



SIEBEL⁷
eBusiness

GLOBAL DEPLOYMENT GUIDE

MIDMARKET EDITION

eBUSINESS APPLICATIONS

VERSION 7.0

12PA1-GD00-07000

OCTOBER 2001

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2001 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

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

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

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

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

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

Proprietary Information

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

Contents

Introduction

Who Should Use This Guide	Intro-2
How This Guide Is Organized	Intro-3
Additional Documentation	Intro-3
What's New in This Release	Intro-3
Contacting Siebel Technical Support	Intro-4
Siebel Systems Welcomes Your Comments	Intro-5

PART 1. Overview

Chapter 1. Overview of Global Deployment

About This Chapter	1-2
What Is Global Deployment?	1-2
Definition of Terms	1-3
Locale	1-3
Internationalization	1-4
Localization	1-10
Siebel Language	1-11
Platform	1-11
Configuration	1-12
Global Deployment	1-12
Planning Your Global Deployment	1-13

Chapter 2. Code Pages and Collation

About Code Pages	2-2
Using Code Pages	2-4
About Western European Microsoft Windows	2-5
Undisplayable Character Conversion	2-5
Collation of the Local and Sample Database	2-6

PART 2. Deploying in a Global Environment

Chapter 3. Global Deployment Scenarios

About Single Code Page Deployments	3-2
Monolingual Scenario	3-3
Multilingual Scenario	3-4

Chapter 4. Configuring Applications for Global Deployments

About This Chapter	4-2
Multilingual Picklists	4-3
About Lists of Values (LOVs)	4-4
Monolingual and Multilingual Lists of Values	4-5
Enabling the Multilingual List of Values	4-7
Integration Considerations	4-22
Configuration Considerations	4-23
MLOV Configuration and Coding Guidelines	4-23
Querying and Multilingual Lists of Values	4-24
Configuring Workflow Manager to Use MLOV-Enabled Fields	4-25
Configuring Assignment Manager to Use MLOV-Enabled Fields	4-29
Configuring Siebel Anywhere for Use with MLOV-Enabled Fields	4-31
Administering the Multilingual List of Values	4-32
Dynamic Toggle Applets	4-36

Global Time Zone Support	4-37
Data Conversion to and from UTC	4-37
Enabling UTC	4-38
Setting the Database to UTC	4-38
Setting UTC System Preferences	4-38
Configuring Fields for Global Time Zone Support	4-39
Administering Global Time Zone Support	4-40
Enabling Global Time Zone in an Upgraded Environment	4-41

Chapter 5. Additional Setup and Configuration

About This Chapter	5-2
Siebel eMail Response	5-2
Siebel eMail Response Overview	5-3
Planning eMail Response Deployment	5-5
Setup for eMail Response International Deployment	5-6
Limitations	5-9
Siebel Anywhere	5-10
Siebel Personalization	5-12
Siebel Reports Server	5-13

Chapter 6. Siebel Web Clients

Regional Settings in the Operating System	6-2
Setting Date Formats	6-3
About the Active Language	6-4
Online Help	6-4
Choosing Fonts	6-5
Third-Party Software	6-6
Application-Wide Data	6-6
Sales Cycles	6-6
Correspondence Templates	6-6
Literature	6-6
SmartScript and Campaigns	6-7

PART 3. Localizing Siebel eBusiness Applications

Chapter 7. Localization

About This Chapter	7-2
The Localization Process	7-2
What Can You Localize?	7-3
Using the Locale Management Utility	7-4
Finding Untranslated Text Strings	7-4
Exporting Strings and Other Locale-Specific Attributes	7-5
Importing Strings and Other Locale-Specific Attributes	7-6
Identifying Modified Objects	7-8

Chapter 8. Siebel Tools and Multilingual Development

Multi-Locale Repository	8-2
Setting the Tools Language Mode	8-4
Checking In and Out Locale-Specific Attributes	8-5
Compiling an .srf File for a Specific Language	8-6
Working with Visual Editors	8-7

Index

Introduction

Who Should Use This Guide	Intro-2
How This Guide Is Organized	Intro-3
Additional Documentation	Intro-3
What's New in This Release	Intro-3
Contacting Siebel Technical Support	Intro-4
Siebel Systems Welcomes Your Comments	Intro-5

Who Should Use This Guide

This guide provides an overview of deploying Siebel eBusiness Applications, MidMarket Edition in global environments. It includes overview information about global deployments, as well as configuration and localization information.

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

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

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

How This Guide Is Organized

This guide provides information necessary to understand, implement, and configure Siebel applications in more than one locale or language. Each chapter describes one aspect of global deployment. Everyone should read [Chapter 1, “Overview of Global Deployment,”](#) first. The other chapters can be read in any order.

Because many aspects of a global deployment are interrelated, you should read the entire book to fully understand the impact of any decisions you make during your global deployment.

Additional Documentation

The product documentation set for Siebel eBusiness Applications is provided on the *Siebel Bookshelf, MidMarket Edition* CD-ROM. For general information about Siebel product documentation, see the *Siebel Bookshelf, MidMarket Edition* home page, *Documentation Roadmap, MidMarket Edition*.

Siebel Systems, Inc., reserves the right to modify the documentation for Siebel eBusiness Applications at any time. For updates to Siebel documentation, go to <http://ebusiness.siebel.com/supportweb/>.

If you want to order additional Siebel documentation and copies of the *Documentation Roadmap, MidMarket Edition* CD-ROM, go to Books Online at <http://ebusiness.siebel.com/booksonline>.

To access both SupportWeb and Books Online, you will need to provide the user name and password you received from Siebel Support Services (support@siebel.com).

What’s New in This Release

For a list of features new in this release, see the “What’s New” books included on the *Siebel Bookshelf, MidMarket Edition*. Your Siebel implementation may not have all the features described in those guides, depending on which software modules you have purchased.

Contacting Siebel Technical Support

Do you know how to access Siebel Technical Support? It is crucial that you understand the requirements for getting support before you encounter technical issues that require Siebel Technical Support's assistance. This will facilitate smooth resolution of your issues. If you have questions, please don't hesitate to contact us.

To maximize your knowledge of Siebel products and your return on investment:

- You must attend Siebel training to become a *designated contact*.
- Your Siebel-trained designated contacts provide technical support to your users. Siebel Technical Support provides support directly to your designated contacts only.

To provide efficient, timely support and access to the Technical Support knowledge base:

- Siebel Technical Support is primarily Web-based; it can be accessed through Siebel SupportWeb (<http://ebusiness.siebel.com/supportweb/>). Please submit new service requests to us through SupportWeb, where you can also search the knowledge base for solutions.
- Designated contacts receive read/write access to Siebel SupportWeb. All other project team members at your company receive read-only accounts so that they can access the knowledge base.

To register for Siebel training, access <http://siebeluniversity.siebel.com/edPortal> and choose Implementation Team Training.

Please submit your technical issues and updates to Siebel SupportWeb (<http://ebusiness.siebel.com/supportweb/>). If you do not have a SupportWeb account, or if you have a question, please contact us at support@siebel.com or call your local Siebel Support Center:

- **North America:** + 1 800 214 0400 or + 1 650 295 5724
- **Brazil (São Paulo):** + 55 11 5110 0800
- **UK (London):** + 44 800 072 6787 or + 44 1784 494949
- **Germany (Munich):** + 49 89 95718 400

- **France (Paris):** + 44 800 072 6787 or + 44 1784 494949
- **Ireland (Galway):** + 44 800 072 6787 or + 44 1784 494949
- **Japan (Tokyo):** 0120 606 750 (Japan domestic only),
+ 81 3 5464 7948 (outside of Japan)
- **Singapore:** + 65 212 9266

Outside of local support center hours, Gold and Rollout Support Option customers can call + 1 800 214 0400 or + 1 650 295 5724.

We appreciate your business and look forward to working with you.

Siebel Systems Welcomes Your Comments

To help us improve our products, we want to know about any corrections or clarifications to this guide that you would find useful. Please include in your message:

- The title and version of the guide (very important)
- The name and version number of the Siebel eBusiness Application you are using
- Your name, job title or functional area, company name, phone number, and email address

Contact us through regular mail or email at:

Siebel Systems, Inc.
Technical Publications Department
2207 Bridgepointe Parkway
San Mateo, CA 94404-5009

doc@siebel.com

We appreciate your feedback.

Introduction

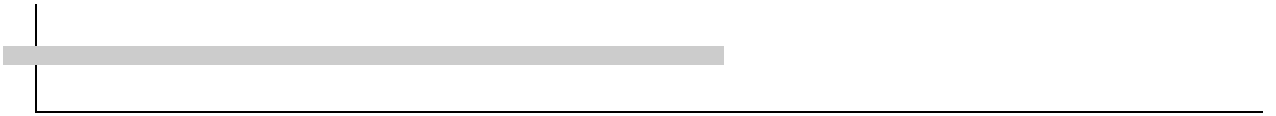
Siebel Systems Welcomes Your Comments

Overview

Part 1

Chapter 1. Overview of Global Deployment

Chapter 2. Code Pages and Collation



Overview of Global Deployment

1

About This Chapter	1-2
What Is Global Deployment?	1-2
Definition of Terms	1-3
Locale	1-3
Internationalization	1-4
Localization	1-10
Siebel Language	1-11
Platform	1-11
Configuration	1-12
Global Deployment	1-12
Planning Your Global Deployment	1-13

About This Chapter

The global deployment of a single software application can mean different things to different people. This chapter defines some common terminology, explores some global deployment scenarios, outlines how Siebel applications address the needs of a global deployment, and discusses how to plan your global deployment.

What Is Global Deployment?

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

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

A single installation of a Siebel application can, with some changes, run in more than one country simultaneously, even though people in these countries may speak different languages. This is called *concurrent multiple language support*, which means that the text in the user interface and the entries in picklists can be displayed to multiple users in any of the languages defined in your Siebel application. The language that is shown to the user varies according to the current active language setting of the Siebel Server or the Mobile Web Client, and the language setting used when the user's repository file was compiled.

Definition of Terms

This section defines the terms used when discussing the global deployment of software, and describes what these terms mean for your Siebel application deployment.

Locale

A locale is a set of rules guiding how common data is displayed to the user or received from the user. These rules are used to format data from a database before displaying the resulting information in output fields on the User Interface. For instance, date and time, numbers, and currencies are all displayed using locale rules.

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

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

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

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

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

On a Windows PC, locales are administered through the Regional Settings found on the Control Panel. For databases and other platforms, different methods are used to handle the locales. Please refer to your system documentation for detailed information about locales.

For more information about locales and Siebel Applications, see *Siebel Server Installation Guide, MidMarket Edition*.

Internationalization

Internationalization (often abbreviated as I18N since there are 18 characters between the initial I and the terminal N) refers to the process of making a software product localizable for any *locale*. Internationalization includes designing software to handle and display data, such as text, diagrams, and numbers, according to the orthography or rules of the language as used in a particular locale. The software might have to input, display, and print characters, sort text, and recognize numbers and dates in different formats. Therefore, certain engineering features must be incorporated into the code to handle these requirements.

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

Some aspects of internationalization include:

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

Your Siebel application has been internationalized. Specific features include:

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

For example, some DLLs are locale-independent, while other DLLs are locale-specific. Locale-specific DLLs are located in the locale-specific installation directory that they support.

- Euro currency support and triangulation
- String, number, and date handling

- Support for multilingual user data such as:
 - Multilingual picklists (MLOV seed data)
 - Multilingual data for product- and catalogue-related entities
- Support for major code pages

The Siebel architecture supports data entry and display in single-byte, multi-byte, and bidirectional languages. However, MidMarket applications support the following single-byte languages: American English, Italian, French, German, Brazilian Portuguese, and Spanish.

Formatting

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

Examples of different formats based on locales include:

- Date and Time
 - 03/10/95 (U.S. format as mm/dd/yy)
 - 950310 (international standard format as yymmdd)
 - 10.03.95 (German format as dd.mm.yy)
- Number
 - 1,234.34 (U.S. format with a comma as digit grouping symbol and a period for the decimal symbol)
 - 1 234,34 (French format with a space as the digit grouping symbol and a comma for the decimal symbol)
 - 1.234,34 (German format with a period as the digit grouping symbol and a comma for the decimal symbol)
- Phone Number
 - + 33 1-23 42 34 56 (French phone number as shown in U.S. regional settings)
 - (415) 295-5000 (U.S. phone number as shown in U.S. regional settings)

Multicurrency Support

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

- Currency Formatting
 - \$32.45 (U.S. format with currency symbol in front of the amount)
 - 99,40 DM (German format with currency symbol behind the amount)
 - Lit 3300 (Italian format with no digits behind the decimal separator)

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

Euro Currency Support

Support for the Euro is reflected in several features.

Triangulation. Triangulation means converting one European Monetary Union currency to another using the Euro as a basis. Triangulation is accomplished by first converting a given currency (for example, French francs) to the Euro and then converting the resulting sum to a second European Union currency (such as the Italian lira). For more information on triangulation, see *Applications Administration Guide, MidMarket Edition*.

Expandability to Dual Currency Display. All reports and applets that display total monetary values can be configured to display the sum in Euro. All list columns that display total monetary values also can display their results in Euro as well as the selected currency. Configuring for dual display is accomplished easily as an onsite customization using Siebel Tools.

Expandability to Dual Currency Display

This can be applied to all Business Components. Use the following procedure to configure dual-currency display.

To configure dual currency display

- 1** Create a new field in the business component which holds the currency code to which the conversion should be performed.
 - This field is *not* a foreign key to another table; it must be of Type = DTYPE_TEXT.
 - Specify as PickList = PickList Currency. In the corresponding Pick Map, associate 'Pick List Field' = 'Currency Code' with the just created new currency code field.
- 2** Create a new field in the business component which holds the converted currency amount.
 - This field must be of Type = DTYPE_CURRENCY.
 - It *must* be a calculated value field, Calculated Value = [unconverted amount] (the field 'unconverted amount' must also be of type DTYPE_CURRENCY).
 - Property 'Exchange Date Field' must point to a field of type DTYPE_DATETIME.
 - Property 'Currency Code Field' points to the currency code field of [Step 1](#).
- 3** Set the following control or list column properties in the applet that displays the converted currency.
 - Runtime = TRUE
 - A Pick or Detail applet need not be specified, because in this case your Siebel application automatically launches the default applet that matches the field type.
- 4** Before that currency conversion takes place, the underlying currency business component must be filled with a minimum of valid values. To access the lists of currencies, conversion dates, and exchange rates, from the application-level menu in your Siebel application, choose View > Site Map > Application Administration > Currencies.
 - The two currencies between which you want to convert must be marked as active (for example, name the original one O and the converted one C).

- At least one exchange rate value must be defined for O to C. (For the reverse conversion C to O, another exchange rate value is required.)
- At least one of the exchange rates of a certain exchange direction must have a date at or before the date that is used as 'Exchange Date'.

