will you be upgrading from? Current versions of Siebel CRM support Unicode.
- Do you know what is involved in migrating your data to Unicode?
- Did you previously deploy a customized language not certified for Siebel CRM? Such languages are also referred to as *unshipped languages*. See also *Localizing an Unshipped Language*.

- Do you know how to upgrade date-time fields to Universal Time Coordinated (UTC)? See also *Deploying Siebel CRM with Global Time Zone*.
- Are the Siebel languages that you require released and available?
- Have you upgraded client computers (including keyboards, fonts, locales installed, and localized operating system) to support the new installed languages?
- Have you evaluated the need to upgrade all of the third-party products to versions that support Unicode, the countries or locales that you will be localizing for, and the languages that you require?

Results to Expect After Upgrading

After you have addressed the previously described issues and performed the necessary upgrade tasks, you will have gone through the following conversion processes:

1. Updating databases and Siebel versions to current versions.
2. Converting data to Unicode or to another code page that is supported with newer versions of Siebel CRM.

   > **Note:** Conversion to Unicode requires the assistance of Oracle's Application Expert Services. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

3. Converting selected LOVs to MLOVs.
4. Converting selected date-time data to UTC.

4 Configuring Global Deployments

Configuring Global Deployments

This chapter discusses how to configure your Siebel CRM applications for a global deployment. It includes the following topics:

- *About Parameters for Language and Locale*
- *About Configuring Language Support for Browser Platforms*
- *About Unicode Fonts*
- *About Integration Considerations*
- *About Application-Wide Data*
- *Setting Up Global Data*
- *Configuring the Siebel Calendar*
- *Configuring Cascading Style Sheets to Specify Different Fonts*
- *Verifying Bidirectional Capability*
- *Configuring Case Insensitivity*

About Parameters for Language and Locale

This topic describes the parameters that can be set on the Application Object Manager in order to specify language, resource language, and locale. On the Siebel Mobile Web Client, equivalent parameters are defined in application configuration files (such as uagent.cfg for Siebel Call Center).

The active language for Siebel CRM is the language in effect for an individual user's session and the language of user interface elements, including multilingual lists of values (MLOVs) that have been enabled. The active language is determined by the setting of the Language Code parameter for the Application Object Manager component to which a Siebel Web Client is connected. The same language is also used for logging and other system messages, unless the resource language is also defined. See also *Language*.

This topic contains the following information:

- *Language Code Parameter*
- *OM - Resource Language Code Parameter*
- *Locale Code Parameter*
- *Language Settings for Siebel Mobile Web Client*

Note: If you are localizing an unshipped language, then see also *Localizing an Unshipped Language*. In particular, for scenarios for setting the language parameters, see *Creating Language and Locale Records* and *Creating Application Object Manager Components*.

Language Code Parameter

This topic is part of *About Parameters for Language and Locale*.

The Language Code parameter (alias Lang) determines the language used for multilingual lists of values (MLOVs) and other application seed data. This parameter also determines which language-specific directory from the product installation will be used for accessing the Siebel runtime repository. The language of application user interface labels is determined by the language for the Siebel runtime repository. This language was determined by the Siebel Tools or Siebel Web Tools language mode in effect when the Siebel runtime repository was published.

For each language that you install and deploy, Application Object Manager components are created which are already configured with Language Code set to this language. In general, you do not need to change the value of the Language Code parameter for these components.

Special requirements apply when you localize an unshipped language, as noted in *Localizing an Unshipped Language*.

Unless the OM - Resource Language Code parameter is set to a different language, the Language Code parameter determines which language-specific directories will be used during runtime operation for resource libraries such as DLL files. These resource libraries determine the language in which system and error messages are displayed and the language in which server log file messages are written.

Each language code used by Siebel CRM uses a three-letter code, such as ENU for U.S. English, FRA for French, THA for Thai, and so on. For more information about the language codes used by Siebel CRM, see *Language*.

> **Note:** On the Siebel Mobile Web Client, the equivalent parameter is the Language parameter in the [Siebel] section of the configuration file. For each language that you install, language-specific directories are created containing the configuration files, in which Language is already set. In general, you do not need to change the value of the Language parameter. Special requirements apply when you localize an unshipped language. For details, see *Localizing an Unshipped Language*.

The setting of the Preferred Language field for a user or employee record overrides the value of the Language Code parameter for determining the language for multilingual lists of values (MLOVs). This setting is stored in the PREF_LANG_ID column in the S_CONTACT table in the Siebel database. This field does not affect Siebel Mobile Web Client or Developer Web Client users.

See also *Localizing Lists of Values and Multilingual Lists of Values*.

See also *OM - Resource Language Code Parameter*.

For more information about the effect of the language mode in Siebel Tools or Siebel Web Tools and about how to set it, see *Using Siebel Tools* . See also *Language*.

OM - Resource Language Code Parameter

This topic is part of *About Parameters for Language and Locale*.

The OM - Resource Language Code parameter (alias ResourceLanguage) can optionally be set to a different language than the Language Code parameter in order to specify the resource language.

The resource language determines which language-specific directories from the product installation will be used during runtime operation for resource libraries such as DLL files. These resource libraries determine the language in which system and error messages are displayed and the language in which server log file messages are written.

Resource library files are provided for all of the languages shipped with Siebel CRM. These files cannot be localized into any other unshipped languages.

For some multilingual deployments, it might be useful, or necessary, to set the resource language to a different value than the Language Code parameter. For example, you might want all of the global users to view MLOV and other seed data in the language determined by Language Code. However, you might choose to set the resource language to a single language familiar to your administrators.

Where Language Code is set to an unshipped language, you must use the resource libraries for a language provided for Siebel CRM. In this case, you can set OM - Resource Language Code to specify which language's resource libraries to use. For example, where Language Code is set to the unshipped language NOR (Norwegian), you might set OM - Resource Language to DEU (German). For details, see *Localizing an Unshipped Language*.

> **Note:** On the Siebel Mobile Web Client, the equivalent parameter is the ResourceLanguage parameter in the [Siebel] section of the configuration file.

See also *Language Code Parameter*.

Locale Code Parameter

This topic is part of *About Parameters for Language and Locale*.

The Locale Code parameter (alias LocaleCode) specifies the locale associated with this Application Object Manager component. This setting is used by the Siebel Web Clients for this Application Object Manager.

The Locale Code setting is a three-letter locale code. A locale is a set of rules guiding how common data is displayed to the user or is received from the user. Siebel CRM supports formatting of data, such as dates, time, numbers, and currency, based on locale settings.

Locales are administered using the Locale Administration view. For each language-specific Application Object Manager component on a Siebel Server, the Locale Code is set to a locale that might be appropriate for users for that language.

If you need to support multiple locales for the same language, then you can use either of the following approaches to deploy your Application Object Managers:

- *On different servers*, configure and run different locale-specific Application Object Managers for this language on different Siebel Servers. For example, an FRA Application Object Manager would be running in France with a French locale, while another FRA Application Object Manager would be running in Canada with a French Canadian locale.
- *On the same server*, create, configure and run different Application Object Managers that are specific to both this language and a particular locale. For example, a FRA Application Object Manager running in France would have a French locale, but you could create another FRA Application Object Manager on the same Siebel Server in France that has a French Canadian locale. For more information about creating and configuring Application Object Managers, see *Siebel System Administration Guide* .

If the Locale Code parameter is not set, then the Application Object Manager and the Siebel Web Clients use the locale defined in the operating system on the computer or operating system instance where you installed Siebel Server. On

Microsoft Windows, this setting is defined in Regional and Language Options in the Control Panel. On UNIX, you set locales as described in *Siebel Installation Guide* .

On the Siebel Mobile Web Client or Developer Web Client on Microsoft Windows, the locale is determined by the settings defined in Regional and Language Options in the Control Panel.

Additional Information About 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. The following table shows where you can find detailed information about performing these procedures.

For information about ...	See ...
Administering Siebel Server locales	*Siebel Applications Administration Guide*
Setting Siebel Server locales	*Siebel Installation Guide*
Setting Siebel Mobile Web Client and Developer Web Client locales	On a Windows PC, locales are administered through settings in Regional and Language Options in the Control Panel

Language Settings for Siebel Mobile Web Client

This topic is part of *About Parameters for Language and Locale*.

The active language for a user session in the Siebel Mobile Web Client is determined by the Language parameter setting in the application 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, language-specific directories for the current version might include:

- `C:\Siebel\Client\bin\deu\uagent.cfg` (configuration file for Siebel Call Center with Language parameter set to DEU, for German)
- `C:\Siebel\Client\bin\enu\uagent.cfg` (configuration file for Siebel Call Center with Language parameter set to ENU, for U.S. English)

Do not modify the Language parameter to change the active language. Instead, the user specifies the appropriate application configuration file for the language in which to run, by using the `/c` switch in the command line of the siebel.exe application shortcut.

For example, for running Siebel Call Center in German for the current version, the Siebel application shortcut created during installation with the DEU language has a property similar to this:

```
/c C:\Siebel\Client\bin\DEU\uagent.cfg
```

Related Books

Siebel Installation Guide

Siebel Remote and Replication Manager Administration Guide

About Configuring Language Support for Browser Platforms

A browser platform can include the following:

- The Web browser itself
- The operating system where the Web browser runs

For the browser standards supported by Siebel Open UI, for particular browser versions that are certified as supporting those standards, and for support information for Siebel CRM in general, see the Certifications tab on My Oracle Support.

For more information about requirements for the Siebel Open UI client, see *Deploying Siebel Open UI* . See also *Siebel Installation Guide* and other applicable documentation.

The client computer might not include all of the fonts and other resources necessary to display all of the Unicode characters that are outside the default non-Unicode code page. Some manual setup might be required in order to view all of the characters correctly in Web pages, including Siebel application screens.

If characters from some languages appear as hollow or solid boxes, then it might be due to lack of language setup in the browser. Box characters always indicate a problem at the user interface level. Question marks appear when Unicode characters are mishandled on the server or in third-party applications. They can also be due to the character set having been incorrectly configured in the database.

Vendor documentation and support should be definitive on these issues, but the following list provides a starting point for resolving issues. Web browsers display text in several contexts, including:

- HTML body text (most of the text in Siebel client screens)
- HTML field input (all of the text input in Siebel client screens)
- ToolTips
- Message boxes (alerts)

Related Books

Deploying Siebel Open UI

Configuring Siebel Open UI

Siebel Installation Guide

Siebel System Administration Guide

About Unicode Fonts

This topic is part of *About Configuring Language Support for Browser Platforms*.

This guide sometimes mentions *Unicode fonts*. This term refers to a font that contains glyphs for most, if not all, of the possible characters in Unicode. Such fonts might not be shipped as standard parts of the operating system. Customers must license them or otherwise obtain them for installation on their client computers.

A commonly used font of this type is Arial Unicode MS, which comes with every installation of Microsoft Office. The version of this font provided with Microsoft Office 2003 and later is compatible with Unicode 2.1, though it does not contain glyphs for all of the Unicode characters.

See also *Configuring Cascading Style Sheets to Specify Different Fonts*.

About Integration Considerations

There are many issues to consider when planning application integration for a global deployment of Siebel CRM. This topic contains the following information:

- *About Character Conversion Errors*
- *About the Transcode Service Business Service*

For detailed information about application integration for global deployments, see *Overview: Siebel Enterprise Application Integration* , documentation for Siebel Enterprise Application Integration (Siebel EAI), and documentation for specific integration products.

For a list of the languages supported by Siebel CRM, and the supported code pages for each database, see 1513102.1 (Article ID) on My Oracle Support.

About Character Conversion Errors

This topic is part of *About Integration Considerations*.

When the Siebel application sends data to an external application using a non-Unicode (traditional) character set encoding, the external application might not be able to accept all of the data if the data does not belong to the character set encoding of the application. In this situation, unrepresentable characters will be converted to a substitution character.