Example

The following is an example of adding dual currency display.

- 1** Go to the business component Opportunity.
 - a** Add a field for the currency code which is converted to (field is stored in a free place in extension table S_OPTY_X):
 - Name = My_Currency, Type = DTYPE_TEXT, Join = S_OPTY_X, Column = ATTRIB_03
 - PickList = PickList Currency
 - b** Add a record in its corresponding Pick Map:
Field = My_Currency, Pick List Field = Currency Code
- 2** Add in the business component Opportunity a field for the converted Revenue:
 - Name = My_Cvt_Revenue, Calculated = TRUE, Calculated Value = [Revenue]
 - Currency Code Field = My_Currency, Exchange Date Field = Sales Stage Date
 - Type = DTYPE_CURRENCY
- 3** Add a new list column to the Opportunity List Applet:
Field = My_Cvt_Revenue, Display Name = Converted Revenue, Runtime = TRUE

Software Architecture Design

The Siebel Server allows a business to segment its business logic either by function or by region. A user connects through the Siebel Gateway Server to an object manager. The following parameters determine the behavior of the object manager:

- **CFGApplication.** Determines the application that the object manager runs.
- **Language.** Determines the language of the GUI and the language of data, such as pick lists that are displayed to the user.
- **Locale Parameters.** Set of 15 locale-specific parameters that determine the format of dates, times, and numbers, and currencies that are displayed to the user.

NOTE: Each Siebel enterprise supports one code page.

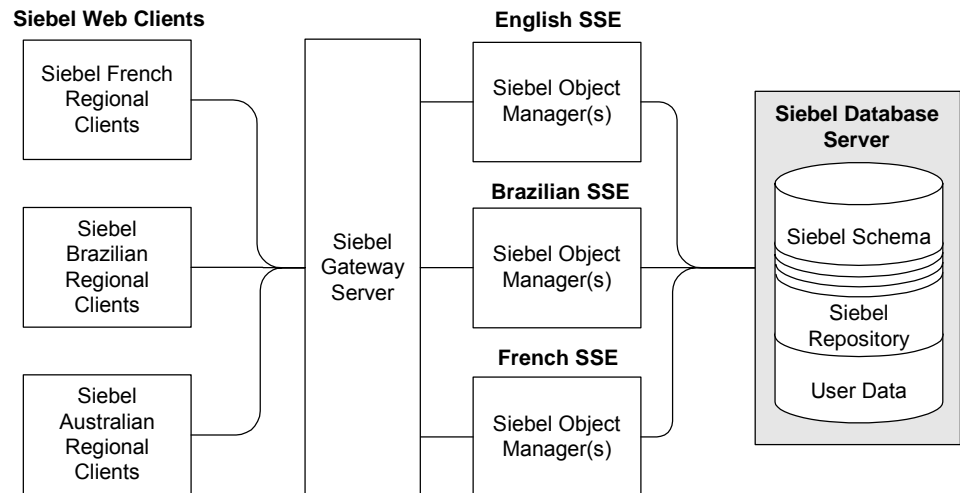


Figure 1-1. Siebel Software Architecture Design

For more information on Siebel architecture, see *Siebel Server Administration Guide, MidMarket Edition*.

Localization

Localization is the process of adapting a program for a specific locale. This can include:

- Translating text strings in the user interface
- Modifying the position, height, and width of controls
- Modifying images

Siebel applications are localized as required by the Siebel customer base. Local language releases are translated and elements of the user interface, including buttons, error messages, reports, online help, and log files, are configured to meet locale requirements. An example of this is shown in Figure 1-2 on page 1-10.

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

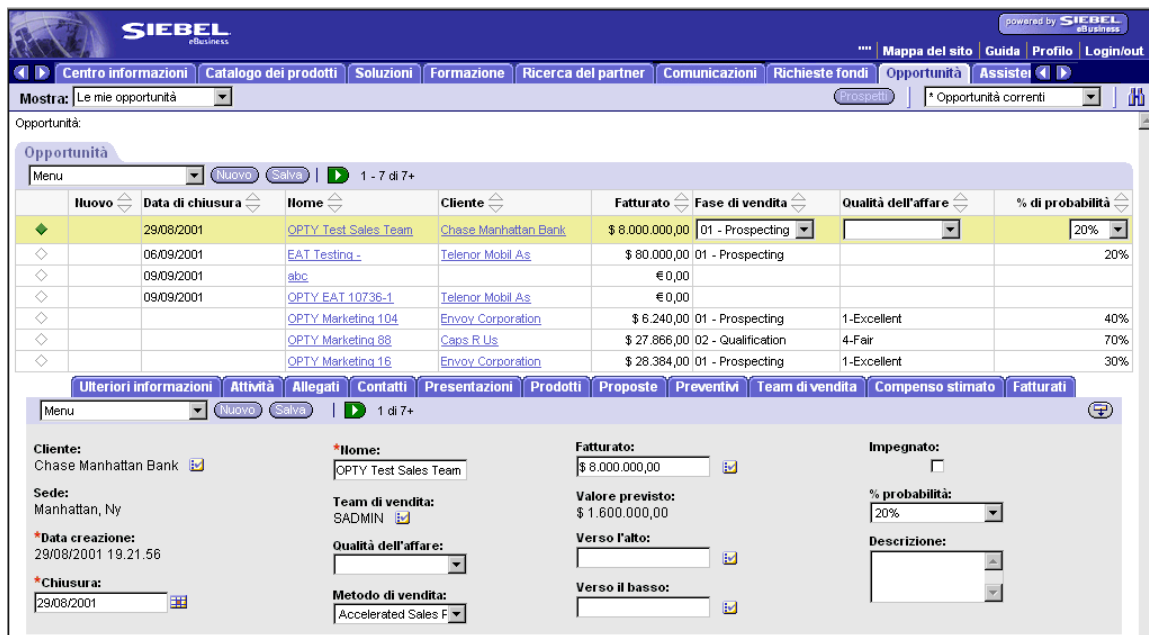


Figure 1-2. Localized User Interface

Siebel Language

The Siebel Language is the language of the Siebel application software installed on the system. The Siebel Language is seen in the Siebel user interface as well as in the messages and seed data installed with the application. The Siebel language is independent from the language of the data that the user enters in the Siebel database. The language of the data is determined by the platform on which the Siebel application is running. For example, although a user may be using an English version of a Siebel application, also referred to as English Siebel, the user could enter a contact name in French, as long as French is supported by the platform.

Platform

The definition of platform includes the following:

- The operating systems of the various entities of a Siebel deployment; the database, the Siebel Servers, and the Mobile and Dedicated Web Clients.
- The code page used by the entities in the Siebel deployment.

See [“Code Pages and Collation” on page 2-1](#) for a more detailed discussion on code pages.

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

This document will use the term *platform* in several places to discuss deployment options as well as specific functionality available in Siebel applications. Keep in mind that Siebel generally does not support mixed platform environments. The reason is that it isn't technically possible to manage an environment that uses multiple code pages on databases and Servers without a genuine risk of losing data in the process.

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

Configuration

Configuration is the process of modifying your Siebel application to change its look, feel, and behavior. You may need to configure your Siebel application during the course of your global deployment, in order to implement desired functionality.

Global Deployment

Global deployment is the process of installing, configuring, testing, and deploying a Siebel application in more than one locale and language. This guide covers many of the special considerations that pertain to global deployment.

Planning Your Global Deployment

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

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

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

- 5 Configure your application based on your company's business requirements.
- 6 Deploy the application to a global user base by doing the following:
 - Install the necessary Siebel Server, Gateway Server, Web Server Extensions, and language packs. This will install the language-specific repository (.srf) files, DLLs, and configuration files (.cfg).
 - Install the necessary database language packs. This will write the language-specific seed data into the database.

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

See Siebel Web Client Administration Guide, MidMarket Edition and Siebel Anywhere Administration Guide, MidMarket Edition for more information.
 - Perform application administration in the language of the database. For example, in Switzerland a database could be installed as German, so all application administration seed data is in German.

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

Code Pages and Collation

2

- About Code Pages2-2
- Using Code Pages2-4
- About Western European Microsoft Windows2-5
- Undisplayable Character Conversion2-5
- Collation of the Local and Sample Database2-6

About Code Pages

Databases store and manipulate data. How the data is interpreted is determined by a code page. A *code page* (also known as a character set) is a table that contains the characters and symbols used in a system or a database. A code page relates a numerical value to each character and symbol in the table. The numerical value is usually expressed as a hexadecimal value. For example, the English uppercase E in a Western European Windows code page occupies the code point 69.

Code pages are a means of providing support for character sets and languages used in different locales. Some code pages include multiple languages. All the standard code pages include ASCII as the first 128 (0-127) characters in the table, including uppercase A-Z, lowercase a-z, the numbers 0-9, and various punctuation symbols. Therefore every standard code page includes English as a possible language. If the letters, numbers, and symbols of a particular language's orthography are included in a code page, then the code page can be used to represent that language.

Siebel eBusiness Applications, MidMarket Edition supports code page 1252 and includes localized versions for English, French, German, Italian, Spanish, and Brazilian Portuguese. For a list of all languages that code page 1252 and other common code pages support, see [Table 2-1 on page 2-3](#).

Computer operating systems and databases can be configured to use a specific code page. Code pages are also defined for the database platform used for your Siebel application. This is decided when you set up your database; you cannot change a database code page at will.

Code page definitions are established as international standards and given an identifying number. Two common sets of code pages are for Windows and for ISO. Most Windows character sets are supersets of ISO (International Organization for Standards) character sets or are closely related to national standards.

Table 2-1. Common Code Pages and Their Languages

Windows Code Page Number	Languages	Commonly Used Code Page Name
1252	ASCII ¹ , Albanian, Basque, Catalan, Afrikaans, Danish, Dutch, Finnish, French, German, Icelandic, Italian, Norwegian, Portuguese, Spanish, Swedish	8859-1 (does not contain Euro Symbol) 8859-15 (contains Euro Symbol)
1250	Czech, German, Hungarian, Polish, Romanian, Slovak, Slovenian, and ASCII	8859-2
1253	Greek and ASCII	8859-7
1251	Russian and ASCII	8859-5
932	Japanese and ASCII	Shift-JIS
949	Korean and ASCII	KSC 5601
950	Traditional Chinese and ASCII	Big5
936	Simplified Chinese and ASCII	GBK

1. The English alphabet, 0-9, and other common characters, called ASCII, are part of all supported code pages.

NOTE: Some code pages such as 1252, can represent several languages—such as Brazilian Portuguese, English, French, German, and Spanish—simultaneously.

For more information about the code pages supported for your Siebel application, see *Siebel System Requirements and Supported Platforms, MidMarket Edition*.

Using Code Pages

The code page supported by your relational database will determine what languages you can support with your Siebel application. For instance, for an Oracle database, any Western European language can be supported with the WE8ISO8859P15 setting, along with the language and territory (american_america) in NLS_LANG. For more details, see [“About Western European Microsoft Windows” on page 2-5](#).

NOTE: You have to make sure that your database installation supports the appropriate character set. For more information about supported database platforms and supported code pages, see *Siebel System Requirements and Supported Platforms, MidMarket Edition*.

Even if your database supports a particular character set, whether or not a character stored in the database can be displayed to the user depends on several factors:

- The character has to be a member of the character set.

NOTE: The Siebel Server, the Siebel Mobile Web Client, and the Dedicated Web Client process data using the code page of the operating system on which they are running.

- The font in use must have a representation of that character.

It is conventional practice to use a substitute character, such as a small square or a question mark to indicate that a character cannot be accurately displayed.

When using a question mark for the substitute character, it should not be confused with the question mark used as punctuation. For example, if you see a text string of “????”, it is likely that it is a string of characters that cannot be displayed. However, in the text string of “Who?”, the question mark is a punctuation mark, not an undisplayable character.

About Western European Microsoft Windows

There is very little functional difference between the varieties of Western European Windows. The main difference is in the language used in the interface. In other words, Spanish and US Windows, both based on 1252, are essentially the same code. Only the language used in the menus and interface differs. You may therefore use Regional Settings in the Control Panel to have, for example, the Spanish Windows function like US Windows, or the US Windows function like Spanish Windows.

For more information about setting Regional Settings, see *Siebel Server Installation Guide, MidMarket Edition*.

Undisplayable Character Conversion

When a Siebel client or code page component accesses data that was entered using a client using a different code page, the ODBC layer may not be able to convert all data to the target code page. If this happens, the database connection layer will convert the specific data into question marks and it may get saved back to the database as question marks overwriting the original content. This is called undisplayable-character conversion.

Siebel Systems supports the Siebel Web client to run on a client operating system of any code page. However, characters (single-byte and multibyte) other than the code page of the object manager will be converted into the code page of the object manager. Siebel Systems does not support the storage of characters other than the code page of the object manager in the Siebel Database. The entire Siebel System (including the database, application Servers, Siebel Web Server Extension, and Mobile and Dedicated Web Clients) must run within the same code page.

Collation of the Local and Sample Database

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

For example, letters with accents may or may not be sorted with the same unaccented letters. If you have *a*, *ä*, and *b*, the following sort orders are possible:

Swedish	English
a	a
b	ä
ä	b

Swedish distinguishes the Umlaut, English does not.

NOTE: Character sets are not the same as the sorting order.

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

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

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

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

Code Pages and Collation

Collation of the Local and Sample Database

Deploying in a Global Environment

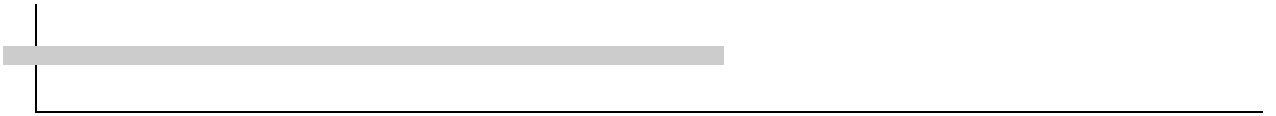
Part 2

Chapter 3. Global Deployment Scenarios

Chapter 4. Configuring Applications for Global Deployments

Chapter 5. Additional Setup and Configuration

Chapter 6. Siebel Web Clients



Global Deployment Scenarios

3

About Single Code Page Deployments	3-2
Monolingual Scenario	3-3
Multilingual Scenario	3-4

About Single Code Page Deployments

In a single-code page deployment model, the Siebel environment is set up to process and display data and the user interface in one code page. You can deploy a single language or multiple languages. For example deployments see “[Monolingual Scenario](#)” on page 3-3 and “[Multilingual Scenario](#)” on page 3-4.

The requirements and characteristics of a single-code page deployment are:

- You can deploy multiple instances that span multiple regions and languages all using the same code page setup.
- You can use all languages belonging to the code page, as long as the platforms, database, Server OS, and Client OS are set up for the same code page.

Siebel Systems supports running the Siebel Web client on a client OS of any code page. However, characters (single and multibyte) other than the code page of the Object Manager will be converted into the code page of the Object Manager. Siebel Systems does not support the storage of characters other than the code page of the object manager in the Siebel Database. The entire Siebel System (including the database, Siebel Servers, Siebel Web Server Extension, Mobile Web Clients and Dedicated Web Clients) must run within the same code page.

- Customization is done through a code page Siebel Tools Clients and UI languages using a code page Siebel repository.
- Replication and Connectivity are supported.
- MLOV enablement is supported.

NOTE: Some features may not be supported for certain UI language versions. See *Siebel System Requirements and Supported Platforms, MidMarket Edition* for more information.

Monolingual Scenario

In a monolingual scenario a single language is used across all Siebel deployments. This scenario simplifies data analysis and is relatively simple to configure and deploy. For example, [Figure 3-1](#) shows an example monolingual deployment of English using code page 1252.

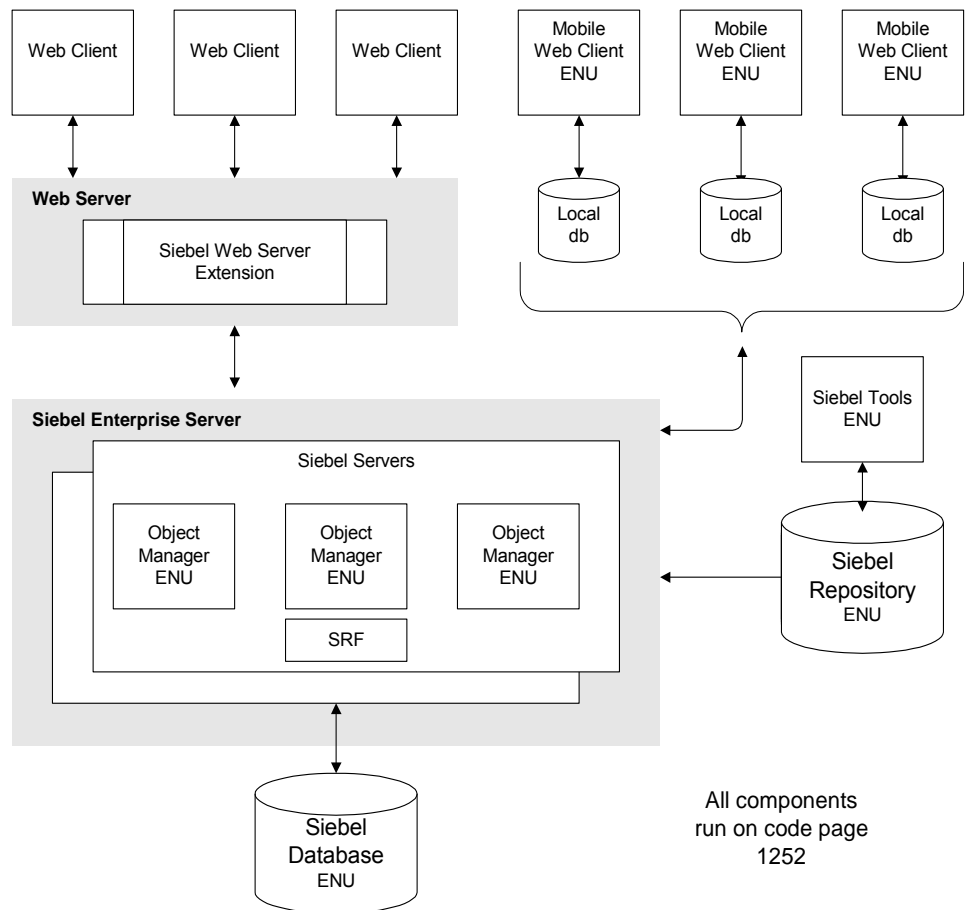


Figure 3-1. Example of a Monolingual Deployment

Multilingual Scenario

In a multilingual scenario, multiple languages are used across deployments. Multilingual development occurs within one code page. A single Siebel database and a single repository are used. For example, [Figure 3-2](#) shows an example Western European deployment (ENU, DEU, ESN) using code page 1252.

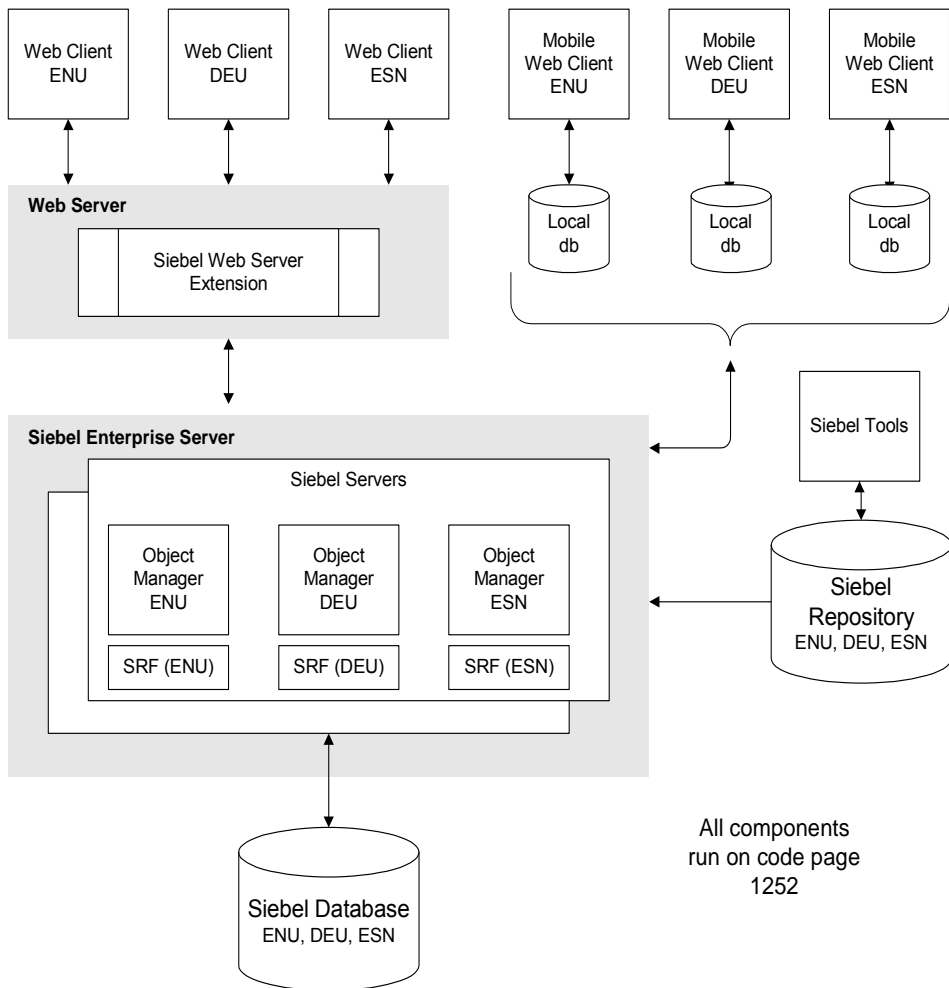


Figure 3-2. Example of a Multilingual Deployment

Configuring Applications for Global Deployments

4

About This Chapter	4-2
Multilingual Picklists	4-3
About Lists of Values (LOVs)	4-4
Monolingual and Multilingual Lists of Values	4-5
Enabling the Multilingual List of Values	4-7
Integration Considerations	4-22
Configuration Considerations	4-23
MLOV Configuration and Coding Guidelines	4-23
Querying and Multilingual Lists of Values	4-24
Configuring Workflow Manager to Use MLOV-Enabled Fields	4-25
Configuring Assignment Manager to Use MLOV-Enabled Fields	4-29
Configuring Siebel Anywhere for Use with MLOV-Enabled Fields	4-31
Administering the Multilingual List of Values	4-32
Dynamic Toggle Applets	4-36
Global Time Zone Support	4-37
Data Conversion to and from UTC	4-37
Enabling UTC	4-38
Setting the Database to UTC	4-38
Setting UTC System Preferences	4-38
Configuring Fields for Global Time Zone Support	4-39
Administering Global Time Zone Support	4-40
Enabling Global Time Zone in an Upgraded Environment	4-41

About This Chapter

This chapter describes concepts and tasks related to global deployments. For example, it covers how to have your application display multilingual picklists. You use Siebel Tools for some tasks covered in this chapter and administration views in the Siebel Web client for others.

The procedures in this chapter assume that you are familiar with using Siebel Tools to define objects such as views, applets, picklists, tables, columns, and so on. Additionally, the sections that cover enabling Workflow, Assignment Manager, and Siebel Anywhere assume you are familiar with those applications.

Multilingual Picklists

There are two kinds of picklists in Siebel applications, dynamic and static.

- **Dynamic picklists** draw values from other business components and display those values in an applet that can contain multiple columns and buttons.
- **Static picklists** draw their values from the list of values table and display those values in simple one-column lists. The values displayed in a static picklist are defined by an administrator. Static picklists can be bounded or unbounded.
 - Bounded picklists force the user to select a value from the list.
 - Unbounded picklists allow users to select a value from the picklist or type a value into the field.

You can configure bounded, static picklists to display multilingual values. This allows you to display values in the active language of the user. It also allows the values selected by a user in one language to be retrieved by users working in other languages.

An example of a multilingual picklist appears in [Figure 4-1](#). When the user clicks the drop-down arrow, a list appears with values displayed in the user’s active language that the user can select to populate the field.

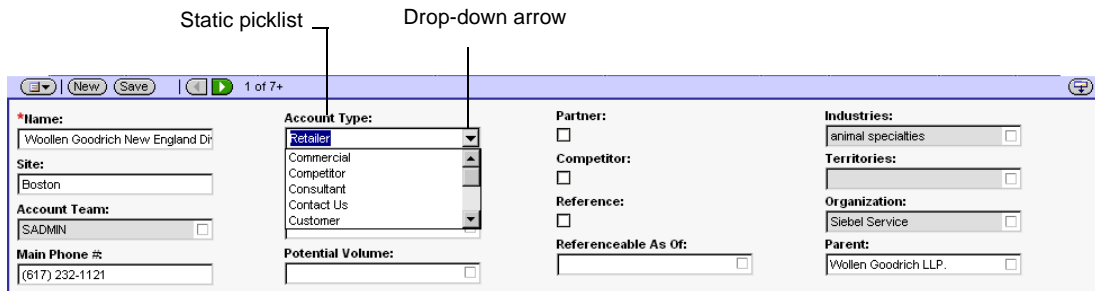


Figure 4-1. Static Picklist

About Lists of Values (LOVs)

Static picklists draw their values from a table that is called the List of Values Table (S_LST_OF_VAL). The data in this table is maintained by an administrator using the List of Values Administration view. See [“Administering the Multilingual List of Values” on page 4-32](#).

The List of Values table consists of many individual Lists of Values (or LOVs). Each LOV provides a set of selection values for a single static picklist. For example, the LOV for opportunity account type (see [Figure 4-1 on page 4-3](#)) includes the values Commercial, Competitor, Consultant, Contact Us, and Customer.

The set of selection values that make up an LOV are distinguished by the LOV type, which is defined in the Type column. Within each LOV, the available selection values are stored in the Display Value list column. An LOV consists of all type/name combinations with the same type. Two example LOVs are shown in [Table 4-1](#). Notice that the values with a Type of ACCOUNT_TYPE are the values that appear in the picklist shown in [Figure 4-1 on page 4-3](#).

Table 4-1. Example LOVs in the LOV Table

Type	Display Value
ACCOUNT_TYPE	Commercial
ACCOUNT_TYPE	Competitor
ACCOUNT_TYPE	Consultant
ACCOUNT_TYPE	Contact Us
ACCOUNT_TYPE	Customer
PERSON_TITLE	Mr.
PERSON_TITLE	Ms.
PERSON_TITLE	Dr.

For more information on maintaining LOVs in general, see *Applications Administration Guide, MidMarket Edition*.

Monolingual and Multilingual Lists of Values

A list of values can be implemented as either monolingual or multilingual, depending on the environment requirements. In either case, the picklist values that appear in the picklist are the same values that are entered in the target field when selected by a user. What is different is whether the values for the picklist are stored in a form that supports multiple translations.

Monolingual List of Values

Monolingual lists of values (LOVs) are appropriate when you are implementing a single-language version of Siebel eBusiness applications. It requires no special configuration. You enter selection values for picklists in the List of Values Administration view. See [“Administering the Multilingual List of Values” on page 4-32](#) for more information.

The values displayed in static picklists are the values stored in the Display Value column of the LOV table. When a user selects a value from a picklist, the display value is the value that populates the field in the user interface and the column in the base table.

Multilingual List of Values

Multilingual lists of values (MLOVs) are appropriate when you are implementing multiple-language versions of Siebel applications and picklists in multiple languages. MLOVs allows you to display values in static picklists in different languages. Values from MLOVs are stored in a form that users working in other languages can retrieve. MLOVs also allow you to roll up data for management reports, regardless of the language of the users who enter the data.

To implement MLOVs, you must configure target columns and add display values in the LOV table with the Multilingual attribute set to True. You configure the target columns in Siebel Tools, and you add display values using the List of Values administration view or in Siebel Tools. See [“Enabling the Multilingual List of Values” on page 4-7](#) and [“Administering the Multilingual List of Values” on page 4-32](#).

NOTE: If you do not add translated values for all entries, the user will see the language-independent code.

After target columns are enabled and multiple language display values are defined, end users are presented with picklists for that target column in their active language. When users view information from the target column, the language-independent code looks up the display value for their active language setting.

The active language is the language setting for the Siebel eBusiness Application client's user interface, as defined by the following:

- For Web clients the active language is determined by locale settings of the object manager.
- For Mobile Web Clients and Dedicated Web Clients, the active language is determined by the Language parameter in the configuration file (.cfg).

For more information about the active language, see [“About the Active Language” on page 6-4](#).

A picklist will show the Display Value for the active language but store the language-independent code when:

- The column of the field using the picklist has the Translation Table property set to S_LST_OF_VAL
- And the Lov Bounded Property is set to True
- And the picklist uses the same LOV Type as specified in the Lov Type property of the column.

NOTE: A picklist's LOV Type should always match the LOV Type of the underlying column (the column on which the pick list's field is based).

An MLOV saves the selected value as a *Language-Independent Code*. The language-independent code for each list of values entry appears in the Language Independent Code list column. Generally the language-independent code value is the same as the American English version of the same term.

NOTE: The language-independent Code length (the value stored in the database) must be equal to the longest display value for the MLOV. If it is not, the display value will be truncated. If the standard column does not meet your requirements and you are using a custom extension column, the column must be VARCHAR and have a maximum length (width) of 30.