The substitution character might be a question mark or an upside-down question mark (indicating character loss in the database), depending on the database. A conversion result like this is referred to as *unrepresentable character conversion*.

The topic *About the Transcode Service Business Service* describes how to avoid this situation, using techniques for character conversion error handling.

The transport and adapter 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.

A second case occurs when you are using a Unicode communication stream, such as XML, to communicate between Siebel CRM and an external, non-Unicode application. In this situation, the character conversion can be handled by the external application after it receives the message. This situation presents the risk that conversion errors might occur and that the external application might not provide the desired error handling functionality.

To avoid problems associated with relying on third-party applications to convert data, you can use the Validate method of the Transcode Service business service, before sending any data to the external application. This method detects

whether character conversion performed by the external application will fail. If it will fail, then you can choose not to send the data. For details, see *About the Transcode Service Business Service*.

About the Transcode Service Business Service

This topic is part of *About Integration Considerations*.

To supplement the character conversion performed by Siebel EAI components, Siebel CRM offers the Transcode Service business service. You can use methods of this business service in your workflow processes to validate and convert data from one character set encoding to another.

For information about how business services handle character conversion errors, see *About Character Conversion Errors*. For the language-independent code values that you can specify in business service method arguments for the source or target encoding, see 1513102.1 (Article ID) on My Oracle Support.

The Transcode Service business service supports two methods:

- **Validate.** This method confirms the input property set hierarchy or the *value* of the input property set. You can use this method to check that a character is valid within a particular character set before proceeding, without actually performing the conversion. The arguments for the Validate method are described in the following table.

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	Specify the language-independent code for the target encoding.
SourceEncoding	No	Specify the language-independent code for the source encoding. This argument is required when ValidationMode is set to *value* and the input value contains binary data. Conversion from binary data in SourceEncoding to binary data in TargetEncoding is implied.

- **Convert.** This method converts the *value* in the input property set to the target encoding in the output. You use this method when data is entering or leaving Siebel applications, and a conversion is required now, so that the next software component in the processing chain can recognize the data. The arguments for the Convert method are described in the following table.

> **Note:** To ignore character conversion errors (that is, invalid character errors or substitution errors) for the Convert method, specify IgnoreConversionErrors = True as a method argument. Otherwise, do not use this argument. (This argument is hidden in Siebel Tools or Siebel Web Tools.)

Method Argument	Required	Description
ConversionMode	Yes	Values can be StringToEncoding, EncodingToString, or EncodingToEncoding.

Method Argument	Required	Description
TargetEncoding	Yes	Specify the language-independent code for the target encoding. Required for StringToEncoding and EncodingToEncoding modes.
SourceEncoding	No	Specify the language-independent code for the source encoding. Required for EncodingToString and EncodingToEncoding modes.

Note: The Convert method converts the actual encoding of a file, including XML files. However, this method does not change the encoding attribute in the file's XML header. For example, if you change an XML file's encoding from UTF-8 to UTF-16, then the XML header still refers to UTF-8, such as in this example: `<?xml version="1.0" encoding="UTF-8"?>`. To make the encoding attribute match the XML file's actual encoding, edit the file in a separate step.

About Application-Wide Data

This topic describes some of the types of application-wide data that need to be considered for global deployments of Siebel CRM. Many of these types of data provide multilingual support in the standard product; additional examples include product catalogs, task-based UI, and time zone translations.

Sales Cycles

Sales cycle stages can be localized into different languages by configuring them according the global data approach described in *Setting Up Global Data*. For more information, see 477104.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 438.

Currency

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

Telephone Numbers

You will need to specify telephone number formats for the countries in which you will be deploying the Siebel CRM applications. For information about how to perform this procedure, see *Siebel Applications Administration Guide* .

Templates

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

- Proposals
- Presentations
- Email
- Fax
- Correspondence

For more information about templates for email or fax, see *Siebel Email Administration Guide* . Each correspondence template is in a specific language. These templates have been localized. For information about creating new correspondence templates, in any language, see *Siebel Correspondence, Proposals, and Presentations Guide* . Each user can specify a default correspondence template; users can choose the appropriate template for their locale.

> **Note:** You can have the language tagged in the template in the appropriate language or locale field. Because the Application Object Manager is multilingual, you can write a single workflow rule, 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 about adding items to the Literature tab, see *Siebel Applications Administration Guide* .

> **Note:** In order to be able to tell the language in which a literature item is written, you might choose to include the item's language in its name and description, or add extension columns to hold those properties, so that they are searchable and could be automatically assigned by Assignment Manager.

iHelp

iHelp is a language-independent feature, which means that:

- The same iHelp can run in multiple languages.
- The same logic can be used, but with a localized message displayed.
- The language is determined by the current application language.

For more information about creating or administering iHelp, see *Siebel Applications Administration Guide* .

Siebel SmartScript

Siebel SmartScript is a language-independent module, which means that:

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

For more information about SmartScript, see *Siebel SmartScript Administration Guide* .

Setting Up Global Data

Global data is master data that is configured to display in the current application language. For example, a Siebel Call Center user accessing an FRA application would see data in French, while another user accessing a DEU application would see the same data in German.

To set up global data for your global deployment, follow steps similar to those in the example tasks in this topic.

Example for Enabling Global Data

The following sample task describes how to use the Opportunity Product business component to enable the Product Description field as global data. Make this update in the Opportunity Product project in your Siebel Tools workspace.

To enable global data (example)

1. In the Object Explorer, navigate to Business Component.
2. In the Name field, perform a query for Opportunity Product.
3. 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

4. In the Object Explorer, select Join, then Join Specification, and add the following record:

Field	Value
Name	Translate

Field	Value
Destination Column	PAR_ROW_ID
Source Field	Product Id

5. In the Object Explorer, select Join, then Join Constraint, and add the following record:

Field	Value
Name	Language
Destination Column	LANG_ID
Value	One of the following values: o Language() corresponds to the value of the Language Code parameter. o LOVLanguage() is the language currently used for LOVs. It might correspond to the value of the Language Code parameter or to the Preferred Language setting for the current user.

> **Note:** LOVLanguage() is designed to work with the multilingual outbound communication functionality. Language() does not work with this functionality.

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

Translating Globally Enabled Fields

The following sample task describes how to translate a globally enabled field, using the Product Description field from the preceding sample procedure as an example.

To translate a globally enabled field

1. Expose the view that contains the field that you want to translate.

   > **Note:** In the example provided previously in this topic, the view would already be exposed, because Products is an object that is global-data ready.

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

Configuring the Siebel Calendar

Calendar applets can be configured to respect the standard calendar layouts for many locales. This includes specifying the first day of the week shown in the first column of a weekly or monthly view and the workdays shown on a 5-Day calendar view. These settings can be populated by the Siebel administrator for each locale as explained in the section on customizing calendars in *Configuring Siebel Open UI* . The Calendar screen in your Siebel applications can be configured to support the requirements of users based in different locales in a global deployment.

Related Books

Siebel Applications Administration Guide

Using Siebel Tools

Configuring Siebel Business Applications

Configuring Siebel Open UI

Configuring Cascading Style Sheets to Specify Different Fonts

To successfully run Siebel CRM applications, your cascading style sheet files (CSS files) for Siebel applications must specify fonts that are available on your systems and that support the languages and locales in which you will run the applications.

By default, the style sheets specify fonts for Windows platforms only. The specified fonts are not Unicode fonts, and might not support the character sets that you require for all of the languages that you deploy. (The term *Unicode font* is used here loosely to refer to any font that contains a wide range of the characters required to support the many languages included in the Unicode standard.)

You can use utilities on your client systems, such as the Character Map utility on Windows, to determine which characters are available in any installed font.

For some example font settings that might be suitable on different operating system platforms for supported languages or locales, see the second table in this topic.

Some scenarios in which you might need to modify font specifications in your cascading style sheets include the following:

- If your database uses Unicode, and some users need to be able to view data using multiple languages, then the cascading style sheets applicable to these users might need to be modified to specify Unicode fonts.

  For example, if U.S. English (ENU) users need to view data using Japanese (JPN) characters, then Unicode fonts would be required for the ENU users, such as to change from Arial to Arial Unicode MS. If the JPN users do not require viewing data in other languages, then a Shift-JIS font could be specified for these users, because the display characteristics might be more suitable.

> **Note:** Style sheet files can include font names that contain non-ASCII characters only if the file is saved using UTF-8 encoding.

- If you are deploying your Siebel applications with any language that is not supported by the default fonts specified in the cascading style sheets, then the style sheets applicable to users for this language must be modified to specify a suitable font. See example fonts listed in the second table in this topic.
- Your style sheets must specify fonts that are available on the client computers and browsers where users run the Siebel applications. See example fonts listed in the second table in this topic.
- You might also need to specify different fonts in order to display special symbols such as mathematical symbols, to change the font to suit your company's design standards, and so on. (These scenarios are not specifically related to deploying languages.)

You might need to modify multiple cascading style sheet files, depending on which Siebel products and features you are deploying.

> **Note:** The first font specified in each applicable cascading style sheet must be valid. Additional font names specified in a list are not used.

Cascading style sheet files and other types of files, such as image files, are stored in the Siebel Application Interface installation. See *Siebel Installation Guide* .

For more information about modifying cascading style sheets, see *Configuring Siebel Business Applications* , *Siebel Developer's Reference* , and *Configuring Siebel Open UI* . See also 477498.1 (Article ID) on My Oracle Support. This document was previously published as Siebel FAQ 2104.

The following table lists some example font settings that might be suitable on different operating system platforms for installations supporting multiple languages or locales. These are Unicode-enabled fonts. Other suitable fonts might be available from font vendors. See also *About Unicode Fonts*.

Windows Font Names	**UNIX Font Names**			
Windows	**AIX**	**HP-UX**	**Linux**	**Solaris**
Arial Unicode MS (usually available only to licensed users of Microsoft Office) Oracle Albany	IBM Thorndale Worldtype (X11.fnt.ucs.ttf) Oracle Albany	Worldtype (X11.fnt.ucs.ttf) Oracle Albany	Oracle Albany	Worldtype (X11.fnt.ucs.ttf) Oracle Albany

The following table lists some example font settings that might be suitable on different operating system platforms for installations supporting a single language or locale. Where applicable, Unicode-enabled fonts such as those listed in the first table in this topic might be deployed instead. Other suitable fonts might be available from font vendors.

> **Note:** Font recommendations are not available at this time for Polish or Russian languages on UNIX or Linux client platforms.

Language or Locale	Siebel Language Code	Windows Font Names	UNIX Font Names			
		Windows	AIX	HP-UX	Linux	Solaris
Arabic (Saudi Arabia)	ARA	Arial	Times New Roman WT J	Naskh	ClearlyU Arabic Extra	Naskh MT
Chinese (Simplified)	CHS	SimSun	Times New Roman WT J	Zycjksun	ClearlyU	Zycjksun
Chinese (Traditional)	CHT	MingLiu	Times New Roman WT J	ARMingtiL	ClearlyU	Zycjksun
Czech	CSY	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Danish	DAN	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Dutch	NLD	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
English (United States)	ENU	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Finnish	FIN	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
French (France)	FRA	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
German (Germany)	DEU	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Hebrew	HEB	Arial	Times New Roman WT J	Arial	ClearlyU	Arial
Italian	ITA	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Japanese	JPN	MS PGothic	Times New Roman WT J	(Font name uses Japanese characters)	ClearlyU	(Font name uses Japanese characters)
Korean	KOR	MS Gulim	Times New Roman WT J	Hybatang	ClearlyU	Gothic

Language or Locale	Siebel Language Code	Windows Font Names	UNIX Font Names			
		Windows	AIX	HP-UX	Linux	Solaris
Polish	PLK	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Portuguese (Brazil)	PTB	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Portuguese (Portugal)	PTG	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Russian	RUS	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Spanish (Modern)	ESN	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Swedish	SVE	Arial	Times New Roman WT J	Arial	Luxi Serif	Arial
Thai	THA	Tahoma	Times New Roman WT J	Angsana	ClearlyU	AngsanaUPC
Turkish	TRK	Arial	Arial	Arial	XFree86-ISO8859-9-75dp fonts	Arial

Verifying Bidirectional Capability

Siebel applications display a left-to-right or right-to-left user interface, according to the current language and locale. For example, Siebel CRM applications display from right-to-left for Arabic (ARA) or Hebrew (HEB).