Caution: Do not modify the language-independent code field.

Figure 4-2 on page 4-32 shows the List of Values Administration view for a MLOV.

Enabling the Multilingual List of Values

To set up an MLOV operation, you must modify the list of values configuration in Siebel Tools, as well as perform administration tasks. You can configure MLOVs by completing the following tasks.

- 1 “Identifying Which Columns to Enable” on page 4-8.
 - a “Making Sure the Type is Translatable” on page 4-9.
 - b “Determining If the Picklist Is Bounded” on page 4-10.
- 2 “Configuring the Multilingual List of Values in Siebel Tools” on page 4-12.
- 3 “Adding Translated Display Values in Application Administration” on page 4-14.
- 4 “Upgrading Existing Data Using the MLOV Upgrade Utility” on page 4-15.
- 5 “Recompiling and Deploying” on page 4-21.

These steps are discussed in the following sections.

NOTE: Configuration of MLOVs can impact performance, especially when the field whose picklist is configured for MLOVs is used as part of a search or sort. Performance characteristics should be considered and verified in conjunction with configuration of MLOVs.

Identifying Which Columns to Enable

Not every list of values type can be enabled as multilingual. You need to determine which columns you can enable based on the LOV type. LOV types must meet the following conditions:

- The column must be marked as translatable. See [“Making Sure the Type is Translatable” on page 4-9](#).
- The picklist must be bounded. See [“Determining If the Picklist Is Bounded” on page 4-10](#).
- The column must not be one of the [“Special Cases.”](#)

NOTE: Do not set up a column for a MLOV unless you are sure that you intend to use that column for your implementation.

Columns storing data that is read by server programs, such as Assignment Manager, Siebel Remote, Siebel Anywhere, or Workflow Manager, require additional configuration. See the following for more information:

- [“Configuring Workflow Manager to Use MLOV-Enabled Fields” on page 4-25](#)
- [“Configuring Assignment Manager to Use MLOV-Enabled Fields” on page 4-29](#)
- [“Configuring Siebel Anywhere for Use with MLOV-Enabled Fields” on page 4-31](#)

Configuring MLOVs may also include changes to the Siebel Visibility Rules. Any reference in a Visibility Rule to an LOV entry for a type you plan to configure for multilingual support must be changed from the Display Value to the language-independent Code. Check the visibility rules for references to any LOV entries as part of your configuration of MLOVs.

NOTE: Custom extension columns can always be MLOV enabled.

To check visibility rules

- 1** In Siebel Tools, navigate to the Dock Object Visibility Rules view.
Use the flat screen view to simplify searching.
- 2** Go to the SQL Statement field and search for literals across all rows that are not null.
- 3** Examine the results for values that need to be translated.

NOTE: Changing visibility rules requires the assistance of Siebel Expert Services.

Making Sure the Type is Translatable

A *translatable* type is a list of values type that can be modified, or translated into additional languages, without affecting the functionality of your application. This is indicated in the Translate setting in the List of Values Administration view in Siebel eBusiness Applications.

If an item is translatable, it can be modified without affecting Siebel eBusiness Applications functionality.

To determine if an LOV type is translatable

- 1** Connect to the server database using Siebel eBusiness Applications.
- 2** From the application-level menu, choose View > Site Map > Application Administration > List of Values.
- 3** For the LOV type that you are interested in, look in the Translate list column for a check mark.

If you add a LOV type, set this list column according to your configuration. Do not change any existing settings shipped with Siebel eBusiness Applications, because these are set to reflect the Siebel eBusiness Applications configuration. Changing this setting will not allow you to enable an LOV type.

Determining If the Picklist Is Bounded

A *bounded picklist* is a picklist where users must choose from the existing choices and cannot enter their own data. An *unbounded picklist* is a picklist where users may either enter their own data or choose from the existing choices. Only bounded picklists can be configured to be multilingual.

To determine if a picklist is bounded from the Picklists list

- 1 Connect to the server database using Siebel Tools.
- 2 Select the Flat tab in the Object Explorer.
- 3 Select the Pick List object type.
- 4 Query the Type Value property for the list of values type you are interested in.
If the Bounded property is checked, then that item is a bounded picklist.

To determine if a picklist is bounded from the Columns list

- 1 Connect to the server database using Siebel Tools.
- 2 Select the Flat tab in the Object Explorer.
- 3 Select the Column object type.
- 4 Query the LOV Type property for the list of values type you are interested in.
If the LOV Bounded property has a check mark, then the picklist for that target column is bounded.

NOTE: The *Translate* property of the column is only for internal use, and has nothing to do with the configuration of MLOVs.

The advantage of checking the list of columns rather than the list of Pick Lists is that you can also find out which tables use that list of values type. However, retrieving the columns list takes much more time.

All columns for a particular LOV type must be bounded. If any of the columns for the LOV type is not bounded, then none of the columns can be set to multilingual for that LOV type.

For example, [Table 4-2](#) shows the columns for the LOV type AVAILABILITY_STATUS. Although three of the columns are LOV bounded, you would not be able to enable these columns as multilingual, because one column (NEXT_AVAIL_CD) is Unbounded. If you were to run the MLOV Upgrade Utility, you would receive an error message that says the columns are inconsistently bounded. See [“About the MLOV Upgrade Log File” on page 4-20](#) for more information.

Table 4-2. Example of Inconsistently Bounded Columns

Name	LOV Type	LOV Bounded
CURR_AVAIL_CD	AVAILABILITY_STATUS	Y
NEXT_AVAIL_CD	AVAILABILITY_STATUS	Y
CURR_AVAIL_CD	AVAILABILITY_STATUS	Y
NEXT_AVAIL_CD	AVAILABILITY_STATUS	N

The Lov Bounded property in the column object is read-only for standard columns in Siebel eBusiness Applications, but is editable for columns that have been added using the Database Extension Designer. If you add columns, you must set both the Lov Type and the Lov Bounded properties for each column individually, consistent with your configuration.

Special Cases

There are special cases that should be considered when determining whether a column can be enabled for multilingual display. The following columns cannot be MLOV enabled.

Table 4-3. Columns That Cannot Be MLOV Enabled

Table	Column	LOV Type
S_OPTORG_CMPT_T	STRGY_CD	TAS_STRATEGY
S_PERIOD	PERF_ROLLUP_PRD_CD	PERIOD_TYPE
S_PERIOD	PERIOD_CD	PERIOD_TYPE
S_ORDER	CARRIER_CD	FS_CARRIER
S_ORDER	CARRIER_PRIO_CD	CARRIER_PRIORITY
S_ORDER	CRDT_CRD_NAME	PAY_ACCNT_CD
S_ORDER	DISCNT_PERCENT	DISCNT_PERCENT
S_ORDER	EAI_EXPRT_STAT_CD	EAI_ORDER_SYNC_STATUS
S_ORDER	FRGHT_TERMS_CD	FREIGHT_TERMS
S_ORDER	PAYMENT_TYPE_CD	PAYMENT_METHOD
S_ORDER	PRIO_CD	FS_INVLOC_ORDPRI
S_ORDER	SHIP_METH_CD	FS_SHIPPING_METHOD
S_ORDER	STATUS_CD	FS_ORDER_STATUS

Configuring the Multilingual List of Values in Siebel Tools

Once you have determined which columns to enable as multilingual, you configure those columns in Siebel Tools.

List of values types are enabled for multilingual support on a target-column basis. Because a list of values type can be used for different target columns, the Multilingual property must be implemented for all target columns that use the same type.

The following procedure describes the process for manually enabling a column. The list of values type ACCOUNT_TYPE is used as an example.

To enable a column for multilingual storage and display

- 1 Open Siebel Tools, and connect to the server database.
- 2 Select the Flat tab in the Object Explorer.
- 3 Select the Column object type.
- 4 Choose Query > New Query, enter the name of the desired list of values type in the LOV Type property, and then press ENTER to execute the query.

In this example, you would search for an LOV Type of ACCOUNT_TYPE. The query shows you the columns that use that LOV type, in this case there is only one column named OU_TYPE_CD.

Name	Parent Table	Lov Type	Lov Bounded	W	Changed	Project	User Name
OU_TYPE_CD	S_ORG_EXT	ACCOUNT_TYPE	<input checked="" type="checkbox"/>			Newtable	Account Type

- 5 Check that all the columns using that type have a check mark in the LOV Bounded property.
- 6 Change the Translation Table Name to S_LST_OF_VAL for all the columns returned by the query.

NOTE: Changing the properties of object definitions directly on the server is a nonstandard practice that is used only for configuration of columns for multilingual storage and display. Under all other circumstances, the correct and safe way to change object definition properties is to check out projects to the local repository, make the desired changes, and check them back in to the server. See *Siebel Tools Reference, MidMarket Edition* for more information about checking in and checking out projects.

Adding Translated Display Values in Application Administration

Once you have configured a column to be multilingual, display values must be defined for each language that will be supported.

To add translated display values

- 1** Using the Siebel Web client, connect to the server database.
- 2** From the application-level menu, choose Site Map > Application Administration > List of Values.
- 3** Find the list of values type for the enabled target columns.
- 4** For every language that will be supported, create a new record for each display value for that list of values type.

For instance, if you plan to support German and French in addition to the existing English display values, create two new records for each display value—one in German and one in French.

- 5** For each new record, the language-independent code must be the same as for the original record, but the entries in the Language and Display Value list columns are set differently, as appropriate.

For more information on adding and maintaining translated values, refer to [“Administering the Multilingual List of Values” on page 4-32](#).

Upgrading Existing Data Using the MLOV Upgrade Utility

After you have configured your application for use with MLOVs and added new display values for all the languages you intend to support, you must upgrade your existing LOV data. You do this using the MLOV Upgrade Utility.

NOTE: Even if you have just completed a new installation of your Siebel application, you must perform this data upgrade.

You run the MLOV Upgrade Utility in two modes.

- **Validation.** Running the utility in this mode validates the current repository for data inconsistencies. If the utility finds inconsistencies, the program stops and writes the errors to a log file.
- **Translation.** Running the utility in this mode:
 - Changes data in target columns that are configured for MLOVs from the display value to the language-independent code.
 - When you set the target column for an LOV Type to multilingual, the utility sets the MULTI-LINGUAL flag in the LOV table (S_LST_OF_VAL) to make sure of consistency between the multilingual state of the target column and its corresponding List of Values in the LOV table (S_LST_OF_VAL).
 - Verifies that all target columns using the desired MLOV type have been enabled.

NOTE: Target columns are columns that store either the display value or the language-independent code as part of user data.

The MLOV Upgrade Utility upgrades target columns that are marked as bounded and updates list of values types that are not already marked as multilingual. You can run the utility as often as you need to; only data that has not already been upgraded will be affected.

NOTE: The upgrade process run by MLOV Upgrade Utility is not reversible.

Running the MLOV Upgrade Utility

You run the MLOV Upgrade Utility using the Siebel Software Configuration Utility. The Siebel Software Configuration Utility is a wizard that will help you define the required parameters for running the MLOV Upgrade Utility. You run the utility in validation mode first, fix errors as they appear, and then run it in translate mode, which will enable your existing data for MLOVs.

NOTE: Before running the MLOV upgrade, drop all indexes from columns that you are upgrading. Once the MLOV upgrade is complete, re-create the indexes.

To upgrade existing LOV data using the MLOV Upgrade Utility

1 Run the MLOV Upgrade Utility in Validation Mode.

- a** Start the MLOV Upgrade Utility by choosing Start > Siebel Enterprise Server 7.0 > Configure DB Server.

NOTE: You can also start the Siebel Software Configuration Utility from the DOS Prompt command line. See [“To start the MLOV Upgrade Utility from the DOS Prompt” on page 4-18](#) for instructions.

- b** Enter the required parameters to run the MLOV Upgrade Utility in validation Mode.

See [Table 4-4](#) for a list of the wizard dialog boxes, options, and required values.

When you run the MLOV Upgrade Utility, it checks for errors and writes them to a log file. The default name of the log file is `mlovupgd_verify.log` and the default location is the `siebsrvr\LOG` directory.

2 Review the log file and resolve errors as necessary.

See [“About the MLOV Upgrade Log File” on page 4-20](#) for more information.

- 3** If an error is detected, you can resume running MLOV Upgrade Utility in validation mode by doing the following:
 - a** From the DOS Prompt, navigate to the \BIN directory of your Siebel Server root directory.

For example: `cd siebsrv\BIN`
 - b** At the command prompt, type:

`siebupg /m master_mlov_verify.ucf`

The MLOV Upgrade Utility resumes running.
- 4** Repeat [Step 1](#) and [Step 2 on page 4-16](#), and [Step 3](#), until no errors are detected.
- 5** Run the MLOV Upgrade Utility in Translation Mode.
 - a** Start the MLOV Upgrade Utility by choosing Start > Siebel Enterprise Server 7.0 > Configure DB Server.

The Siebel Configuration Utility appears.
 - b** Enter the required parameters to run the MLOV Upgrade Utility in translation mode.

See [Table 4-4 on page 4-19](#) for a list of the wizard dialog boxes, options, and required values.

The MLOV Upgrade Utility enables your existing data for MLOV. For columns configured for MLOVs, the MLOV Upgrade Utility finds LOV values in user data that are not in S_LST_OF_VAL and inserts them into S_LST_OF_VAL as inactive. It changes the display value of bounded columns to the language-independent code and sets the value for the Multilingual attribute to true.

To start the MLOV Upgrade Utility from the DOS Prompt

- 1 From the DOS Prompt, navigate to the \BIN directory of your Siebel Server root directory.

For example: `cd siebsrv/BIN`

- 2 Run the MLOV Upgrade Utility in validation mode by typing the following at the command prompt:

```
ssincfgw -l language_code -v Y
```

where *language_code* is the three-letter code (all capitals) for the language in which you want to display the GUI.

For example, to run the MLOV Upgrade Utility in English, you would type:

```
ssincfgw -l ENU -v Y
```

The Open dialog box appears.

- 3 Select `dbsrvr.scm` and then click Open.

The Siebel Software Configuration Utility -DB Server Configuration dialog box appears.

MLOV Upgrade Utility Parameters

To run the MLOV Upgrade Utility, complete the dialog boxes listed in [Table 4-4](#) and enter or select the values as you go.

Table 4-4. MLOV Upgrade Utility

In this dialog box	Enter or select the following
Siebel Enterprise Parameters: Gateway Server Address	Gateway Server Address Enterprise Server Address
Installation and Configuration Parameters: Siebel Server Directory	Siebel Server Directory
Installation and Configuration Parameters: Siebel Database Server Directory	Database Server Directory
Database Server Options: Siebel Database Operation	Run Database Utilities
Database Utilities: Database Utility Selection	Multi-lingual List of Values Conversion
MLOV Parameters: MLOV Operation	Validate or Translate, depending on the mode you want to run.
Installation and Configuration Parameters: Language Selection	Base language of your Siebel application.
Installation and Configuration Parameters: RDBMS Platform	RDBMS Platform
Installation and Configuration Parameters: ODBC Data Source Name	ODBC Data Source Name
Installation and Configuration Parameters: Database User Name	Database User Name Database Password
Installation and Configuration Parameters: Table Owner	Table Owner Name Table Owner Password
MLOV Parameters: Repository Name	Repository Name
Configuration Parameter Review	Review the parameters you have defined and then click Finish

About the MLOV Upgrade Log File

After the utility runs in either validation mode or translation mode, it writes any errors to a log file. The default names of the log files are `mlovupgd_verify.log` and `mlovupgd_translate.log`. The files are located in the `siebsrvr/LOG` directory.

- **LOVs Inconsistently Bounded or Translation Table Property Not Set to S_LST_VAL.** The message that appears in the log file for LOVs that have the bounded property on columns where they are used set inconsistently (one bounded and one not bounded) or LOV domains that do not have the Translation Table property set to `S_LST_VAL` is the following:

The following Validation checks for:

- 1- Two or more columns defined in the same LOV domain are inconsistently bounded (one bounded, one not)
- 2- Two or more columns are defined in the same LOV domain and at least one of them does not have a Translation Table Name of `S_LST_OF_VAL`.

Any errors of these types are listed in the log file. The information listed includes the LOV Type, Column, and Table.

To fix the LOV types that appear in the log file

- 1 Open Siebel Tools, and connect to the server database.
- 2 Select the Flat tab in the Object Explorer.
- 3 Select the Column object type.
- 4 Choose Query > New Query, and then enter the name of the list of values type that has a problem in the LOV Type property.
- 5 Press Enter to execute the query.
- 6 For all the columns displayed, make sure all of them are LOV BOUNDED = Y.
- 7 Set the Translation Table Name for all the columns displayed to `S_LST_OF_VAL`.

- 8 Run MLOV Upgrade Utility in validation mode to make sure that there are no more errors.
- **LOV Domains Not in the S_LST_OF_VAL Table.** The message that appears in the log file for LOV domains that are not represented in S_LST_OF_VAL table is the following:

The following Validation checks for:

```
LOV domains in the repository that are not represented in
S_LST_OF_VAL
```

This message means that an LOV domain is in the repository, but is not represented as a value in the list of values table, with a list of values type of LOV_TYPE. This can happen when you delete a record in the list of values table, instead of deactivating it, or when you enter an incorrect entry in the LOV Type property for a column added using a database extension.

For more information, refer to [“Deleting Versus Deactivating Records” on page 4-35](#).

To fix this problem, add the LOV domain in the List of Values Administration view and specify LOV_TYPE in the Type list column, or correct the entry in the LOV Type property in the repository. See [“Adding Records” on page 4-34](#) for more information.

For any values found in the target tables without matching records in the list of values table, the script will create a matching record in the list of values table. These records are marked as inactive. Remember to add language-specific entries for these base records, so that they display in the active language.

Recompiling and Deploying

Every time you change the configuration to enable another column to be multilingual, you must compile a new .srf file based on the newly configured repository. Only the Newtable project needs to be compiled again. For more information on compiling an .srf file, see *Siebel Tools Reference, MidMarket Edition*.

Additionally, you need to deploy the changes to users so that users can see the configured picklists in the desired language.

Integration Considerations

Enabling MLOVs does not affect just the Siebel eBusiness Applications client and the relevant target tables. Other features in your Siebel eBusiness Applications implementation must also consider this new configuration.

Enterprise Integration Manager

With Enterprise Integration Manager (EIM), you can import and export data. You can import data into both the list of values table and other tables in Siebel eBusiness Applications.

When importing data into the list of values table, the source table must have a language code and a name-value pair. This pair consists of the Display Value and the Language Independent Code.

When importing data into any other table, you must provide a language code for the `/LANGUAGE` command-line parameter for EIM. The source table must include the display value for multilingual columns in the language specified in the parameter. EIM validates imported data against list of values entries. The incoming data will be converted to associated language-independent codes during the import.

List of values entries that are marked inactive are ignored during the validation of multilingual LOV values during import by EIM.

When exporting data, you must specify a language code for the `/LANGUAGE` parameter, so that EIM can correctly translate the language-independent code in the table to the display value during the export.

For more information about command-line parameters for EIM, and information on EIM in general, see *Siebel Enterprise Integration Manager Administration Guide, MidMarket Edition*.

Configuration Considerations

MLOVs are implemented below the business component level, so there are no special configuration considerations, other than what is described here. Fields that point to MLOVs with enabled target columns will automatically return display values that match the client language setting.

Since MLOVs are configured on a column basis, target columns that are not configured to be multilingual will behave as before; that is, target columns will store display values instead of language-independent codes.

For display, the underlying language-independent code is converted to its corresponding display value using a Siebel eBusiness Applications lookup. For searching and sorting, however, a database join is performed by your Siebel application to the list of values table. Therefore, when configuring the application, make sure that any configuration directly involving the list of values table is compatible with your Siebel application MLOV functionality.

It is only possible to have one multilingual picklist type running off each column. This means that for a table that has more than one business component mapped to it and hence several fields mapped to the same column, it is not possible to have a multilingual LOV attached to only one of the fields. This is checked by the MLOV upgrade utility running in validation mode. See [“Upgrading Existing Data Using the MLOV Upgrade Utility” on page 4-15](#).

MLOV Configuration and Coding Guidelines

- **Pre/Post default values for fields with LOV picklists.** Always use the `LookupValue` function with `Expr`: in front of it; otherwise it will not work. The first argument is the LOV Type and the second is the LIC. The function returns the language-specific Display Value. For example:

```
Expr: "LookupValue ("FS_PROD_ALLOC_RULES", "Default")"
```

- **Dynamic drilldowns and toggle applets.** These are usually based on a field that has a LOV value. For example, a dynamic drilldown might navigate the user to a Credit Card screen if the account type is equal to *Credit Card* or to a Savings screen if the account type is equal to *Savings*. Do not hard-code the drilldown or toggle conditions. Rather, use the `LookupValue` function (as described in the previous bullet).

- **Search specs for business components, links, applets, and picklists.** Always use the LookupValue function. For example:

```
[Invoice Code] = LookupValue('FS_INVOICE_CODE', 'Auction')
```

- **VB functionality.** VB doesn't offer a function to retrieve the language-specific Display Value. However, the Display Value must never be hard-coded; you should use the language-independent code instead. To write VB code using the language-independent code only, you must create calculated fields that hold the language-specific translation for a language-independent code.

Querying and Multilingual Lists of Values

To run queries against fields that are controlled by MLOVs, use the Display Value for the search specification; do not use the Language Independent Code for querying. Querying will translate the search specification to the appropriate Language Independent Code to perform the query.

The Display Value used as the search specification should correspond to the Language being used by the application performing the query. If the query is being run through one of the Siebel interfaces (such as CORBA or COM), then the Language used for this translation is specified in the .cfg file used with the interface.

There is no difference to the user in the apparent functionality of the product when MLOVs are on or off. Internally, searches are applied using a function applied to the language-independent code. You can also do this with predefined queries and search expressions in the repository by using the same function (LookupValue (LOV Type, Language-Independent Code)). For more information about functions, see *Object Types Reference, MidMarket Edition*. For more information about the LookupValue function and how it is used with MLOVs, see [“MLOV Configuration and Coding Guidelines” on page 4-23](#).

Configuring Workflow Manager to Use MLOV-Enabled Fields

Additional configuration is required to enable Siebel Workflow to use MLOV-enabled fields. Siebel Workflow Manager compares values in target tables with values in Workflow Administration tables to determine whether a particular condition is true. For columns enabled for MLOVs, the value stored in the target table is the Language-Independent Code rather than the Display Value. However, the value in the Workflow Administration table is the Display Value. Workflow Manager cannot evaluate a condition by comparing the Language-Independent Code to the Display Value.

To enable Workflow Manager to work with MLOV-enabled columns, you must configure Workflow entities so that they compare the language-independent code in the target table with the language-independent code in the Workflow administration table. You must do this for the following entities:

- Policy Conditions
- Action Arguments

Policy Conditions

Before you enable Policy Conditions, you must:

- Determine all the business component fields that are enabled for MLOVs.
- Of the fields that are MLOV-enabled, identify which ones are referenced by Policy conditions.

For each of the fields that reference a Workflow Policy condition, you must complete the following tasks:

- Create a new picklist to display LIC values.
- Create a new applet to display LIC values.
- Configure the Workflow Policy Column to use the new picklist and applet.
- Repick the values for existing workflow policies.

To create a LIC picklist for a Workflow Policy Column

- 1** In Siebel Tools, navigate to the Workflow Policy Column object type that you want to enable to use with MLOVs.
- 2** Find the Workflow Policy Column that references the MLOV enabled field.
- 3** In the PickList property field, click the picklist name.
The Pick Lists window appears in the Object List Editor.
- 4** Create a new picklist by copying the existing one and append LIC to the name.
For example, Picklist Account Status LIC.

To create a new LIC applet for a Workflow Policy Column

- 1** Navigate back to the Workflow Policy Column selected in the previous procedure.
- 2** In the Applet property field, double-click the name of the associated applet.
The Applet window appears in the Object List Editor.
- 3** Create a new applet by copying the existing one and append LIC to the name.
- 4** Add a new list column to the applet for the language-independent Code.
 - a** In the Object List Editor, select List object type and then select the List Column object type.
 - b** In the List Column window, create a new record by copying an existing one and set the Field property to “Name”.

To configure the Workflow Policy Column

- 1** Navigate back to the Workflow Policy Column selected in the previous section.
- 2** In the PickList property field, select the new picklist created in [Step 4](#) of the procedure “[To create a LIC picklist for a Workflow Policy Column](#)” on page 4-26.
- 3** In the Source Field property, change the value from Value to Name.
- 4** Compile changes.

To Repick the Values

- 1** Log on using a client connected to the modified repository file.
- 2** From the application level menu, choose View > Site Map > Workflow Administration > Workflow Policies.
- 3** Repick the Values by selecting the conditions and reselecting the appropriate display values.

This will store the language-independent code.

Action Arguments

Before you enable Action Arguments, you must:

- Determine all the business component fields that are enabled for MLOVs.
- Of the fields that are MLOV enabled, identify which ones are referenced by Policy conditions.

For each of the fields that reference a Action Argument, you must complete the following tasks:

- Create a new picklist to display LIC values.
- Create a new applet to display LIC values.
- Configure the Action Arguments to use the new picklist and applet.
- Repick the values for the existing work flow policies.

To create a LIC picklist for a Workflow Policy Program Argument

- 1** In Siebel Tools, navigate to the Workflow Policy Program object type and the Workflow Policy Program that contains the argument that you want to enable for use with MLOVs.
- 2** Select the Workflow Policy Program Argument object type (child of Workflow Policy Program) and then select the Argument you want to enable for use with MLOVs.
- 3** In the PickList property field, click the picklist name.
The PickLists window appears in the Object List Editor.
- 4** Create a new picklist by copying the existing one and append LIC to the name.

To create a new LIC applet for a Workflow Policy Program Argument

- 1** Navigate back to the Workflow Policy Program Argument selected in the previous procedure.
- 2** In the Applet property field, double-click the name of the associated applet.
If no applet exists, you must create one.

The Applet window appears in the Object List Editor.
- 3** Create a new applet by copying the existing one and append LIC to the name.
- 4** Add a new list column to the applet for the language-independent code.
 - a** In the Object List Editor, select List object type and then select the List Column object type.
 - b** In the List Column window, create a new record by copying an existing one and set the Field property to Name.

To Configure the Workflow Policy Program Argument

- 1** Navigate back to the Workflow Policy Program Argument selected in the previous section.
- 2** In the PickList property field, select the new picklist created in [Step 4](#) of the procedure “[To create a LIC picklist for a Workflow Policy Program Argument](#)” on [page 4-27](#).
- 3** In the Source Field property, change the value from Value to Name.
- 4** Compile changes.

To Repick the Values

- 1** Log on using a client connected to the modified repository file.
- 2** From the application level menu, choose View > Site Map > Workflow Administration > Workflow Policies.
- 3** Repick the Values of arguments for existing workflow policies.

Configuring Assignment Manager to Use MLOV-Enabled Fields

Additional configuration is required to enable Assignment Manager to use MLOV-enabled fields. Siebel Assignment Manager compares values in target tables with values in Assignment Manager administration tables to determine whether a particular condition is true. For columns enabled for MLOVs, the value stored in the target table is the Language-Independent Code rather than the Display Value. However, the value in the Assignment Manager administration table is the Display Value. Assignment Manager cannot evaluate a condition by comparing the Language-Independent Code to the Display Value.

To enable Assignment Manager to work with MLOV-enabled columns, you must configure Assignment Manager entities so that they compare the language-independent code in the target table with the language-independent code in the Assignment Manager administration table. You must do this for the following entities:

- Criteria Values
- Criteria Skills
- Workload Rules

Criteria Values and Criteria Skills

Before configuring Criteria Values and Criteria Skills, you must:

- Determine all the business component fields that are enabled for MLOVs
- Of the fields that are MLOV enabled, identify which ones are referenced by Criteria Values or Criteria Skills.

For each of the fields that reference Criteria Values or Criteria Skills (Assignment Attributes), you must set the Translate column to True and define the language-independent code field as the Translate Pick Field.

To configure Assignment Criteria and Skills for MLOVs

- 1** In the Object Explorer, select the Assignment Attribute object type.
- 2** In the Object List Editor, select the Assignment Attribute that you want to work with MLOV enabled fields.
- 3** Set the Translate property for the Assignment Attribute to True.
- 4** Set the Translate Pick Field property to the field name that stores the language-independent code.

Typically the Name field stores the language-independent code.

- 5** Compile changes.

Workload Rules

Before configuring Workload Rules, you must:

- Determine all the business component fields that are enabled for MLOVs.
- Of the fields that are MLOV enabled, identify which ones are referenced by Workload Rules.

For each of the fields that reference Workload rules, you must complete the following tasks:

- Create a new picklist to display LIC values.
- Create a new applet to display LIC values.
- Configure the Workflow Policy Column to use the new picklist and applet.
- Repick the values for existing records.

The detailed steps for completing these tasks are the same as the steps for configuring Workflow Policy Columns covered in [“Policy Conditions” on page 4-25](#).

Configuring Siebel Anywhere for Use with MLOV-Enabled Fields

Siebel Anywhere requires additional configuration to be able to use fields enabled for MLOVs.