Framesets in Siebel Web templates, as used by the Siebel Web Engine, automatically reflect the UI directionality (either left-to-right or right-to-left) defined for the current locale.

Verifying UI Directionality Settings for Arabic or Hebrew

User interface elements display from right-to-left for Arabic or Hebrew. It is recommended to verify that UI directionality is set to Right To Left for the Arabic and Hebrew locales.

- For the Siebel Web Client, the UI directionality setting is already correctly defined by default as Right To Left for the Arabic and Hebrew locales, as shown in the Locales view, which is part of the Administration - Data screen. Do not modify this setting.

- For Siebel Mobile Web Clients and Developer Web Clients, verify the following setting in the Arabic or Hebrew application configuration (CFG) file for your Siebel application, such as uagent.cfg for Siebel Call Center:

```
[InfraObjMgr]
UIDirectionality = RTL
```

Modifying Fonts in Cascading Style Sheets

Depending on which fonts are available on your client computers, you might need to modify cascading style sheets (CSS files) to specify different fonts for use with your language. For more information, see *Configuring Cascading Style Sheets to Specify Different Fonts*.

Configuring Case Insensitivity

You can create indexes that directly support case-insensitive queries on eligible text columns. Such queries might be particularly useful for multilingual deployments.

The Case Insensitivity Wizard configures specified columns for case-insensitive queries by defining case-insensitive columns and indexes in the repository. The wizard also sets the Default Insensitivity property for these columns to DB Case & Accent.

The exact steps and requirements for configuring case and accent insensitivity might differ for new install scenarios and upgrade scenarios.

For detailed information about running the Case Insensitivity Wizard to configure case-insensitive queries for specified columns, see applicable background and task topics in the following documents on *Siebel Bookshelf* :

- *Configuring Siebel Business Applications*
- *Siebel Database Upgrade Guide*
- *Siebel Performance Tuning Guide*

5 Deploying Siebel CRM with Global Time Zone

Deploying Siebel CRM with Global Time Zone

This chapter discusses how to deploy your Siebel CRM applications using the global time zone feature, for both new deployments and upgrades. It includes the following topics:

- *About Global Time Zone and Universal Time Coordinated (UTC)*
- *Setting UTC System Preferences*
- *Setting the Database Server to UTC*
- *Enabling Custom Date-time Fields and Columns for UTC*
- *Converting Historical Data to UTC*
- *Administering Time Zones*

About Global Time Zone and Universal Time Coordinated (UTC)

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 Universal Time Coordinated (UTC). Using UTC is strongly recommended. UTC is enabled by default for Siebel applications.

> **CAUTION:** Oracle does not support new Siebel CRM deployments that do not use UTC. For existing deployments, if you do not want to deploy with UTC or are not ready to do so, then you must review *Guidelines for Both UTC and Non-UTC Deployments* and related topics. Also review this topic for relevant information.

For information about UTC in upgrade scenarios, see *Siebel Database Upgrade Guide* .

> **Note:** UTC is essentially the equivalent of Greenwich Mean Time (GMT). However, certain localities using GMT or British Standard Time (BST) as a time zone observe daylight savings time (DST). DST does not apply for UTC itself, because it serves as a baseline against which all of the other time zone and DST offset adjustments apply.

The global time zone feature allows you to store date-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.

Note: Windows regional settings on client computers do not affect UTC. Optionally, you can set the database server to UTC, although you can deploy your Siebel applications to use UTC without doing so. For more information, see *Setting the Database Server to UTC*.

This topic contains the following information:

- *About UTC System Preferences*
- *About UTC Data Conversion*
- *Example of UTC Data Conversion*
- *Guidelines for Both UTC and Non-UTC Deployments*
- *About Enabling UTC for Existing Deployments*
- *About Time-Related Functions and UTC*

About UTC System Preferences

This topic is part of *About Global Time Zone and Universal Time Coordinated (UTC)*.

The global time zone feature makes use of two UTC-related system preferences:

- **Universal Time Coordinated.** Set to True for UTC deployments (and set to True by default). As described in *Converting Historical Data to UTC*, the UTC conversion utility sets this system preference to True as its last step after completing the UTC conversion process. You do not need to set this system preference directly.
- **Default Time Zone.** Specifies a time zone that is to be used when the Time Zone field in the current user's profile is NULL, as described in *About UTC Data Conversion*. Where a user has specified a default time zone, this time zone overrides the setting of the Default Time Zone system preference.
- For example, for a company with the majority of employees in California, you might set this system preference to *(GMT-08:00) Pacific Time (US & Canada); Tijuana*, signifying GMT minus 8 hours. The value you enter must correspond to the name of the time zone as represented in the Time Zone Administration view. See also *Administering Time Zones*.

  Note: The Default Time Zone system preference has no default value. If you deploy with UTC, then you must specify a value for this system preference. Otherwise, incorrect time stamp values will be created and displayed where no time zone preference is specified for the current user.

  For information about setting the Universal Time Coordinated and Default Time Zone system preferences, see *Setting UTC System Preferences*.

Note: The Default Time Zone system preference is never used if Universal Time Coordinated is False (that is, if you are not using UTC). In this case, time stamps for records use the local time set for the operating system clock of the computer or operating system instance where you installed Siebel Server (which can use any time zone), except for time stamp values that are set explicitly such as appointment start or end times.

About UTC Data Conversion

This topic is part of *About Global Time Zone and Universal Time Coordinated (UTC)*.

The global time zone feature converts date-time data to and from UTC. When Siebel applications are deployed using UTC, some date-time fields store data as UTC data.

To support UTC conversion, applicable date-time field object definitions use the data type DTYPE_UTCDATETIME (instead of DTYPE_DATETIME), and applicable column object definitions use the physical type UTC Date Time (instead of Date Time). UTC conversion code is invoked by the Application Object Manager in user sessions where applicable, as described in this topic.

> **Note:** To be subject to UTC conversion, custom date-time fields that you create must be date-time fields *only*, not date-only or time-only fields, and must use the previously noted data types. See also *Enabling Custom Date-time Fields and Columns for UTC*.

There are some exceptions where date-time fields are not enabled for UTC. For more information or for help with configuring UTC, create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support.

Where the Universal Time Coordinated system preference is True, UTC conversion in runtime operations occurs as described in the following paragraphs:

- During runtime operation, date-time data that is entered by users (for example, appointment start or end times) 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, then the time zone defined by the Default Time Zone system preference is used for this purpose instead. Daylight savings time adjustments take effect according to the calendar date for the entered data.
- Date-time data that is displayed to users is converted *from* UTC based on the time zone that is stored in the user profile, including any daylight savings time adjustments. If the Time Zone field in the user's profile is NULL, then the time zone defined by the Default Time Zone system preference (with daylight savings time adjustments) is used for this purpose instead.
- Date-time data that is entered into the database by the system (date-time stamp) is converted to UTC by the Application Object Manager before it is stored, based on the time zone of the computer or operating system instance where you installed Siebel Server.
- Siebel components that are not Object Manager-based (for example, Workflow Policies or Siebel EIM components on the Siebel Server) invoke UTC-specific database functions in order to handle date-time data consistently for UTC deployments.

See also *About UTC System Preferences*.

Example of UTC Data Conversion

This topic is part of *About Global Time Zone and Universal Time Coordinated (UTC)*.

To illustrate how UTC data conversion works, suppose a scenario where User 1 is in New York, User 2 is in San Francisco, and the database server and the Siebel Server are in Utah. Today's date is November 20th. The time zone for User 1 is equivalent to UTC minus 5 (Eastern Standard Time), and the time zone for User 2 is equivalent to UTC minus 8 (Pacific Standard Time).

User 1 logs in and books a meeting with User 2. User 1 sets the meeting for December 15, 1 pm New York time, which is UTC minus 5. The time stored in the database is changed by the Application Object Manager component on the Siebel

Server before storage by applying this user's time zone offset, UTC minus 5, to the local time value. So the meeting time stored is 1 pm (EST) plus 5 hours, which is 6 pm UTC (1800 in 24-hour clock time, which is what is actually stored).

Note that because the meeting date is in December, and because the date when the users are viewing the meeting is in November, standard time applies and no adjustments for daylight savings time (DST) are necessary.

User 2 (in San Francisco) now looks at her calendar. She sees the time for the meeting as 10 am. The Application Object Manager has read the value of 1800 from the database and applied her time zone offset, UTC minus 8, to this value. 1800 minus 8 hours is 10 am (PST).

The described time-zone handling behavior can be generalized as follows:

- For a time zone offset like UTC minus x, the value x is added to a local time value to get the UTC time value for storage, or is subtracted from a UTC value to get the local time for display to a user.
- For a time zone offset like UTC plus x, the value x is subtracted from a local time value to get the UTC time value for storage, or is added to a UTC value to get the local time for display to a user.

Guidelines for Both UTC and Non-UTC Deployments

This topic is part of *About Global Time Zone and Universal Time Coordinated (UTC)*.

This topic describes some guidelines that apply for deploying with UTC, or for deploying without UTC. Additional requirements are noted in other topics to which they apply.

> **CAUTION:** Once you have already deployed your Siebel applications using UTC and created user date-time data in UTC format, *it is not supported to stop using UTC* by setting the Universal Time Coordinated system preference to False. Otherwise, incorrect time stamp values might be created and displayed.

If you do not want to deploy with UTC or are not ready to do so, then it is possible to set the Universal Time Coordinated system preference to False in limited circumstances:

- For a new installation, you can set Universal Time Coordinated to False after completing installation steps, *as long as you have not created any user date-time data in UTC format*.
- For an upgrade to the current version from a prior Siebel version deployed without UTC, Universal Time Coordinated might be either False or True, depending on the version you upgrade from. You can set this system preference to False, *as long as you have not created any user date-time data in UTC format, and have not converted historical date-time data to UTC format*.

For more information, see *About UTC System Preferences* and *Setting UTC System Preferences*.

> **CAUTION:** For an upgrade to the current version, if Universal Time Coordinated is True following the upgrade and you do intend to deploy with UTC, then you *must* run the UTC conversion utility before any new date-time data is created in UTC format. For details, see *Converting Historical Data to UTC*.

You perform all of the UTC conversion steps within the overall context of upgrade and other roll-out processes described in *Siebel Bookshelf* documentation.

For deployments using Siebel Replication Manager, first synchronize all of the regional nodes, then re-extract the regional nodes after you upgrade and convert to UTC.

For deployments using Siebel Remote and Siebel Mobile Web Clients, first synchronize all of the local databases, then reinitialize and re-extract the local databases after you upgrade and convert to UTC. Synchronize all of the local databases before upgrading.

If you deploy with UTC for new or upgrade deployments, then you must set the Default Time Zone system preference. See *About UTC System Preferences* and *Setting UTC System Preferences*.

When deploying with UTC for new or upgrade deployments that contain custom date-time fields, you must perform the tasks in *Enabling Custom Date-time Fields and Columns for UTC*.

Deploying Siebel applications using UTC does not require you to set the operating system clock for the database server computer or operating system instance to UTC (a GMT time zone without DST). This optional task is described in *Setting the Database Server to UTC*.

Siebel eScript provides a set of methods for working with UTC data, along with methods for non-UTC date-time data. Use the appropriate methods for your deployment. All processing of UTC data performed by eScripts is completely independent of the UTC conversion capabilities of the Application Object Manager.