To configure Siebel Anywhere for MLOVs

- 1** Open Siebel Tools.
- 2** In the Object Explorer, select the Table object type.
- 3** In the Tables window, query for S_UPG_KIT.
- 4** In the Object Explorer, select Column object type (child of Table).
- 5** In the Column window, select the STATUS column.
- 6** In the Translation Table Name field, click the drop-down list and select the S_LST_OF_VAL.
- 7** Compile an .SRF file.

This enables Siebel Anywhere to use MLOV enabled fields.

After completing this procedure, you can perform standard tasks associated with Siebel Anywhere, such as creating a client repository upgrade kit and distributing to clients.

- Distribute the upgrade kit to Mobile Web Clients.
- Upgrade Siebel Servers with the new .SRF file.

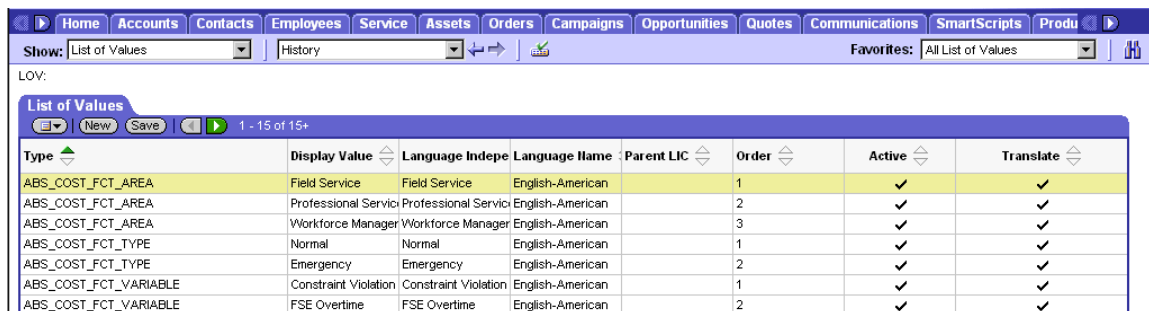
NOTE: Distributing Siebel executable to multilingual remote clients requires additional configuration. See [“Siebel Anywhere” on page 5-10](#).

Administering the Multilingual List of Values

After you have configured your Siebel eBusiness Application for MLOVs, use the List of Values views to administer and monitor LOV entries. You can view LOV data in the List of Values view or in the List of Values Explorer view. Each view shows the same information, but organizes it differently.

List of Values View (Flat View)

The flat view, shown in [Figure 4-2](#) displays all the entries for each list of values type. To access the List of Values view, from the application-level menu choose View > Site Map > Application Administration > List of Values.



The screenshot shows the Siebel application interface with the 'List of Values' view selected. The table below displays the data for the 'List of Values' view.

Type	Display Value	Language Indepe	Language Name	Parent LIC	Order	Active	Translate
ABS_COST_FCT_AREA	Field Service	Field Service	English-American		1	✓	✓
ABS_COST_FCT_AREA	Professional Servi	Professional Servi	English-American		2	✓	✓
ABS_COST_FCT_AREA	Workforce Manager	Workforce Manager	English-American		3	✓	✓
ABS_COST_FCT_TYPE	Normal	Normal	English-American		1	✓	✓
ABS_COST_FCT_TYPE	Emergency	Emergency	English-American		2	✓	✓
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation	English-American		1	✓	✓
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime	English-American		2	✓	✓

Figure 4-2. List of Values Explorer View (Flat View)

List of Values Explorer View

The List of Values Explorer view, shown in Figure 4-3, displays the same data as the flat view, but the list of values entries are displayed in a hierarchical, tree structure. When you open the Types folder, and then the Values folder, all entries for the selected type appear in the List of Values list. To access the List of Values Explorer view, from the application-level menu, choose View > Site Map > Application Administration > List of Values Explorer.

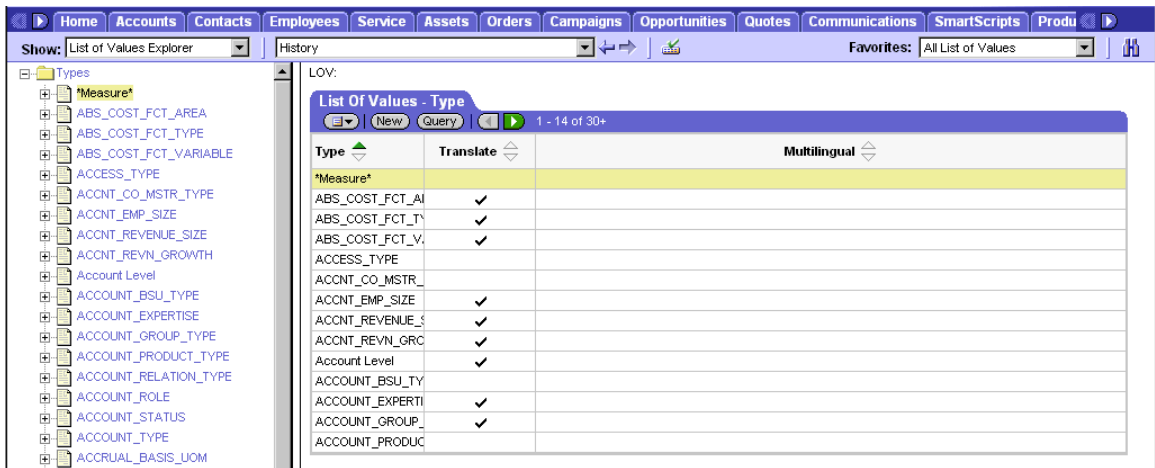


Figure 4-3. List of Values Administration View

Important List of Values Fields

Several list columns in the list of values views help you to administer multilingual LOVs:

- Multilingual.** This field indicates which list of values types have been configured to be multilingual. When you run the MLOV upgrade utility (mlovupgd.exe), it sets this flag for the list of values entries. For more information on the upgrade script, refer to [“Upgrading Existing Data Using the MLOV Upgrade Utility”](#) on page 4-15.

If you add entries after the script has been executed, you must manually update this information to reflect your configuration.

- **Language.** This field indicates for which language the entry is valid. The entries for this picklist come from the Language Administration view. To access this view, from the application-level menu, choose View > Site Map > Application Administration > Languages.

- **Translate.** This field indicates whether the entry's display value can be changed—for instance, translated to another language.

Only the LOV types that are marked as translatable are candidates for multilingual configuration. For any entries added, you must update this information manually to reflect your configuration. Do not change existing Siebel entries. Doing so will not allow the LOV to be translated.

- **Language-Independent Code (or Code in the Explorer view).** This field is the internal code used for a list of values entry. It is stored in the database when MLOV is enabled and referenced by configurations. The language-independent code must be 30 characters or less. The language-independent code cannot be changed.
- **Display Value.** This field is required and holds the text that will appear in picklists. The display value is stored in the database when MLOV is not enabled.

If there are display values for more than one language for a list of values entry, the display value shown is determined by the current active language.

Adding Records

When you add a new list of values record for a LOV type that has been multilingual-enabled, you also need to add records for all supported languages. For example, when adding a new entry for FREIGHT-TERMS type, you need to add values for all supported languages.

If you add a new entry and do not add corresponding additional records for each supported language, the language-independent code will be displayed instead of the display value when a user with one of these languages tries to view the information.

Adding records for all the languages you support is also important for Assignment Manager. For more information, refer to [“Configuration Considerations” on page 4-23](#).

When adding new list of values entries that have a parent-child relationship in a multilingual deployment, set the parent language-independent code to the value of the parent in the same language. For example, if you are adding a subarea for service requests in French, the Parent Language Independent Code must be set to the French value for the area of which the entry is a child.

For more information about adding records to the LOV table, see *Applications Administration Guide, MidMarket Edition*.

Deleting Versus Deactivating Records

As you administer MLOVs, you may find that there are records that you no longer need and would like to make inactive. If you delete a MLOV record, then records in other tables that have already been entered using that list of values record will no longer display correctly. The display value in the list of values entries is used to display the language-specific text.

Instead of deleting a record, inactivate it. Inactive values that have already been used and are referenced in other tables in the database will still display correctly. Inactive records are not included in any picklists, however, and are ignored by EIM when it performs validation against LOVs.

The Active list column in the List of Values Administration view is checked by default. To deactivate a record, remove the check mark.

If you try to delete a record, you will get a message asking you if you really want to delete the record, or just deactivate it. If you choose Inactivate from the dialog box, the check mark in the Active list column is removed.

If you do delete an LOV record, any language-independent codes in the target columns referring to the deleted record will then display the language-independent code. Searching and sorting will not function correctly on these values.

Constraints

LOVs are constrained in the following ways:

- You cannot have children of a list of values entry active when the parent list of values entry is inactive.
- Hierarchical LOVs do not support MLOVs.
- LOV_TYPE should not be enabled for multilingual list of values. It has a single-language entry only.

Dynamic Toggle Applets

Dynamic Toggle Applets allow you to display data differently for different locales in a single database. For example, you are deploying an English application in two locales, the United States and Canada and the expense report applet needs to display mileage in either miles or kilometers, depending on the locale of the user. Dynamic Toggle Applets can be configured to display miles for U.S. users and kilometers for Canadian users.

Basically, you create one Applet Toggle object definition for each locale-specific version of the applet. The parent Applet (of these Applet Toggles) is the default—the one that appears when the test condition (the locale parameter, in this case) has no match.

To configure Applet Toggle objects in Siebel Tools

- 1** Create an additional column in the business component used by the set of applets (S_ORG_EXT for Account, for example) to hold the locale parameter.
- 2** Create a new field in the business component to map the new column.
- 3** Create a locale-specific version of the applet for each locale requiring separate data entry capabilities.

Include the new Locale field (as a picklist) in each of the applets so there is a way to collect this data for each record. Also, create a default version of the applet to use if no locale is specified or the specified locale does not have its own applet.

- 4** Create an Applet Toggle child object definition (of the one designated as the default) for each locale-specific applet.

Use the Applet, Auto Toggle Field, and Auto Toggle Value properties in each applet toggle to specify the applet to navigate to, the name of the locale field, and the name of the locale that causes this applet to appear when this locale name appears in the data.

See [“MLOV Configuration and Coding Guidelines” on page 4-23](#) for more information about configuring fields for Applet Toggles.

- 5** Compile, test, and deploy.

For more information, see the section on applet toggle behavior in *Siebel Tools Reference, MidMarket Edition*.

Global Time Zone Support

Global deployments typically span multiple geographies 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. Global Time Zone uses the Universal Time Zone (UTC) standard. UTC is equivalent to Greenwich Mean Time, but without daylight savings time. It allows you to store date and time data in a common format and provides the following key benefits:

- Users can view dates and times in their local time zone regardless of where work items were created.
- Users in different time zones can jointly handle team-based activities, such as managing service requests.
- Customer needs can be met by sales or service assistance from agents based in various locations and time zones.

Data Conversion to and from UTC

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

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

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

NOTE: Windows regional settings on the application server or client machines do not affect Global Time Zone.

Enabling UTC

Although UTC is optional, Siebel Systems recommends enabling UTC. Enabling UTC requires the following steps:

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

See [“Setting UTC System Preferences”](#) on page 4-38.

For upgrades, additional conversion steps are required. See [“Enabling Global Time Zone in an Upgraded Environment”](#) on page 4-41 for more information.

Setting the Database to UTC

To enable UTC you need to set the operating system of your database server machine to UTC or its equivalent. The method of setting the system time to UTC on your database server machine(s) will vary depending on the operating system being used. Many operating systems have a time zone setting called UTC or something similar. Others have time zone options described as Greenwich Mean Time without daylight savings time—this is the equivalent of UTC.

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

Setting UTC System Preferences

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

- **Universal Time Coordinated.** Turns Global Time Zone support on for the entire system. The UTC system preference should be set to the following:
 - For new installations, set the UTC system preference to TRUE.
 - For upgrades, set the UTC system preference to FALSE.

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

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

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

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

See [“Enabling Global Time Zone in an Upgraded Environment” on page 4-41](#) for more information about upgrading historical data to UTC.

Configuring Fields for Global Time Zone Support

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

You enable fields for Global Time Zone support by setting the Physical Type property for the column and the Type property of the field to which the column is mapped.

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

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

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

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

Administering Global Time Zone Support

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

- [“Modifying Time Zones” on page 4-40.](#)
- [“Maintaining Translations of Time Zones” on page 4-41.](#)

Modifying Time Zones

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

To Modify Time Zone Settings

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

Some fields are described in the following table.

Field	Description
Name	Name of time zone.
Standard Abbreviation	The abbreviation for the time zone.
UTC Offset	Offset, in minutes, from UTC when daylight savings is not in effect.
DST Abbreviation	The abbreviations for daylight savings time.
DST Bias	Incremental DST offset, in minutes, from UTC when DST is not in effect.
DST Start Ordinal	Part of the rule that determines when DST starts. For example, if the rule is <i>the first Sunday in April</i> , <i>First</i> is defined in this field.

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

Maintaining Translations of Time Zones

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

To maintain translations of time zones

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

The Time Zone Translation form appears.

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

Enabling Global Time Zone in an Upgraded Environment

When enabling an upgraded Siebel environment for global time zone, you should convert your historical data to make all existing date/time values consistent with global time zone logic. Global time zone stores date/time values in the database adjusted to UTC time. For more information about how global time zone processes data/time data, see [“Data Conversion to and from UTC” on page 4-37](#).

Although enabling your environment for global time zone is optional, Siebel Systems recommends that you perform the following procedure immediately after you upgrade to Siebel 7.

Converting historical data into a global time zone enabled environment requires you to perform the following tasks:

- 1** Upgrade your development and production environments to Siebel 7. See *Upgrade Guide for Microsoft Windows, MidMarket Edition* for information about upgrading to Siebel 7.
- 2** Prepare your data for conversion to global time zone. See [“Preparing Your Data for Conversion to Global Time Zone.”](#)
- 3** Convert your historical data, using the Siebel Software Configuration Utility. See [“Converting Your Data to be Global Time Zone-Enabled”](#) on page 4-48.
- 4** Review the UTC log file for errors. See [“Reviewing the UTC Conversion Log Files”](#) on page 4-52.
- 5** Turn on global time zone. See [“Turning On Global Time Zone”](#) on page 4-53.

Preparing Your Data for Conversion to Global Time Zone

Before you convert your data to a global time zone-enabled environment, you need to prepare your historical data.

- Configure fields and columns (if necessary) for global time zone.
- Edit the `driver_utc.ucf` file for UTC conversion.
- Set the default time zone preference for each of your users.
- Save a report of your user time zones.
- Partition large tables to assure sufficient log space.
- Allocate maximum database transaction log space.

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

Configure Fields for Global Time Zone

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

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

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

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

Edit the driver_utc.ucf File for UTC Conversion

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

An example from the `driver_utc.ucf` file appears below:

```
[File Execute Entry 7]

Type = FileExecute

File Name = $SiebelRoot\bin\utcupgd

Check Return Code = 1

Return Code Compliance = 0

16 Bit App = 0

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

Number of 10 Second Wait Loops = 2000

Prompt User For Status = 0

Parallelizable Item = 0

Title Message Num = 0

Estimated Disk Space = 0

Backup Db = 0
```

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

- 1** Create a new input file in the `driver_utc.ucf` file by copying a complete step from the `driver_utc.ucf` file and pasting the copied step immediately after the step which you duplicated.
- 2** In the new step, change the input file parameter (which follows `/j` in the command line) to the name of your new input file.

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

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

to

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

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

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

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

to

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

- 4** Repeat [Step 1](#) through [Step 3](#) for each new input file.
- 5** Renumber the file execute entry numbers for your new step and for each subsequent step, in order.

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

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

to

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

and so on, for each subsequent step.

Set Time Zones for Users

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

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

Save a Report of User Time Zones

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

Partitioning Prerequisite for the Oracle Database Platform

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

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

Partition Large Tables to Prevent Insufficient Database Log Space

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

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

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

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

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

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

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

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

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

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

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

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

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

Allocate Maximum Log Space

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

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

Converting Your Data to be Global Time Zone-Enabled

To update existing date/time data in your data tables to be global time zone-enabled, you need to run the Universal Time Code conversion utility from within the Siebel Software Configuration Utility. This utility will help you define the required parameters for UTC conversion, and set the universal time coordinated (UTC) system preference to TRUE.

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

For example:

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

where:

the first column value (in this example, `CREATED`) indicates the column name

the second column value (in this example, `CREATED_BY`) indicates the conversion method.

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

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

Running the UTC Conversion Utility

The following procedure will convert your date/time data to global time zone enabled.

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

- 1 Launch the Siebel Software Configuration Utility by selecting Start > Programs > Siebel Servers 7.0 > Configure DB Server.

The Siebel Software Configuration Utility appears.

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

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

See [Table 4-5 on page 4-51](#) 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/time values to their UTC equivalent. After successfully converting the UTC-enabled date/time fields, the conversion utility sets the value for the Universal Time Coordinated system preference to TRUE.

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

- 3 Review the log file generated by the UTC conversion, and resolve errors as necessary. See [“Reviewing the UTC Conversion Log Files” on page 4-52](#).

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

To manually rerun the UTC conversion utility, type the following command at the Windows DOS command prompt from the `SIEBEL_ROOT\siebsrvr_root\BIN` directory:

```
siebupg.exe /m master_utc.ucf
```

UTC Conversion Parameters

Table 4-5. Siebel Software Configuration Utility (1 of 2)

At this prompt	Enter or select the following
Siebel Enterprise Parameters: Gateway Server Address	<ul style="list-style-type: none"> ■ Gateway Server Address (This is typically the machine name.) ■ Enterprise Server Address
Installation and Configuration Parameters: Siebel Server Directory	<ul style="list-style-type: none"> ■ Siebel Server Directory (This is the <i>SIEBSRVR_ROOT</i> directory, for example, <i>D:\sea7xx\siebsrvr.</i>)
Installation and Configuration Parameters: Siebel Database Server Directory	<ul style="list-style-type: none"> ■ Database Server Directory
Database Server Options: Siebel Database Operation	<ul style="list-style-type: none"> ■ Run Database Utilities
Database Utilities: Database Utility Selection	<ul style="list-style-type: none"> ■ Universal Time Code Conversion
Installation and Configuration Parameters: Language Selection	<ul style="list-style-type: none"> ■ Base language of your Siebel application (This is the primary language of your prior environment.)
Installation and Configuration Parameters: RDBMS Platform	<ul style="list-style-type: none"> ■ RDBMS Platform
Installation and Configuration Parameters: ODBC Data Source Name	<ul style="list-style-type: none"> ■ ODBC data source¹ (This was created automatically by the Siebel Server installation, using the format <i>SiebSrvr_EnterpriseName.</i>)
Installation and Configuration Parameters: Database User Name	<ul style="list-style-type: none"> ■ Database User Name (User name of the Siebel administrator, for example, <i>sadmin</i>) ■ Database Password
Installation and Configuration Parameters: Table Owner	<ul style="list-style-type: none"> ■ Table Owner Name (Note for Microsoft SQL: this is the login for the owner of the database, not necessarily the default owner of the database in DBO.) ■ Table Owner Password

Table 4-5. Siebel Software Configuration Utility (2 of 2)

At this prompt	Enter or select the following
UTC Parameters: Repository Name	■ Repository Name
Configuration Parameter Review	Review the parameters you have defined and then click Finish.

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

Reviewing the UTC Conversion Log Files

After the UTC conversion utility runs, if errors are encountered, the utility records those errors to log files. The log files are located in the `siebsrvr\LOG` directory. The default log files are listed below. You may have additional log files if you edited the input file. Carefully review the log files for errors.

```
utc_drop_temp_tab.log
null_timezone.log
utc_insert_to_tmp_tab.log
utc_create_tmp_ind.log
utc_run_stats.log
s_camp_con_00.log
s_camp_con_01.log
s_camp_con_02.log
s_camp_con_03.log
s_communication_00.log
s_communication_01.log
s_communication_02.log
s_communication_03.log
s_evt_act_00.log
s_evt_act_01.log
s_evt_act_02.log
s_evt_act_03.log
s_evt_act_04.log
s_evt_act_05.log
s_contact.log
s_org_ext.log
s_srv_req.log
utc_columns.log
denorm.log
utc_drop_temp_tab2.log
```

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

Manually Launching the UTC Conversion Utility

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

To manually launch the utility, type the following command from the `SIEBEL_ROOT\siebsrvr_root\BIN` directory:

```
siebupg.exe /m master_utc.ucf
```

Turning On Global Time Zone

After you have confirmed that your UTC conversion was successful, turn on UTC by performing the following steps:

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

See [“Enabling UTC” on page 4-38](#) for more information.

Additional Setup and Configuration

5

About This Chapter	5-2
Siebel eMail Response	5-2
Siebel eMail Response Overview	5-3
Planning eMail Response Deployment	5-5
Setup for eMail Response International Deployment	5-6
Limitations	5-9
Siebel Anywhere	5-10
Siebel Personalization	5-12
Siebel Reports Server	5-13

About This Chapter

This chapter covers additional application-specific configuration and setup topics for multilingual deployments. Also see *Release Notes, MidMarket Edition* for more information.

Siebel eMail Response

Siebel eMail Response can be deployed in a multilingual environment and configured to handle email in multiple languages. eMail Response can receive messages in multiple languages and respond to messages in the sender's language.

This section describes special considerations for deploying Siebel eMail Response in multilingual environments.

For general information about Siebel eMail Response, see *Siebel eMail Response Administration Guide, MidMarket Edition*.

Siebel eMail Response Overview

In a multilingual deployment of eMail Response, you will need to set up mailboxes for each of the languages supported by your organization. For example, if your organization plans to support French and German, then you need to set up two mailboxes. You might name the French mailbox `fra_support@siebel.com`, and the German mailbox `deu_support@siebel.com`. Users would send messages to one of the two addresses depending on their language. [Figure 5-1](#) shows an overview of the eMail Response architecture.

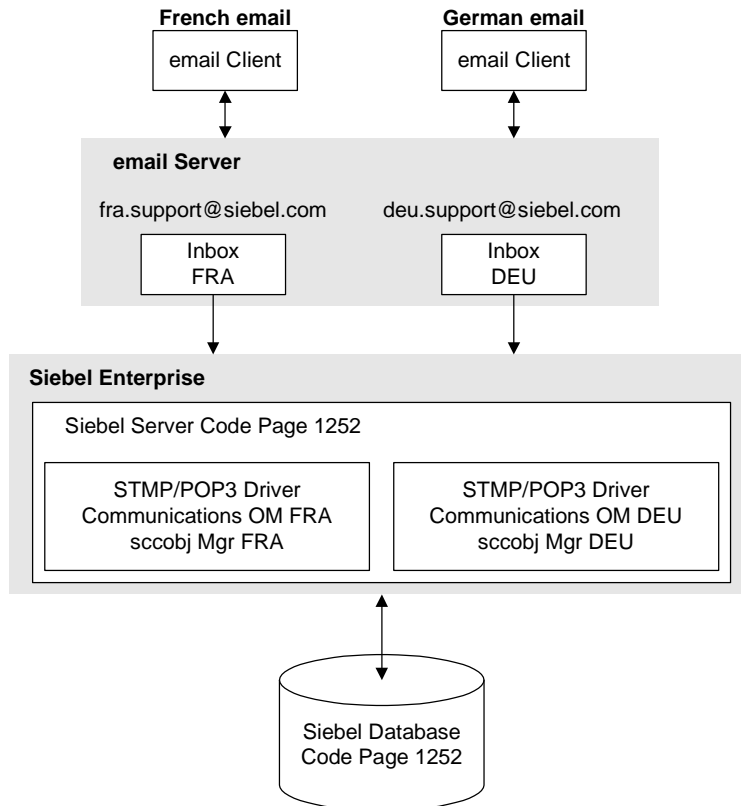


Figure 5-1. Siebel eMail Response Architecture

In a multilingual environment, Siebel eMail Response does the following:

- Monitors mailboxes set up for each language that your organization supports.
- Creates a record in the Siebel database that stores the content of the email along with the language of the email message. The original email message is also stored as an attachment to the record.
- Sends an acknowledgement message using the original language of the message. For example, if a user sends a message in French, an acknowledgement message will be sent to the user in French.
- Allows an agent to compose and send messages using response templates in a preferred language.
- Checks spelling for the most common international languages.

Replying To Messages

Agents review messages using a Siebel client and compose and send response messages using response templates. The list of available templates consists of all templates in all languages. It is not filtered by the language the agent is handling.

NOTE: Templates and Solutions should be translated into the language supported by your organization during your planning stage.

About Communication Server

Siebel eMail Response sends the message to the Communication Outbound Manager. Depending on the number of languages supported by the organization, an equal number of Outbound Managers will have to be defined to handle outgoing messages. For example, if your organization plans to support French and German languages, two Outbound Managers will have to be defined. Each one will handle outgoing messages for each of the languages respectively. For messages that include data from multilingual list of values, Outbound Manager will populate the outgoing message with the data in the language specified by the .SRF of the Outbound Manager.

See [“Configuring Communication Server for Outbound Messages” on page 5-7.](#)

Planning eMail Response Deployment

Before deploying eMail Response in an international environment, analyze your organization's requirements.

- Does any of the groups in your organization receive messages in languages other than English? If so, do they have an email address set up to receive international messages? How are international email messages handled currently?
- How many languages does your organization plan to support?
- If your organization plans to support three languages (French, German and Italian), then you need to create three mailboxes and three response groups. Each response group will support a language-specific mailbox.

NOTE: The language field column is not displayed in the activities applet (Comm Inbound Item List Applet). To see the language of the activity, users will have to select the More Info tab and click on the Show More icon to see the Language column.

- What templates have you decided to use in eMail response replies? If your organization is planning to support multiple languages and decides to add new templates, the template text will have to be translated to the languages that the customer plans to support.

See [“Creating Templates” on page 5-9](#).

Setup for eMail Response International Deployment

Prerequisites Tasks

If your organization plans to support international languages, make sure to install the DB language packs for each of the languages supported. The language pack will install the templates used in eMail Response in a specific language.

For more information on Database language pack installation, refer to *Siebel Server Installation Guide, MidMarket Edition*.

Installing Language Packs

When you install Database Language packs, the installation will also install the templates associated with eMail Response. You can create additional templates to meet their business requirements following best practices guidelines. Setup [“Creating Templates” on page 5-9](#).

For example, when the customer installs the language packs, with French as the base language and English and German as additional languages, the installer will install the following templates:

- Base language templates are installed as: `templ_name` and `templ_name_fra`.
- Additional templates are installed as: `templ_name_deu`, `templ_name_enu`, and so on (based on the selected languages).

The templates will have an English name with the suffix of the language. The contents of the templates will be translated to the language selected. For example, the auto acknowledgement template will be eMail Response - Auto Acknowledgment FRA and the actual content of the template will be in French.

Setting Up Mailboxes

You will need to set up different mailboxes for each language that you plan to support.

In your email server, create one mailbox per language. Each of these mailboxes will be used to handle messages in different languages. If the application will support French and German messages, then set up two mailboxes (for example, `fra_support@yourcompany.com` and `deu_support@yourcompany.com`). For French messages, users will have to send their messages to `fra_support@yourcompany.com`. For German messages, users will have to send their messages to `deu_support@yourcompany.com`.

Configuring Communication Server for Outbound Messages

You need to have a Communication Outbound Manager Server Component for each language you support. The components are created after you install the language packs by running a script that creates all necessary server component for each supported language and automatically sets the appropriate parameters. Running this script is a step in the installation procedures documented in *Siebel Server Installation Guide, MidMarket Edition*.

Creating a Communication Driver Profile

Siebel communication profiles map to individual accounts or mailboxes. It is recommended to create a unique profile with the same name as the email address. For example, if you have a mailbox named `fra_support@yourcompany.com`, create a profile with the same name.

Configuring Response Groups

For email messages in multiple languages, you will need to create a response group for each language and associate profiles to it. For example, if you plan to receive French messages, create a Response Group named “FRA Response Group” and associate the profile `fra_support@yourcompany.com` to the response group.

Each response group will monitor a particular language mailbox. You will need to create one response group per language. For example, the Communication Outbound Manager FRA will monitor the profiles associated with FRA Response Group.

Each response group will have different input argument settings. For each response group, you need to pass these additional input arguments:

- Language-dependent templates used by the workflow
- Name of the communication outbound manager to invoke
- Default language of the email

Below is an example of the input arguments that need to be provided when using the eMail Response - Process Message workflow process:

- `CommOutboundMgrName = CommOutboundMgr_FRA`
- `AckTemplateName = eMail Response - Send Acknowledgement FRA`
- `MsgDefaultLanguageCode = FRA`

The language-dependent Communication Outbound Manager will be passed to the Workflow Process. When an agent using eMail Response replies to an email message from the Siebel client, the application will use the Communication Outbound Manager component definition specified here.

An acknowledgement template needs to be specified as an input argument to the Response Group. This input argument will be passed to the Workflow Process eMail Response - Process Message so the workflow will know which acknowledgement template to use. For example, when a user sends an email in French, the user will receive an auto acknowledge message in French.

You need to specify each template that is needed by the workflow process as an input argument of the response group. For example, the eMail Response - Process Service Request uses five templates. You will need to pass all five French templates. If you have modified the workflow to include additional templates, you need to include them as input arguments as well.

MsgDefaultLanguageCode is the language that the workflow process will use to set the language of the activity record created. For example, when a user sends an email to the `fra_support@yourcompany.com`, the application creates an activity and the language of the activity is FRA.