> **CAUTION:** If you import appointment or activity data into the Siebel application (with UTC enabled) from another system that does not use UTC, then additional requirements apply. For more information, see *Converting Historical Data to UTC*.

Related Books

Siebel Installation Guide

Siebel Database Upgrade Guide

Siebel Remote and Replication Manager Administration Guide

Using Siebel Tools

Siebel Developer's Reference

Siebel eScript Language Reference

About Enabling UTC for Existing Deployments

This topic is part of *About Global Time Zone and Universal Time Coordinated (UTC)*.

If you are upgrading a non-UTC deployment of a prior version and are now deploying with UTC, or if you have decided to convert an existing non-UTC implementation to UTC, then you must perform applicable tasks described in the following topics:

- *Setting UTC System Preferences*
- *Setting the Database Server to UTC* (this task is optional)
- *Enabling Custom Date-time Fields and Columns for UTC* (this task also applies whenever you create custom date-time fields)
- *Converting Historical Data to UTC*

See also *Administering Time Zones*.

About Time-Related Functions and UTC

This topic is part of *About Global Time Zone and Universal Time Coordinated (UTC)*.

Users often need to know the current date or time when entering complex queries or building PDQs. The functions Timestamp() and Today() return different results where time zones are concerned. Regardless of whether UTC is enabled, it is possible for the date returned by these functions to be different, depending on where the servers are located and on the user's local time zone offset.

- Timestamp() returns the current date and time value adjusted for the time zone of the user.
- Today() returns only the current date, and this is relative to the time zone of the database server.

For more information about these and other functions, see *Siebel Developer's Reference* .

Setting UTC System Preferences

UTC deployments use two system preferences, as described in *About UTC System Preferences*.

Generally, you do not need to set the Universal Time Coordinated system preference, because it is set to True by default for new deployments, and the UTC conversion utility also sets it to True when it completes the conversion process.

However, for a UTC deployment, you must set the Default Time Zone system preference to the time zone that is to be used when the Time Zone field in the current user's profile is NULL. Complete the procedure in this topic to set this system preference.

For more information about upgrading historical data to UTC (in an upgrade or non-upgrade environment), see *Converting Historical Data to UTC*.

If you do not want to deploy with UTC or are not ready to do so, then you must review *Guidelines for Both UTC and Non-UTC Deployments* and related topics.

To set the UTC system preferences

1. Navigate to the Administration - Application screen, then the System Preferences view.
2. Locate the Universal Time Coordinated system preference and review the value in the System Preference Value field. The default value is True.
    - If you are deploying with UTC, then do not change the value. (Even if the value is False, as for an existing non-UTC deployment, you do not need to change it, because the UTC conversion utility sets it to True automatically.)
    - If you are deploying without UTC, then you can change the value to False under limited circumstances, as outlined in *Guidelines for Both UTC and Non-UTC Deployments*.
3. Locate the Default Time Zone system preference. In the System Preference Value field, enter your preferred default time zone.

    For example, for a company with the majority of employees in California, you might set this system preference to *(GMT-08:00) Pacific Time (US & Canada); Tijuana*, signifying GMT minus 8 hours. The value you enter must correspond to the name of the time zone as represented in the Time Zone Administration view. See also *Administering Time Zones*.

Setting the Database Server to UTC

As part of enabling UTC for a new or upgrade deployment, you can optionally set the operating system time of the database server for the Siebel database to a time zone based on GMT.

> **Note:** Deploying Siebel applications using UTC does not require you to set the operating system clock for the database server computer or operating system instance to UTC (a GMT time zone without DST), as described here. This task is optional. See also *Guidelines for Both UTC and Non-UTC Deployments*.

The specific method of setting the system time to UTC on the database server depends on the operating system you are using. Many operating systems have a time zone setting called UTC or something similar. Other systems have time zone options described as Greenwich Mean Time without daylight savings time (the equivalent of UTC).

> **Note:** Some databases, such as Oracle Database, provide listener programs which can be run on other computers or operating system instances than where the database is installed. If you set the system clock on the database server computer or operating system instance to UTC, then it is recommended to also set the time to UTC where the listener programs are installed.

To set the database server computer or operating system instance to use UTC in Windows

1. In the Windows task bar, double-click the time indicator to display the Date and Time Properties dialog box.
2. Click the Time Zone tab.
3. Specify (GMT) Casablanca, Monrovia.

   Select this GMT time zone for UTC because daylight savings time (DST) is not observed. For more information, see *About Global Time Zone and Universal Time Coordinated (UTC)*.
4. Turn off the option Automatically adjust clock for daylight saving changes.

   > **Note:** This step is not strictly required if you chose (GMT) Casablanca, Monrovia instead of a different GMT time zone, but can be considered precautionary.
5. Click OK.

To set the database server computer or operating system instance to use UTC in UNIX

- Set TZ=UTC0 in the `/etc/config` file.

Use the system-level time zone unless you need to have a different time zone for a particular purpose. In this case, set the time zone at the shell level by following the vendor documentation for your operating system.

Enabling Custom Date-time Fields and Columns for UTC

Some date-time fields (and their underlying columns) in Siebel CRM applications are enabled for UTC. Applicable fields include fields that are manually populated through entering date-time data and fields that the system populates by generating a date-time stamp. If you create custom date-time fields and columns that will store UTC data, then you must enable them for UTC, so that data entered in these fields is consistent with data entered in other UTC-enabled date-time fields.

See also *About UTC Data Conversion* and related topics in *About Global Time Zone and Universal Time Coordinated (UTC)*.

If you configure additional date-time columns for UTC, then you need to add these columns to the file utc_columns.inp. This input file tells the UTC conversion utility which columns need to be converted to UTC. Do this before you run the UTC conversion utility, which is described in *Converting Historical Data to UTC*.

For more information about configuring column and field object definitions, see *Using Siebel Tools* .

To enable custom date-time fields for UTC

- In Siebel Tools or Siebel Web Tools, set properties for each applicable column object definition and its associated field object definition:
  - For the column, set the Physical Type property to UTC Date Time.
  - For the field, set the Type property to DTYPE_UTCDATETIME.

If the Universal Time Coordinated system preference is set to True, then after you compile your changes these custom date-time fields will be enabled for UTC.

Converting Historical Data to UTC

When you enable the global time zone feature, you need to convert any historical (non-UTC) time data to UTC. Perform applicable tasks in this topic if you are upgrading from a Siebel deployment of a prior version that did not use UTC, or if you have decided to convert an existing non-UTC implementation to UTC.

See also *Guidelines for Both UTC and Non-UTC Deployments*.

Converting historical data helps ensure that existing date-time values are consistent with the global time zone logic. The global time zone feature stores date-time values in the database adjusted to UTC time. If you do not convert it, then your historical data will be incorrect for a UTC deployment.

For more information about how the global time zone feature processes date-time data in UTC, see *About UTC Data Conversion*.

Although enabling UTC is optional, it is recommended that you perform the following procedures immediately after upgrading to the current Siebel release. If you have upgraded a non-UTC deployment, then you *must* perform the

conversion steps described here before creating any new UTC date-time data, unless you have set the Universal Time Coordinated system preference to False. For more information, see *Guidelines for Both UTC and Non-UTC Deployments*.

The tasks in this topic run the UTC conversion utility from the Database Configuration Wizard. For more information about running the Database Configuration Wizard, see *Siebel Installation Guide* and *Siebel Database Upgrade Guide* .

Before beginning to convert historical data to UTC, you must upgrade your development and production environments to the current Siebel release. For more information, see *Siebel Database Upgrade Guide* .

Before you run the UTC conversion utility, drop all of the database triggers. You can re-create or re-enable the triggers after the UTC conversion is complete.

> **CAUTION:** If you are preparing external data such as activities or appointments to be brought into a Siebel database using Siebel EIM or Siebel EAI, then you must modify the time stamp for each individual record to be consistent with UTC. *This task is the customer's responsibility, and must be performed before the data is imported into the Siebel database.* For each record, the updated time stamp must be based on the applicable UTC offset. The offset must take into account both the time zone offset from UTC (for the appointment's location) and any applicable daylight savings time (DST) offset from the standard time zone that will be in effect (for the appointment's calendar date).

Process of Converting Historical Data to UTC

To convert historical data for use in an environment using UTC, perform the following tasks:

1. Reviewing *About the UTC Conversion Utility*
2. *Preparing Your Data for Conversion to UTC*
3. *Running the UTC Conversion Utility*
4. *Reviewing the UTC Conversion Log Files*

You might also need to do the following:

- *Manually Starting the UTC Conversion Utility*

About the UTC Conversion Utility

This topic is part of *Converting Historical 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 the Database Configuration Wizard, as described in *Running the UTC Conversion Utility*. This utility helps you define the required parameters for UTC conversion.

After conversion is complete, the UTC conversion utility sets the Universal Time Coordinated system preference to True. (When you start the utility, the setting can be either True or False.)

Running the UTC conversion utility reads a series of input files that control the conversion of your date-time data to UTC. Several input files are provided by default. On the Siebel Server computer or operating system instance where you installed the Database Configuration Utilities, these input files are located in the directory `SIEBEL_ROOT\DBSRVR\DB_PLATFORM` , where DB_PLATFORM is ORACLE, MSSQL, DB2UDB, or DB2390. The input file names start with utc and have the extension .inp.

The file driver_utc.ucf identifies the UTC conversion input files, which contain parameters specifying the columns that you are converting to UTC. Each of these input files updates the appropriate database columns to the UTC format in a single database transaction.

Each input file entry 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.

> **Note:** It is recommended to modify the default input files, or to create additional input files, to specify all of the applicable date-time columns to be converted, and to specify the most suitable conversion method for each column. The conversion methods are explained later in this topic.

Each input file entry is structured as follows:

```
[TABLE_NAME]
Clause = WHERE_CLAUSE
Column = COLUMN_NAME, CONVERSION_METHOD
...
```

where:

- TABLE_NAME is the database table containing the date-time columns that you will be converting.
- WHERE_CLAUSE, an optional value for Clause, can be used to specify a subset of columns to be converted. For an example, see *Preparing Your Data for Conversion to UTC*.
- COLUMN_NAME, the first value for Column, indicates a column to be converted (for example, CREATED).
- CONVERSION_METHOD, the second value for Column, indicates the conversion method to be used for this column. Possible values are SERVER_TIME and CREATED_BY.

Conversion Methods Used by the UTC Conversion Utility

The UTC conversion utility uses the following conversion methods:

- **SERVER_TIME.** Conversion will be done according to the time zone of the server, as specified in the file server_time.inp. This method is used for date and time values that are not associated with a user.

  You must use the correct syntax in the server_time.inp file. If the exact values are not specified, then the UTC conversion utility will not work. Use the time zone setting for your 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 U.S. Central Standard Time (CST):

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

- **CREATED_BY.** Conversion will be done according to the time zone of the user who created the record. The UTC conversion utility will look up the profile of the user who created the record being converted, and derive the corresponding time zone. In general, this conversion method is appropriate for converting data created by individual users, where creation time is relevant.

  An example of this type of data would be service requests originally created by users in multiple time zones, where for each record the time stamp before conversion is in the user's local time. In this case, you might choose to modify the input file s_srv_req.inp to use CREATED_BY instead of SERVER_TIME. Each record's time stamp will be converted to UTC based on the user's time zone.

Preparing Your Data for Conversion to UTC

This topic is part of *Converting Historical Data to UTC*.

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

- Configure custom fields and columns (if necessary) for UTC. For more information, see *Enabling Custom Date-time Fields and Columns for UTC*.
- Set time zones for each of your users.
- Save a report of your user time zones.
- Edit the driver_utc.ucf file to specify more input files.
- Modify the default input files as needed for your UTC deployment.
- Modify input files for partitioned tables, to assure sufficient log space.
- Allocate maximum database transaction log space.

Note: Log space is controlled through transaction logs or database log files.

Set Time Zones for Users

For each of your users, specify the time zone in the Contacts screen. This data is stored in the S_CONTACT table. You must also specify a value for the Default Time Zone system preference. For more information, see *Setting UTC System Preferences*.

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.

Edit the driver_utc.ucf File to Specify More Input Files

The file driver_utc.ucf identifies the input files for UTC conversion. If you create additional input files, then you need to add them to driver_utc.ucf. A default set of input files is provided. You might need to create additional files to specify more columns to convert.

An example from the driver_utc.ucf file appears as follows:

```
[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 driver_utc.ucf file parameters to accommodate additional input files

1. Specify 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 that you copied.
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 preceding example in the driver_utc.ucf file, change the name of the input file in the new step from this:

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

   to reflect the new input file name:

   ```
   /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 that corresponds to your new input file.

   Using the preceding example in the driver_utc.ucf file, change the log file name from this:

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

   to reflect the new log file name:

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

4. Repeat all of the preceding steps 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 preceding example in the driver_utc.ucf file (and assuming no preceding steps have changed), you would change the execute entry number in the new step from this:

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

   to reflect the next step in the sequence:

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

   and so on, for each subsequent step.

Modify Default Input Files as Needed

Modify the default input files for the UTC conversion utility as appropriate for your deployment, or create additional input files. An example of customizing input files to partition data into separate input files is provided later in this topic. See also *About the UTC Conversion Utility*.

Partitioning Prerequisite for Oracle Database

If you are using Oracle Database, some tables are partitioned by default, by using month (in the CREATED column). Make sure that the user running the UTC conversion utility has execute privilege on month. The following task describes how to give the execute permission to users who do not have it.

To grant execute permission

1. Connect to the database server as the tableowner.
2. Execute the following command:

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

Modify Input Files for Partitioned Tables

As the UTC conversion utility processes input files, it typically processes each table as a whole. However, tables with very large record counts might encounter errors due to constraints on log space at the database level.

The UTC conversion utility prevents errors that could 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 utility updates each record set individually to convert all of the rows in a partitioned table.

The UTC conversion utility uses partition keys to control how a table is divided into record sets. For example, large tables might 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 following tables are delivered with partition keys:

- S_CAMP_CON
- S_COMMUNICATION
- S_EVT_ACT
- S_SRV_REQ
- S_ORG_EXT
- S_CONTACT

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.

> **Note:** If you create additional partitioned files, then you might decrease performance. If you require a different partitioning method, or if you want reduced partitioning in order to optimize performance, then create a service request (SR) on My Oracle Support. Alternatively, contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

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

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

```
Column = TODO_PLAN_START_DT, CREATED_BY
```

Note: If the WHERE clause is blank, then the table will not be partitioned, and will be processed as a whole.

Allocate Maximum Log Space

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

Running the UTC Conversion Utility

This topic is part of *Converting Historical Data to UTC*.

Complete the following task to run the UTC conversion utility and convert your date-time data to UTC.

See also *About the UTC Conversion Utility* and *Preparing Your Data for Conversion to UTC*.

Note: Running the UTC conversion utility might take a long time, depending on how many records and how many columns are being converted.

For detailed information about starting the Database Configuration Wizard, see *Siebel Installation Guide* .

Note: The values that you enter into the Database Configuration Wizard are case-sensitive. You cannot type spaces in parameter values; use underscores (_) instead. Click Next to go to the next screen.

To run the UTC conversion utility from the Database Configuration Wizard

1. Start the Database Configuration Wizard, as described in *Siebel Installation Guide* .

   The Database Configuration Wizard appears.
2. Enter required parameters that apply for every Database Configuration Wizard task.
3. Choose the option Run Database Utilities.
4. Choose the option Universal Time Code Conversion.

   See the table that follows 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-time fields, the UTC conversion utility sets the value for the Universal Time Coordinated system preference to True.

   The utility checks for errors, and writes any errors to a log file.
5. Review the log file generated by the UTC conversion process, and resolve errors as necessary. See *Reviewing the UTC Conversion Log Files*.

   Note: If the UTC conversion fails for any reason, then you must review the log files and resolve any errors encountered. Then you need to manually rerun the UTC conversion utility, as described in *Manually Starting the UTC Conversion Utility*.

The following table shows parameters to use with the Database Configuration Wizard and the UTC conversion utility.

At this prompt...	Enter or select the following...
Siebel Gateway Address Enterprise Server Name	Specify the following: • The computer or operating system instance name where you installed Siebel Gateway. • The name of the Siebel Enterprise Server.
Siebel Server Directory	Specify the Siebel Server installation location: the `SIEBSRVR_ROOT` directory. For example, for the current version, the location might be `C:\siebel\ses\siebsrvr` on Microsoft Windows or `/siebel/ses/siebsrvr` on UNIX or Linux.
Siebel Database Server Directory	Specify the Siebel Database Configuration Utilities installation location: the `DBSRVR_ROOT` directory. For example, the location might be `C:\siebel\ses\dbsrvr` on Microsoft Windows or `/siebel/ses/dbsrvr` on UNIX or Linux.
Database Platform	Specify the RDBMS platform that you are using.
Siebel Database Operation	Choose Run Database Utilities.
Database Utilities Options	Choose Universal Time Code Conversion.
Base (Primary) Language	Specify the primary language for your deployment.
ODBC Data Source Name	Specify the ODBC data source. The data source was created automatically during Siebel Enterprise configuration, using the format EnterpriseName_DSN. **Note:** Do not enter the value *Siebel_DSN* unless Siebel is the name of the Enterprise. To find the name of the ODBC data source on Microsoft Windows, go to Start, then Settings, then Control Panel, then Administrative Tools, and then Data Sources (ODBC). Click the System DSN tab to find the name of the ODBC data source.
Database User Name and Password	Enter the database user name and password. (Specify the user name of the Siebel administrator, for example, SADMIN.)
Database Table Owner and Password	Enter the database table owner name and password. **Note:** For Microsoft SQL Server: this is the login for the owner of the database, not necessarily the default owner of the database in DBO.
UTC Repository Name	Enter the name of the Siebel repository for which you are performing UTC conversion.
Security Group ID	Enter the name of the security group ID, such as SSE_ROLE.

At this prompt...	Enter or select the following...
Log Output Directory	Enter the name of the directory in which log output is to be generated. By default the log output is generated in the `utc` subdirectory.
Apply Configuration Changes	Specify whether to apply configuration changes when the wizard completes, or if you will apply changes later.
Configuration Parameter Review	Review the parameters that you have defined and then click Finish.

Reviewing the UTC Conversion Log Files

This topic is part of *Converting Historical Data to UTC*.

After the UTC conversion utility runs, if errors are encountered, then the utility records those errors to log files. The log files are located in the `log` subdirectory of `SIEBSRVR_ROOT` .

Carefully review the log files for errors. Some of the errors listed in the log file are acceptable, but others will require resolution.

An example of an acceptable error is when the utc_drop_temp_tab.log file generates the error “ORA-00942: table or view does not exist” when initially dropping the S_CONTACT_TMP and S_TIMEZONE_TMP tables.

The following are the default log files. You might have additional log files if you modified the input files.

```
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 might have created, and resolve errors as necessary. If the UTC conversion utility was interrupted after it encountered an error, then you need to manually restart the UTC conversion utility.

Manually Starting the UTC Conversion Utility

This topic is part of *Converting Historical Data to UTC*.

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

To manually start the UTC conversion utility

1. Navigate to the `bin` subdirectory in `SIEBSRVR_ROOT` .
2. Enter the command to start the UTC conversion utility:
   - In a Windows DOS prompt, type: `siebupg.exe /m master_utc.ucf`
   - In a UNIX shell, type: `srvrupgwiz /m master_utc.ucf`

> **Note:** If you run the UTC conversion utility again to perform UTC conversion, then you must repeat all of the steps in this procedure.

Administering Time Zones

The time zone records that are shipped with Siebel CRM as seed data include the world time zones. These are the same time zones used by Microsoft in its operating systems. Time zone data includes the time zone name and standard abbreviation, its offset from UTC, daylight savings time (DST) data, and so on.

Procedures are provided later in this topic for modifying time zones and for maintaining translations of time zones.

For more information about managing time zone data, see also *Siebel Applications Administration Guide* .

Modifying Time Zones

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

Daylight savings time rules are defined in detail records for applicable time zone records. Multiple detail records can be defined for any time zone. Rules to accommodate changes that took effect in 2007 are defined for U.S. time zones subject to daylight savings time. (Various alerts on My Oracle Support describe DST issues and resolutions for previous Siebel CRM releases.)

Creating and using new time zone records is not supported.

To modify time zone settings

1. Navigate to the Administration - Data screen, then the Time Zone Administration view.
2. In the Time Zones list, select the time zone record that you want to modify.
3. Modify time zone fields as necessary. Fields are described in the following table.

Field	Description
Name	Full name of time zone. **Note:** For UTC deployments, you specify the full name of the applicable time zone as the value for the Default Time Zone system preference. For more information, see *About UTC System Preferences*.
Active	Specifies whether a time zone record is considered active (that is, when Active equals True).
Standard Abbreviation	The abbreviation for the time zone.
DST Abbreviation	The abbreviation for daylight savings time for the time zone.

4. In the Details list, modify daylight savings time rule settings as necessary, or create a new rule. (There is no New button, but you can create a new rule using Ctrl+N. Alternatively, you can copy an existing rule and modify the copy.) Fields are described in the following table.

Field	Description
Start Date	The date when a time zone rule is to take effect, such as 01/01/2021.
UTC Offset	The time difference in minutes between local time and UTC. For example, the (U.S.) Eastern Standard Time has a UTC offset of minus 300 minutes (GMT minus 5). Offsets are not necessarily in increments of hours (that is, divisible by 60). For example, Adelaide (Australia) has a UTC offset of 570 minutes (GMT plus 9:30), or 9.5 hours.
DST Bias	The difference in time, in minutes, DST makes (where applicable). For example, the (U.S.) Eastern Standard Time (GMT minus 5) has a DST bias of 60.
DST Start Ordinal	Part of the rule that determines when DST starts. For example, if the rule is *the first Sunday in April*, then *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*, then *Sunday* is defined in this field.

Field	Description
DST Start Month	Part of the rule that determines when DST starts. For example, if the rule is *the first Sunday in April*, then *April* is defined in this field.
DST Start Time	Start time for DST, measured in minutes into the day when DST starts. For example, 2 a.m., a common DST start time, is equivalent to 120 minutes.
DST End Ordinal	Part of the rule that determines when DST ends. For example, if the rule is *the last Sunday in October*, then *Last* is defined in this field.
DST End Day	Part of the rule that determines when DST ends.
DST End Month	Part of the rule that determines when DST ends.
DST End Time	End time for DST, measured in minutes into the day when DST ends.

5. After updating time zone settings, you must restart the Siebel Server.

   For more information about restarting the Siebel Server, see *Siebel System Administration Guide* .

Administering Time Zone Translations

You can view and maintain translations of the text-based fields for each time zone by using the Time Zone Administration view in the Administration - Data screen.

To administer translations of time zones

1. Navigate to the Administration - Data screen, then the Time Zone Administration view.
2. In the Time Zones list, select the time zone for which you want to view or modify translations.
3. The Time Zone Translations list displays existing translations for the current time zone. If you require a new translation, then add a new record and enter the translated version of the current time zone.
4. After updating time zone settings, you must restart the Siebel Server.

   For more information about restarting the Siebel Server, see *Siebel System Administration Guide* .

6 Localizing Global Deployments

Localizing Global Deployments

This chapter describes how to localize your customized Siebel CRM applications for a global deployment. It includes the following topics:

- *About the Localization Process*
- *Defining the Scope of the Localization*
- *Developing a Glossary for Translatable Terminology*
- *About Working with Translators*
- *Localizing Lists of Values and Multilingual Lists of Values*
- *Localizing an Unshipped Language*
- *About Testing Globalized Software*

About the Localization Process

Localizing a Siebel application includes translating the user interface and modifying other attributes to meet locale-specific requirements.

> **Note:** It is recommended that you complete as much configuration as possible *before* you begin localizing the application. One reason for doing so is that configuration potentially alters what needs to be translated, so it is better to have a stable application. Also, the check-in and check-out mechanism in Siebel Tools or Siebel Web Tools is meant to be performed in only one language; if you perform localization tasks on the client during configuration, then there is a potential for data loss.

For detailed information about working with user interface strings (including symbolic strings) and other locale-specific data, see *Using Siebel Tools* . You can use the Locale Management Utility (LMU) in Siebel Tools to manage strings and other data. More information about the LMU is provided later in this topic.

For information about localizing strings for Lists of Values (LOVs) and multilingual lists of values (MLOVs), see *Localizing Lists of Values and Multilingual Lists of Values*.

> **Note:** See also *Localizing an Unshipped Language*. In general, you perform the tasks in that topic, where applicable, *before* you perform the other localization tasks described in this chapter.

Localization Process

The localization process typically includes the following steps:

1. Identify the applications or projects that you want to localize.

2. Develop a localization glossary.
3. Export strings and other localizable attributes using the Locale Management Utility.

   This utility exports the objects to a flat file.

   When exporting symbolic strings, export the projects containing the symbolic strings, such as the Symbolic Strings project or a project that you created that contains symbolic strings. For guidelines, see *Using Siebel Tools* .

   > **Note:** If you are localizing strings for an unshipped language (such as ENG, for British English), then you might export strings from a source language that was shipped for Siebel CRM (such as ENU, for U.S. English), and specify your unshipped language as the target language (such as ENG). In the export file, copy the text from the source language to the target language. (Where many strings are similar or identical between the source and target languages, as with ENG and ENU, copying strings will speed up the translation process.) See also *Localizing an Unshipped Language*.

4. 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.
5. Import modified string and modified object definitions into the repository using the Locale Management Utility.
6. If necessary, then 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 changed objects.
7. With the correct language mode specified in Siebel Tools or Siebel Web Tools, publish the updated Siebel runtime repository.
8. Distribute the Siebel runtime repository to the appropriate Siebel Servers and clients.

Defining the Scope of the Localization

Define what it is that you need to have localized. The content to be localized could include any or all of the following items, or could include other items that are not listed:

- Software
- Strings in the user interface
- Lists of values
- Seed data
- Bitmaps
- iHelp
- Training materials
- Documentation
- Templates and correspondence

Nonlocalizable Elements

The following are nonlocalizable elements:

- Resource libraries such as DLL files

- Message files
- Log files
- Splash screens
- The following are selected applets for the Siebel client:
  - Change Records Popup Applet (SWE)
  - Parametric Search Form Applet

Developing a Glossary for Translatable Terminology

It is strongly recommended that you develop a localization glossary. Developing a glossary of your translatable terminology offers the following advantages:

- It maintains a consistent translation of terms over the lifetime of your localization.
- It shortens the time that it takes for successive translators to do their work, because the bulk of the terminology has already been translated and they can then spend their time deliberating on how to translate terms that are new in the current release.

  Note: Update the glossary with each release to reflect new translated terms.

- You can identify terms that you do not want translated, for example, legal names of marketable products or other terms.

The following table provides a sample of a localization glossary.

Siebel ENU Term	Siebel DEU Term	Customer-Specific Term
Account Affiliation	Firmenniederlassung	Beziehungen - Firma
Affiliations	Zugehörigkeiten	Beziehungen
Best/Visit Times	Best Times:günstigste Zeit	Besuchs- und Öffnungszeiten
Brick Subtype	Sektor Subtyp	Brick-Subtyp
Contact Affiliation	Kontaktaufnahme mit Zweigniederlassungen	Beziehungen - Person
Precall	Voranmeldung	Besuchsvorbereitung

Begin your localization effort by having your marketing department work with the translators to create this glossary. The marketing department can help the translators determine which terms should not be translated, while the translators can offer the marketing department suggestions on terminology in the target language from a localization standpoint.

About Working with Translators

This topic discusses issues in choosing and training translators, and managing localization schedules in coordination with your translators.

Choosing Translators

Consider the following questions when choosing translators to work with:

Will you use in-house translators or translation vendors?

Although in-house translators might seem cheaper in the short-term, bilingual workers often have other responsibilities and perform translations in addition to their main functions. Consequently, the results that you get might not be as good or as sustained as those you would get from hiring an outside translation vendor.

Another factor to consider is that, just because an employee is from the country that you are localizing for, it does not necessarily mean that they have the knowledge to translate your terminology correctly into the current professional vocabulary for your industry in the target locale.

If you do choose in-house translators, then make sure that they read the trade publications in your field published in their native country, at minimum.

There are many translation vendors to choose from, many of whom specialize in software internationalization, localization, or globalization, depending on your needs. For a selection of specialists, see the Web site of the Localization Industry Standards Association:

```
http://www.lisa.org
```

How do you choose the right vendor?

While cost is one factor to consider, to judge whether the cost quoted is good value or not depends on the services that you think you need and on what other vendors offer for a similar cost. Find out from each vendor what their charges entail and compare each vendor's offering based on the value of the service to you as well as on its cost.

For example, some vendors bundle the cost of project management into a quote, while others break out this cost. Some will charge extra for defect fixes, while others will not.

Do you require that your translators work on site or can they work remotely?

Most translation vendors use skilled specialists expert in translation for particular industries, who might be located in other U.S. states or even other countries, as opposed to in-house translators. Be prepared, whenever possible, to ship your translators flat files containing the localizable text strings and to clearly identify which strings should be translated.

Sometimes it might be most effective to work closely with qualified translators at your site. Doing so might be appropriate when the product is being localized very rapidly. Having translators on site allows defect-fixing to occur quickly. It might also be appropriate when the product requires engineers to input localized text, as when there is no separate localization utility. Having translators on site to work together with the engineers will be highly advantageous, but is generally more expensive.

Training Translators

Even the most experienced translators need training from you to learn about the products they are going to be localizing. Such training can include:

- Preparing a translator information kit that includes marketing brochures, marketing requirements documents, design documents, statements of direction, white papers, or other information that explains the nature of your company's business. Provide the translators with a context for their translation work.
- How to use the application for which they are translating text strings or documentation, and the purposes the application serves.
- How to report product defects during localization testing: for example, truncated, localized labels in the user interface.
- How to safely enter translation strings in files that your engineers send to the translators, such as by using email or FTP, and that the translators send back in a similar manner.

Localization Schedules

Agree to a Statement of Work or timeline (schedule) for the translation project early on. Understand what is involved on the translator's part, such as time needed for translation review either by vendor reviewers or by subject matter and language experts within your field offices. Developing a timeline helps all of the parties ensure that the timeline is realistic.

Consider whether your reviewers in the field consider their review of the translation a high priority when scheduling the timeline for their participation. Make sure that you have their manager's commitment for participating in the review.

If your software development project incurs delays, then do not expect the translators to shorten the time that you have agreed to for their efforts unless you are willing to accept lower quality. Good translations take time and no machine can take the place of a human being in this regard.

Localizing Lists of Values and Multilingual Lists of Values

Lists of values (LOVs), which are used in static picklists in the Siebel application user interface, are also localized for many multilingual deployments. LOVs can also be converted into multilingual lists of values (MLOVs).

MLOVs and multilingual picklists allow values to be selected by a user in one language and retrieved by users working in other languages. The value that is stored in the database for the record is the Language Independent Code (LIC) from the LOV record, rather than the Display Value, as is true for monolingual picklists.

For MLOVs to work correctly, Language Independent Code (LIC) and Display Value must always be consistent within the same LOV type and language, as follows. For records with the same LOV type and language, but different Parent LIC or organization:

- Multiple records with the same LIC value must have the same display value.
- Multiple records with the same display value must have the same LIC value.

Note: Where the LOV type is not organization-specific and not hierarchical, user keys normally prevent the creation of different LIC and display value combinations for the same LOV type and language on the server database. It is recommended to create LOVs on the server database whenever possible, because NULL value handling on the local database allows creating LOV records that do not meet the described requirements. (NULL handling is similar on Microsoft SQL Server.) Offending records violating user keys will not successfully replicate to the server database on synchronization.

As noted in *About Parameters for Language and Locale*, the Language Code parameter controls the language for MLOV display. If a Preferred Language is specified for a user, then this setting overrides the parameter value to determine the MLOV language.

In some localization efforts, you might choose to display LOVs or MLOVs in a common language rather than to use the same language as the application user interface. For example, for users working with a user interface in Czech (a language shipped for Siebel CRM) or Norwegian (an unshipped language), it might be appropriate for LOV or MLOV values to display in English, German, or some other language, depending on the business needs of your deployment.

Note: If you want LOV or MLOV values to display in an unshipped language, then you can copy values from one of the languages shipped for Siebel CRM and translate them into your unshipped language. In general, it is recommended to copy values from a language other than ENU, because many LOV values are provided for ENU that are not translated into any other language. For more information about localizing an unshipped language, see *Localizing an Unshipped Language*.

LOVs, including MLOVs, can also be organization-enabled, and some LOVs can also be hierarchical. For more information about configuring these types of LOVs, see 473813.1 (Article ID) on My Oracle Support. This document was previously published as Siebel Technical Note 632.

You must engage Oracle's Application Expert Services both for the initial configuration and for the upgrade of these items at the next Siebel version upgrade. Contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Note: In general, LOVs for newly deployed languages are inactive by default. You must enable the LOVs that you require by using Siebel EIM or the List of Values view in the Administration - Data screen. Alternatively, you can engage Oracle's Application Expert Services for this purpose.

For detailed information about creating and administering LOVs and MLOVs, and about converting LOVs to MLOVs, see *Configuring Siebel Business Applications* and *Siebel Applications Administration Guide* . Review all of the applicable guidelines and requirements.

Localizing an Unshipped Language

This topic describes how to localize an unshipped language.

In general, you perform the localization tasks described here *before* you perform other localization tasks described in this chapter. After performing the tasks described here, review your remaining localization needs for all of the files or data associated with an unshipped language.

An example of an unshipped language, Norwegian (NOR), is used throughout this topic.

Note: The tasks described here apply to the current Siebel CRM release. In general, tasks are performed after, or in conjunction with, the installation or migration tasks described in *Siebel Installation Guide* , and other tasks for preparing to use Siebel CRM applications. Although some localized elements for an unshipped language migrate as part of a migration installation from a previous release, the customer must verify the migration of all localized elements for unshipped languages, and might need to migrate certain elements manually. Note that some of the directories and files mentioned in this chapter have been reorganized relative to prior releases.

CAUTION: The process of localizing an unshipped language is considered to be customer configuration. Such configuration changes are not upgraded as part of the upgrade process described in *Siebel Database Upgrade Guide* .

See also *About the Localization Process* and *Localizing Lists of Values and Multilingual Lists of Values*.

Process of Localizing an Unshipped Language

To localize an unshipped language in your Siebel application, perform the following tasks:

1. *Creating Language and Locale Records*
2. *Creating List of Values Records for an Unshipped Language*
3. *Creating New Language Subdirectories and Copying Language-Specific Files*
4. *Creating Application Object Manager Components*
5. *Creating Application Configurations on the Siebel Application Interface*
6. *Configuring Siebel Mobile Web Clients*
7. *Testing an Unshipped Language*
8. *Completing Localization for an Unshipped Language*

Creating Language and Locale Records

This topic is part of *Localizing an Unshipped Language*.

Localizing an unshipped language requires that appropriate Siebel language and locale records exist in the Siebel database. If these records do not already exist, then you must create them in the Siebel application. You do this in the Administration - Data screen, in the Languages and Locale views.

Each language code used by Siebel applications uses a three-letter code, such as ENU for U.S. English, FRA for French, THA for Thai, and so on. Using language codes with only two characters does not work and is not supported. For more information about the language codes used by Siebel CRM, see *Language*.

These codes follow conventions used by Microsoft, where the first two letters represent a two-letter language code from ISO Standard 639, and the third letter can be used to differentiate the countries where this language is used. Multiple entries can represent different countries that use variants of the same language (for example, PTG for Portuguese - Portugal, and PTB for Portuguese - Brazil).

When you create language records for an unshipped language, use the language code names published by Microsoft. For example, the language code defined for Norwegian, a language that is not shipped for Siebel CRM, is NOR.

When creating language records, observe the following guidelines:

- Language codes must use three letters and must be defined using all capital letters (such as NOR rather than nor, for Norwegian).
- Your language code *must* use the same first two letters as the Microsoft code for the language, in order to ensure that the correct internal libraries will be used. For example, if you want to create a language code for

Austrian German, then name it DEA, so it will use the same code page as German (DEU). Similarly, use ENG for British English. For more information about the Microsoft codes, see Microsoft documentation.

- Where possible, use one of the existing language records that are listed in the Languages view in the Data - Administration screen. The Siebel language records are stored in the S_LANG table.

Two parameters, Language Code and (optionally) OM - Resource Language Code, are used to specify how an application uses languages. For more information, see *About Parameters for Language and Locale*.

To create a Siebel language record for an unshipped language

1. In the Siebel application, navigate to the Administration - Data screen, then the Languages view.
2. Verify that a record exists for the unshipped languages you are localizing. If necessary, then create a new record for the unshipped language.

   For example, for Norwegian, specify Norwegian as the Name value and specify NOR as the Code value.
3. Start Siebel Tools or Siebel Web Tools with a connection to the server database, and verify that the language record displays correctly. Also verify in the List of Values Administration screen that you can enter LOV records for the new language.

Note: If necessary, then, after you create the language record, also create a record for any new locale that you require. More information about locales is provided, as follows.

About Creating a Locale

If an existing Siebel locale does not cover the users of the unshipped language that you are implementing, then you must also create a locale record for the location for these users.

Locales correspond to user-specific or computer-specific settings such as Regional and Language Options in the Control Panel on Microsoft Windows.

For example, if you create a Norwegian (NOR) language record, then Norwegian-speaking users might use an existing locale, such as for Sweden, or might require using a locale for Norway in order to specify settings such as dialing code, date formatting, and so on. In some cases, time zones will differ between locales.

When creating a locale, copying an existing locale for which some of the same settings apply will make data entry easier. After you create a new locale, create translation records to provide the name for the locale for all of the languages that you are using.

Siebel locale records are stored in the S_LOCALE table.

See also *Additional Information About Setting Up and Administering Locales*.

For detailed information about creating locales, see *Siebel Applications Administration Guide* .

Creating List of Values Records for an Unshipped Language

This topic is part of *Localizing an Unshipped Language*.

You must create List of Values records for an unshipped language that you are implementing. You do this by copying certain existing List of Values records for the primary language and then modifying the copies.

To create List of Values records for an unshipped language

1. In the Siebel application, navigate to the Administration - Data screen, then the List Of Values view.
2. Query for records where Type equals ACCESS_RESOURCE_TYPE. This query should return at least four records.
3. Make a copy of each record returned by the query.
4. For each record copy, set values for the Language Independent Code (LIC) and Display Value fields the same as the corresponding record for the primary language (such as ENU). Set the language to the unshipped language, such as NOR, for Norwegian.
5. Set all of the other properties the same as the corresponding record for the primary language (for example: Active Flg, MultiLingual Flag, Order).

Creating New Language Subdirectories and Copying Language-Specific Files

This topic is part of *Localizing an Unshipped Language*.

When you are localizing an unshipped language, you create new language subdirectories to use for the unshipped language. You also copy language-specific files so you can use the copied files with the unshipped language.

If you localize the user interface and optionally MLOV values or other seed data into Norwegian (NOR), then you would set the Language Code parameter to NOR. However, resource library files located in language-specific subdirectories are not provided in Norwegian for Siebel CRM and cannot be localized into Norwegian. For this reason, you would set the OM - Resource Language Code parameter to DEU or ENU in order to use resource libraries from subdirectories for DEU or ENU. For more information, see *About Parameters for Language and Locale*.

You create language subdirectories on the Siebel Server, on the Siebel Application Interface, and on the Siebel Mobile Web Client, as described in the following tasks. The directory names that you use must correspond to the language code that you specified in *Creating Language and Locale Records*.

- For Siebel Server, you perform the applicable tasks in this topic and in *Creating Application Object Manager Components*.
- For Siebel Application Interface, you perform the applicable tasks in this topic and in *Creating Application Configurations on the Siebel Application Interface*.
- For Siebel Mobile Web Client, you perform the applicable tasks in this topic and in *Configuring Siebel Mobile Web Clients*.

> **Note:** Follow-up tasks might be required after performing the tasks in this section. For example, if the original configuration files that you copied are updated later, then you might have to recopy the original files or make similar updates in copied files. You also might need to make updates in the Siebel repository and publish them for users.

The following table explains the language placeholders that are referred to in the procedures in this task and in the remaining tasks listed in *Process of Localizing an Unshipped Language*.

Placeholder	Definition
XXX	The unshipped language that you are localizing, for example, NOR (Norwegian).

Placeholder	Definition
YYY	The base language of the application: for example, DEU (German) or ENU (U.S. English). In some cases, you copy files from language directories for the shipped language *YYY* into new directories for the unshipped language *XXX*. Sometimes you update the names of copied files to include *XXX* instead of *YYY*. **Note:** Alternatively, you can copy files from some other shipped language to the unshipped language that you are localizing, instead of copying them from the base language. For example, perhaps ENU (U.S. English) is the base language, but you choose to copy DEU files instead of ENU files.

Related Book

Siebel Installation Guide

Creating Language Subdirectories and Copying Language-Specific Files for an Unshipped Language on the Siebel Server

Use the following procedure to create language subdirectories and copy language-specific files for an unshipped language on the Siebel Server.

To create language subdirectories and copy language-specific files for an unshipped language on the Siebel Server

1. On the Siebel Server computer or operating system instance, create a new language subdirectory ***XXX*** under `SIEBSRVR_ROOT\bin`. Copy the configuration files from ***`SIEBSRVR_ROOT`***`\bin\`***`YYY`*** into `SIEBSRVR_ROOT\bin\`***`XXX`*** . The configuration files (that is, the CFG files such as uagent.cfg for Siebel Call Center) will be used by the Application Object Managers that you will create for new language *XXX*.

   Also copy all of the binary files (that is, the library files in the `bin\`***`YYY`*** directory) to the target directory, along with the configuration files.

   Note: After completing the localization tasks, you must recopy the files from `bin\`***`YYY`*** to `bin\`***`XXX`*** after each time that you install a new Siebel CRM release for the Siebel Server, to keep your localized product up to date. For more information about installing Siebel CRM releases, see *Siebel Installation Guide* .

2. Create a new language subdirectory ***XXX*** under `SIEBSRVR_ROOT\msgtmpl`. Copy the contents of `SIEBSRVR_ROOT\msgtmpl\`***`YYY`*** into directory ***XXX*** .

Related Books

Siebel Installation Guide

Configuring Siebel Business Applications

Creating Language Subdirectories and Copying Language-Specific Files for an Unshipped Language on the Siebel Application Interface

Use the following procedure to create language subdirectories and copy language-specific files for an unshipped language on the Siebel Application Interface.

To create language subdirectories and copy language-specific files for an unshipped language on the Siebel Application Interface

1. On the Siebel Application Interface computer or operating system instance, create a new language subdirectory ***XXX*** under **SIEBEL_AI_ROOT\applicationcontainer_external\siebelwebroot**, where *XXX* is the unshipped language that you are localizing, such as NOR for Norwegian. Then copy the directory **SIEBEL_AI_ROOT\applicationcontainer_external\siebelwebroot*YYY*\help** into directory **SIEBEL_AI_ROOT\applicationcontainer_external\siebelwebroot*XXX*** . Update these files for language *XXX*, as appropriate for your deployment.
2. In **SIEBEL_AI_ROOT\applicationcontainer_external\siebelwebroot\scripts**, copy the file swemessages_*yyy*.js (in the same directory) and rename the copy swemessages_*xxx*.js, such as swemessages_nor.js for Norwegian.

   > **Note:** After completing the localization tasks, you must recopy the file swemessages_*yyy*.js after each time that you install a new Siebel CRM release for the Siebel Application Interface, and then rename the file again, as described, to keep your localized product up to date. For more information about installing Siebel CRM releases, see *Siebel Installation Guide* .

3. Similar to what you did in Step 2, identify any other language-specific files, then copy the files and rename the copies to refer to new language *XXX* instead of existing language *YYY*.
4. In **SIEBEL_AI_ROOT\applicationcontainer_external\conf**, edit the file server.xml. Add entries for each unshipped language, as follows:

   ```
   ...
   <Context docBase="${catalina.base}/siebelwebroot/<lang>" path="/<lang>" />
   <Context docBase="${catalina.base}/siebelwebroot/<lang>" path="/<context_name>/<lang>" />
   ...
   ```

   For example, you might add the following entries to support Norwegian:

   ```
   ...
   <Context docBase="${catalina.base}/siebelwebroot/nor" path="/nor" />
   <Context docBase="${catalina.base}/siebelwebroot/nor" path="/siebel/nor" />
   ...
   ```

Related Books

Siebel Installation Guide

Siebel Security Guide

Creating Language Subdirectories and Copying Language-Specific Files for an Unshipped Language on the Siebel Mobile Web Client

Use the following procedure to create language subdirectories and copy language-specific files for an unshipped language on the Siebel Mobile Web Client.

To create language subdirectories and copy language-specific files for an unshipped language on a Siebel Mobile Web Client

1. Create a new language subdirectory ***XXX*** under **SIEBEL_CLIENT_ROOT\bin**. Copy the configuration files from **SIEBEL_CLIENT_ROOT\bin*YYY*** into directory ***XXX*** . The configuration files (CFG files such as uagent.cfg for Siebel Call Center) will be used by the application for language *XXX*.

Also copy all of the binary files (that is, the library files in the `bin\YYY` directory) to the target directory, along with the configuration files.

> **Note:** After completing the localization tasks, you must recopy the files from the `bin\YYY` directory after each time that you install a new Siebel CRM release for the Siebel client, to keep your localized product up to date. For more information about installing Siebel CRM releases, see *Siebel Installation Guide* .

2. Create a new language subdirectory `XXXSIA` (such as `NORSIA`) under `SIEBEL_CLIENT_ROOT\bin`. Copy the contents of `YYYSIA` (such as `DEUSIA` or `ENUSIA`) into directory `XXX` .

   > **Note:** After completing the localization tasks, you must recopy the files from `YYYSIA` to `XXXSIA` after each time that you install a new Siebel CRM release for the Siebel client, to keep your localized product up to date.

3. Create a new language subdirectory `XXX` under `SIEBEL_CLIENT_ROOT\msgtmpl`. Copy the contents of `SIEBEL_CLIENT_ROOT\msgtmpl\YYY` into directory `XXX` .
4. Create a new language subdirectory `XXX` under `SIEBEL_CLIENT_ROOT\public`, where *XXX* is the unshipped language that you are localizing, such as NOR for Norwegian. Then copy the contents of `SIEBEL_CLIENT_ROOT\public\YYY\help` into directory `SIEBEL_CLIENT_ROOT\public\XXX\help`. Update these files for language *XXX*, as appropriate for your deployment.
5. In `SIEBEL_CLIENT_ROOT\public\scripts`, copy the file swemessages_*yyy*.js (in the same directory) and rename the copy swemessages_*xxx*.js, such as swemessages_nor.js for Norwegian.

   > **Note:** After completing the localization tasks, you must recopy the file swemessages_*yyy*.js and then rename the file again, as described, after each time that you install a new Siebel CRM release for the Siebel client, to keep your localized product up to date.

6. Similar to what you did in the previous step, identify any other language-specific files, then copy the files and rename the copies to refer to new language *XXX* instead of existing language *YYY*.

Related Book

Siebel Installation Guide

Creating Application Object Manager Components

This topic is part of *Localizing an Unshipped Language*.

Next, you must create and configure Application Object Manager components that you are going to use for the unshipped language.

This task applies to Siebel Web Client users. It does not apply to Siebel Mobile Web Client or Developer Web Client users.

> **Tip:** Follow this procedure for a single component for testing purposes, before you create all of the other components that you require.

For detailed information about creating and configuring server components, see *Siebel System Administration Guide* .

To configure an Application Object Manager for an unshipped language

1. Start an employee application such as Siebel Call Center in the base language, such as DEU or ENU. Navigate to the Administration - Server Configuration screen, then the Enterprise Explorer view.

2. In the explorer tree, expand the enterprise, then click Component Definitions.
3. In the Component Definitions list, select an Application Object Manager component on which you will base your new component. For example, select Call Center Object Manager (ENU) then choose Copy Record from the menu.
4. Provide values like the following for the server component that you created:
    - **Component.** Call Center Object Manager (NOR)
    - **Alias.** SCCObjMgr_nor
    - **Component Type.** Application Object Manager (the value is copied automatically from the source record that you selected in the previous step)
    - **Component Group.** Siebel Call Center
    - **Description.** Call Center Object Manager (NOR) for Norway
    - **OM - Configuration File.** uagent.cfg
    - **Application Name.** Siebel Universal Agent
    - **Application Splashtext.** Siebel Call Center
    - **Application Title.** Siebel Call Center
5. Provide parameter values for this Application Object Manager component. In particular, specify the Language Code and OM - Resource Language Code parameters. Depending on the requirements for your deployment, you might set the language parameters in different ways.
    a. With the new component selected in the Component Definitions list, expand Component Definitions in the explorer tree, expand the selected Application Object Manager element, then click Parameters.
    b. In the Component Parameters list, query for the Language Code parameter. Set the Value field to the three-letter code for the language *XXX* that you are localizing, such as NOR for Norwegian.
    c. Query for the OM - Resource Language Code parameter. Set the Value field to the three-letter code for the shipped language *YYY* (such as DEU or ENU).
    d. With the new component selected in the Component Definitions list, expand Component Definitions in the explorer tree, expand the selected Application Object Manager element, then click Parameters.

    For more information about the language parameters, see *About Parameters for Language and Locale*.
6. Navigate to Administration - Server Configuration, Enterprise Explorer, Siebel Enterprise, Siebel, Component Definitions. Then click the Activate button to activate the new component definitions.
7. For Application Object Manager components used for Siebel Mobile disconnected application synchronization, also verify relevant documented settings, such as those in the Application Administration view in the Administration - Siebel Mobile screen.

Related Books

Siebel System Administration Guide

Siebel Mobile Guide: Disconnected

Creating Application Configurations on the Siebel Application Interface

This topic is part of *Localizing an Unshipped Language*.

For each new Application Object Manager component that you create for an unshipped language that you are localizing, as described in *Creating Application Object Manager Components*, you must create and configure an application configuration for the Siebel Application Interface, using Siebel Management Console.

You can copy existing application configuration settings and adapt them for the new language (for example, changing *enu* to *nor*), including selecting different options.

For more information about creating application configurations as part of configuring the Siebel Application Interface, see *Siebel Installation Guide* .

Related Book

Siebel Installation Guide

Configuring Siebel Mobile Web Clients

This topic is part of *Localizing an Unshipped Language*.

For a Siebel Mobile Web Client or Developer Web Client, in addition to the applicable steps described in *Creating New Language Subdirectories and Copying Language-Specific Files*, you must perform the steps described here for your unshipped language.

> **Note:** The Siebel Developer Web Client is not supported for user deployments. For more information, see *Siebel Installation Guide* .

Related Books

Siebel Installation Guide

Siebel System Administration Guide

Modifying Application Configuration Files

For any applications that you will support for Siebel Mobile Web Client or Developer Web Client users, you modify the configuration files located in the new language subdirectory `xxx` under `SIEBEL_CLIENT_ROOT\bin`, which you created by following the steps in *Creating New Language Subdirectories and Copying Language-Specific Files*.

For example, for Siebel Call Center for Norwegian (NOR), edit `SIEBEL_CLIENT_ROOT\bin\NOR\uagent.cfg` to include parameter values similar to the following:

```
[Siebel]
Language = NOR
ResourceLanguage = DEU
...
[InfraUIFramework]
MsgTemplateDir = C:\Siebel\Client\msgtempl\NOR
```

This example, where you set the Language parameter to NOR and set the ResourceLanguage parameter to DEU, ENU, or another value, supports one localization scenario. For more information, see *About Parameters for Language and Locale* and *Creating Application Object Manager Components*.

The Language parameter is equivalent to the server parameter Language Code. The ResourceLanguage parameter is equivalent to the server parameter OM - Resource Language Code.

In the configuration file, replace other language references, such as enu or ENU (for U.S. English), with nor or NOR (for Norwegian), where appropriate, to support your unshipped language.

Modifying Application Shortcuts for Siebel Mobile Web Clients

For a Siebel Mobile Web Client or Developer Web Client, create application shortcuts for your unshipped language *XXX* by copying existing shortcuts for the base language *YYY*.

For example, you might copy the shortcut for Siebel Call Center - ENU and rename it Siebel Call Center - NOR. Modify the properties of this shortcut so the shortcut target is similar to the following:

```
"C:\Siebel\Client\BIN\siebel.exe" /c "C:\Siebel\Client\bin\NOR\uagent.cfg"
```

Testing an Unshipped Language

This topic is part of *Localizing an Unshipped Language*.

You must test the Siebel application that uses the unshipped language. To some extent, interim testing can verify the tasks previously performed, prior to completing all of your localization tasks, as described in *Completing Localization for an Unshipped Language*. Retest again at appropriate points to verify all of the remaining localization steps.

To test the Siebel applications for the unshipped language

1. Restart the Siebel Server to have the new settings take effect.
2. Start the Application Object Manager for your unshipped language *XXX* (such as NOR for Norwegian) and test the application.

   If you copied elements from the base language *YYY* to use for the unshipped language *XXX*, then the application user interface still appears in language *YYY*, because you haven't yet performed the localization steps in *Completing Localization for an Unshipped Language*. The application splash screen, which is not localizable, always appears in the base language *YYY*. (For more information about nonlocalizable elements, see *Defining the Scope of the Localization*.)

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

3. Navigate to the Administration - Data screen, then the List of Values view. Select an LOV that is based on a column that can be MLOV-enabled.

Related Books

Configuring Siebel Business Applications

Using Siebel Tools

Configuring Siebel Open UI

Siebel Installation Guide

Completing Localization for an Unshipped Language

This topic is part of *Localizing an Unshipped Language*.

To complete the localization process for the unshipped language *XXX*, you must translate all of the appropriate content that is associated with this language. When you have translated repository elements, you must distribute the Siebel runtime repository for this language to Siebel Servers and Siebel Mobile Web Clients.

Language-specific content that you can translate or create includes the following items. See also the list of nonlocalizable elements in *Defining the Scope of the Localization*.

- (In Siebel Tools or Siebel Web Tools) Localizable Siebel Repository strings in the Siebel database. Add translations for symbolic string references, object locales, message categories, and so on.
- (In the Siebel application) Translatable language-specific values (Display Value field) for lists of values (LOVs) and multilingual lists of values (MLOVs).

  Add all of the values that you need for your new *XXX* language. For example, LOVs used by workflow processes must be localized into the new language.

  > **Note:** If an existing LOV record for your base language or another language does not have the Translate flag checked, then do not copy and translate this record. Copy only those records that are flagged as translatable.

  For detailed information about configuring LOVs and MLOVs, including the use of flags such as Translate and Multilingual, see *Configuring Siebel Business Applications* .
- (In Siebel application) Other seed data such as time zone translations, iHelp files, SmartScripts, correspondence templates, email templates, Siebel Anywhere administration data, and so on.
- Text that appears in image files or Web templates.

Test all of the changes before you complete the project and roll out the localized application.

For more information about tasks such as these that you perform in a Siebel Tools or Siebel Web Tools development environment, see books such as *Configuring Siebel Business Applications* and *Using Siebel Tools* .

Related Books

Configuring Siebel Business Applications

Using Siebel Tools

Configuring Siebel Open UI

Siebel Installation Guide

About Testing Globalized Software

This topic can help you to test the efforts of your software internationalization and localization teams.

System Configuration

To test the configuration, you first need to establish what your key platforms are for your database servers, your Siebel Servers, your clients, and your operating systems, for each language that you intend to deploy within your network.

You must test each supported platform, keeping in mind that there might be subtle differences or additional third-party software required in some cases.

Testing for Internationalization

Test your globally deployed applications to verify that all of the internationalization-related features function correctly.

For example, to test for character integrity during saving and retrieving data, you might create test cases that use a wide range of characters from your target languages. For a particular target language or set of languages, enter such data into fields and save the record. Check that you can search for the records and find them. Select the records and view them, checking that all of the data values that you input are unchanged. After entering, saving, and retrieving the data records in the target language, no square boxes or question marks should appear in the data.

Some of the issues to consider when creating test cases include:

- Test for character integrity during saving and retrieving data
- Test for user interface character integrity
- Verify interaction with native language operating systems and application response to locale settings
- Verify functionality with date format (including date-time fields) and time format
- Verify functionality with time zone support
- Verify functionality with number format
- Verify functionality with buffer sizes, check for overflow
- Test file names with international characters
- Verify accelerator keys
- Verify support for the euro currency and its symbol (€)
- Verify decimal symbol calculations and arithmetic operations
- Verify calendar functionality
- Verify list separator
- Verify reporting and printing operations
- Verify sorting
- Verify query or find operations
- Verify phone number formats
- Verify layout for name fields, for example support for middle initial
- Verify layout for address fields, for example support for state or province
- Verify that postal code is not a required field for a country with no postal codes
- Verify measurement units, metric or imperial

- Verify interaction with third-party software
- Verify import or export operations
- Verify taxation system support
- Verify that translations are not truncated or otherwise corrupt
- Verify that only translatable items are translated

Testing for Linguistic Quality

Use people in your field offices whenever possible to help you check the localization for linguistic accuracy and consistency. Test any custom terms with the field users, and pick terms that will be used often. For example, if you use a term like *currency rollup*, then make sure that it is correctly translated for your field users.

Using Automated Test Software

Third-party software is available to automate some test functions, although it does not replace the need for human testing.

Automated test software is very useful to test a wide range of characters, because an automated test can check all of the possible characters in Unicode, for instance. If an automated test is combined with a random test-data generator, then it is also likely to vary the data values much more widely than a human tester will.

On the other hand, an automated test cannot easily detect characters that have been clipped or otherwise hidden on the user interface, or other anomalies that the human eye can detect quickly.

Of course, it is important to use automated test software that supports Unicode data values and error messages containing failing data values. If several languages are being tested, then it will be most convenient if the test software can switch between each language automatically.

Defect Reporting

Build an efficient process for defect fixing between reviewers, translators, and the development team.

Make sure that your defect tracking system also supports Unicode data. It is very hard to explain a problem with data values in another language if the defect system does not allow you to put the failing characters into the title or the record of the defect report.

An example of such an application is Siebel Approval Manager. For more information, see *Siebel Approval Manager Guide* .