Below is an example of the input arguments that you would need to provide for the eMail Response - Process Service Request workflow process:

- CommOutboundMgrName - CommOutboundMgr_FRA
- HelpTemplate = eMail Response - SR Help _FRA
- StatusTemplate = eMail Response - SR Status_FRA
- QueryTemplate = eMail Response - SR Query_FRA
- UpdateTemplate = eMail Response - SR Update_FRA
- SubmitTemplate = eMail Response - SR Submit_FRA
- MsgDefaultLanguageCode = FRA

For more information on how to create profiles and response groups, see *Siebel eMail Response Administration Guide, MidMarket Edition*.

Creating Templates

When creating templates in multilingual deployments, follow the convention of naming the templates in English with a language-code suffix (for example, `MyNewTemplate_ITA`). This will help maintain consistency across the application. It is the naming convention used by Siebel Systems for translating the standard templates used by eMail Response into the languages Siebel supports. It is recommended that you follow the same naming convention.

Additionally, if you create templates, additional logic must be added to the workflow so that they can be invoked from the workflow process and so that Workflow will be able to detect the language of the template.

For instruction on how to create templates, see *Siebel eMail Response Administration Guide, MidMarket Edition*.

Limitations

- The email address field supports only ASCII characters. Any characters that are not ASCII are not supported in the email address field and they will be removed. If incoming email addresses contain non-ASCII characters, those characters will also be removed before a response is sent.
- One Response Group cannot handle multiple languages.
- Siebel eMail Response does not perform language detection. If an email message is sent to the wrong mailbox, the language of the activity record will be the default language that is specified in the input argument of the response group.
- Messages that are in languages that do not fall in the Windows code page number 1252 or code page name 8859-1 will be processed as undisplayable characters and the original raw email message will be attached to the activity record.
- When responding the email messages, the RE: and FW: will be in English and not translated to the various languages supported.

For more information about supported languages, see *Siebel System Requirements and Supported Platforms, MidMarket Edition*.

Siebel Anywhere

In order for Siebel Anywhere to distribute Siebel executables to Siebel Mobile and Dedicated Web Clients with multiple language packs, you need to follow the procedure in the following section.

NOTE: This procedure assumes you are familiar with using Siebel Anywhere to create upgrade kits. See *Siebel Anywhere Administration Guide, MidMarket Edition* for detailed information about creating upgrade kits.

To distribute Siebel executables in multilingual environments

- 1 Choose Start > Programs > Siebel 7.0 > Packager and use the Siebel Packager Wizard to create a package for the base language pack and each additional language pack that you need to distribute to Mobile and Dedicated Web Clients.

Name each package something unique that identifies the language. For example, package_base.

The packages for base language and each additional language are stored in different directories.

- The package for base is stored in:
`siebel_client_root\packager\temp\package_name`
- The package for additional languages are stored in:
`siebel_client_root\packager\temp\language\package_name`

- 2 From the application-level menu in the Siebel application choose View > Site Map > Siebel Anywhere Administration > Upgrade Kit, and then create upgrade kits for each of the packages created in [Step 1](#).

NOTE: Uninstall previous versions of Siebel for the Base package only. Do not uninstall previous versions for packages of other language packs. You set the Uninstall Previous Versions option in the Add Files page of the Upgrade Kit Wizard.

- 3** From the application-level menu, choose View > Site Map > Siebel Anywhere Administration > Configurations and create a configuration for the combination of languages you want to distribute.
- 4** From the application-level menu, choose View > Site Map > Siebel Anywhere Administration > Employees and add records for the employees that will use the configuration created in [Step 3](#).
- 5** From the application-level menu, choose View > Site Map > Siebel Anywhere Administration > Configurations and click Distribute to distribute the upgrade kit to this configuration.

When employees defined for the configuration log on or synchronize with the server, the upgrade kits are downloaded and the upgrade wizard applies the kits.

Siebel Personalization

This section describes guidelines for configuring personalization rules in a multilingual environment.

NOTE: This section assumes you are familiar with Siebel Personalization and understand how to create personalization rules. If you need more information about these topics, see *Personalization Administration Guide, MidMarket Edition*.

Multilingual deployments require a language-specific version of each personalization rule. For the personalization rules that you create, you need to also create language-specific versions. For example, suppose your organization is deploying Siebel in English, German, and French. And suppose you want to create a generic personalization rule called “Welcome” that is applied to the Siebel Sales Application ruleset.

```
"<div class='Welcome'><b>Welcome back " + GetProfileAttr ("Full Name")
```

You would need to create three language-specific personalization rules, one for each supported language. For example, the rule for English could be named Welcome ENU and would be defined as:

```
"<div class='Welcome'><b>Welcome back " + GetProfileAttr ("Full Name")
```

It would contain the Conditional Expression:

```
Language() = 'ENU'  
And Sequence number 1.
```

The French rule could be called *Welcome FRA* and would be defined as:

```
"<div class='Welcome'><b>Bon Jour " + GetProfileAttr ("Full Name")
```

It would contain the following 'Conditional Expression'

```
Language() = 'FRA'
```

```
And sequence number 2.
```

And another rule may be created for German, similar to the French and English rules. It would have a sequence number of 3.

If the rule with sequence 1 evaluates to FALSE, sequence 2 is evaluated, and so on.

Siebel Reports Server

In order for Reports Server to correctly interpret all data that is to be printed, the settings of locale-dependent parameters in Windows Regional Settings should match the equivalent parameters in the Siebel Object Manager.

For example suppose you are deploying Siebel Reports Server in Italy. The Siebel Object Manager and the Reports Server hosts are running under the ITA locale. The Siebel application uses the same thousands and decimal separators for numbers and currency (a comma for thousands and a dot for decimals). However, at the Windows operating system level, the separators are different. You would need to modify the thousand and decimal separators for number and currency (in the ITA locale) on the Reports Server machine to be in sync with the sThousand and sDecimal parameters in the object manager. More specifically, for ITA sThousand and sDecimal parameters are set to dot and comma respectively on Object Manager. Therefore, in Windows Regional Settings on the Reports Server, you would need to set the digits grouping symbol and decimal symbol properties to dot and comma for both numbers and currency.

Verify that the locale-specific parameters in Regional Settings match the corresponding parameters in Siebel Object Manager. See [Table 5-1](#) for a list of these parameters.

For more information about setting parameters for object managers, see *Siebel Server Administration Guide, MidMarket Edition*.

For more information about Reports Server, see *Siebel Reports Administration Guide, MidMarket Edition*.

Table 5-1. Windows Regional Settings and Object Manager Parameters (1 of 2)

Tab in Regional Settings Dialog Box	Windows Parameter	Siebel Object Manager Parameter	Description
Number	Decimal symbol	sDecimal	Character used to separate the integer part from the fractional part of a number and currency.
Number	Digit grouping symbol	sThousand	Symbol used to separate thousands in numbers and currencies with more than three digits.
Number	Number of digits after decimal	iDigits	Value defining the number of decimal digits that should be used in a number.
Number	Display leading zeros	iLzero	iLzero = 0 -> Display without leading zero siLzero = 1 -> Display with leading zeros
Currency	Decimal symbol	sDecimal	Character used to separate the integer part from the fractional part of a number and currency.
Currency	Digit grouping symbol	sThousand	
		iCurrency	iCurrency = 0 -> no separation between currency symbol prefix and number iCurrency = 1 -> no separation between currency symbol suffix and number iCurrency = 2 -> one character separation between currency symbol prefix and number iCurrency = 3 -> one character separation between currency symbol suffix and number
Currency	Negative currency format	iNegCurr	iNegCurr = 0 -> (\$1.1) iNegCurr = 1 -> -\$1.1 iNegCurr = 2 -> \$-1.1 iNegCurr = 3 -> \$1.1- iNegCurr = 4 -> (1.1\$) iNegCurr = 5 -> -1.1\$ iNegCurr = 6 -> 1.1-\$ iNegCurr = 7 -> 1.1\$- iNegCurr = 8 -> -1.1 \$ iNegCurr = 9 -> -\$ 1.1 iNegCurr = 10 -> 1.1 \$- iNegCurr = 11 -> \$ 1.1- iNegCurr = 12 -> \$ -1.1 iNegCurr = 13 -> 1.1- \$ iNegCurr = 14 -> (\$ 1.1) iNegCurr = 15 -> (1.1 \$)

Table 5-1. Windows Regional Settings and Object Manager Parameters (2 of 2)

Tab in Regional Settings Dialog Box	Windows Parameter	Siebel Object Manager Parameter	Description
Parameter	Time separator	sTime	Time separator. This character is displayed between hours and minutes, and between minutes and seconds.
Parameter	Time style	iTime	iTime = 0 -> 12-hour clock iTime = 1 -> 24-hour clock.
Parameter	Time style	iTLZero	Specifies whether or not the hours should have a leading zero. iTLZero = 0 -> without leading zero iTLZero = 1 -> with leading zero
Parameter	AM symbol	s1159	This setting contains the trailing string used for times between 00:00 and 11:59.
Parameter	PM symbol	s2359	Trailing string for times between 12:00 and 23:59, when in 12-hour clock format.
Date	Date separator	sDate	Character used to separate the integer part from the fractional part of a number and currency.
Date	Short date style	sShortDate	Symbol used to separate thousands in numbers and currencies with more than three digits
Date	Long date style	sLongDate	Value defining the number of decimal digits that should be used in a number.

Regional Settings in the Operating System	6-2
Setting Date Formats	6-3
About the Active Language	6-4
Online Help	6-4
Choosing Fonts	6-5
Third-Party Software	6-6
Application-Wide Data	6-6
Sales Cycles	6-6
Correspondence Templates	6-6
Literature	6-6
SmartScript and Campaigns	6-7

Regional Settings in the Operating System

Regional settings in the operating system control the way some data in the user interface is displayed. On the server machine, regional settings are used when locale-specific parameters are not defined for an object manager. On Mobile Web Client and Dedicated Web Client machines, regional settings of the operating system are always used.

Data affected by the operating system includes:

- Number
- Currency
- Time
- Date
- Phone number country codes (but not the formatting of the phone number)

For more information on phone number formatting, see *Applications Administration Guide, MidMarket Edition*.

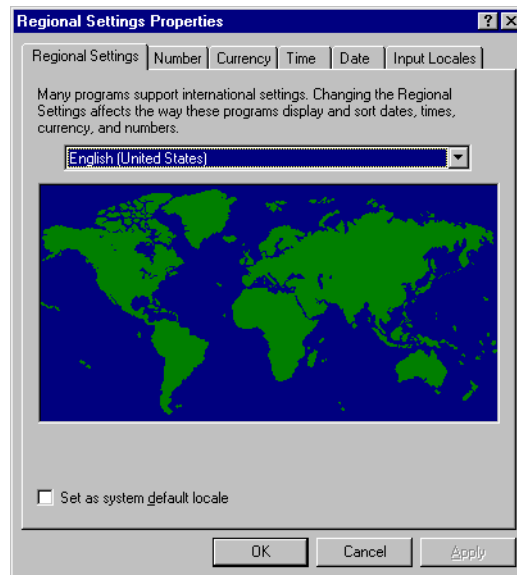
In Windows, these settings are defined by the Control Panel Regional Settings.

NOTE: The keyboard input is also set in the Regional Settings Properties dialog box.

To view regional settings for Windows NT

- 1 Choose Start > Settings > Control Panel.
- 2 Double-click the Regional Settings button.

The Regional Settings Properties dialog appears.



- 3 Make your selection, and click OK.

Setting Date Formats

Setting date formats can be different depending on which version of Windows you are using.

About the Active Language

The active language for Siebel Clients is determined by the following:

Web Clients. The active language is determined by the language parameter of the object manager to which the Web client is connected.

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

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

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

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

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

Online Help

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

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

Choosing Fonts

You can change which fonts are used in the Siebel Client by making changes in the configuration file. The font you select is especially important for global deployments because:

- The font must include the characters (letters) for the language being used.

Not every font can show every character. For more information about code pages, see [Chapter 2, “Code Pages and Collation.”](#)

- String length expansion is a consideration for some languages.

For example, German text is 30% larger, on average, than the equivalent English text. By using a smaller font, the localized strings can be used without the need to reconfigure the user interface.

[Table 6-1](#) shows the recommended fonts for localized versions of your Siebel application.

Table 6-1. Recommended Fonts for Localized Siebel Applications

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

Third-Party Software

The versions of third-party software that you need for your localized Siebel application are described in *Release Notes, MidMarket Edition* and *Siebel System Requirements and Supported Platforms, MidMarket Edition*.

As with your operating system, third-party software must support the same code page as your Siebel application.

Application-Wide Data

Sales Cycles

You can only have sales cycles in one language.

Correspondence Templates

Each correspondence template is in a specific language. These templates have been localized. For information on creating new correspondence templates, in any language, see *Applications Administration Guide, MidMarket Edition*.

Each user can specify a default correspondence template; users can choose the appropriate template for their locale.

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

Literature

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

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

SmartScript and Campaigns

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

- A SmartScript can be run in multiple languages.
- The same logic can be used in a SmartScript, but with a localized message displayed.
- The target language can be manually selected when a script is started. An example of this is shown in [Figure 6-1](#).
- The target language can be set automatically when a script is started.

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

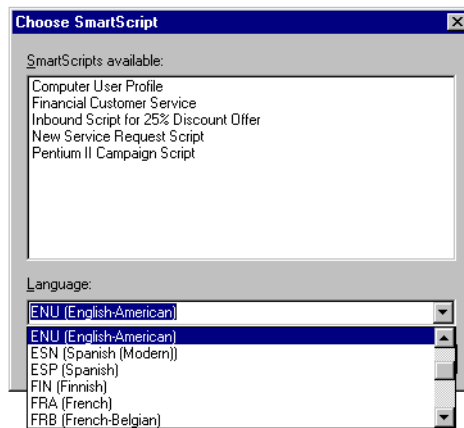


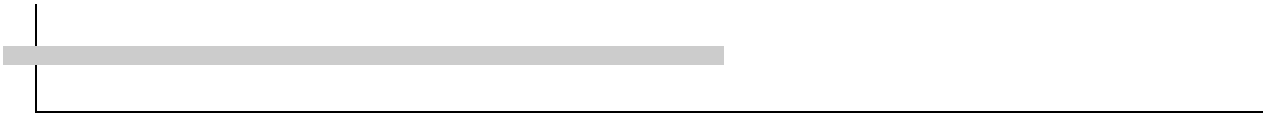
Figure 6-1. Dynamic SmartScript Language Selection - the Choose SmartScript Dialog Box

Localizing Siebel eBusiness Applications

Part 3

Chapter 7. Localization

Chapter 8. Siebel Tools and Multilingual Development



About This Chapter	7-2
The Localization Process	7-2
What Can You Localize?	7-3
Using the Locale Management Utility	7-4
Finding Untranslated Text Strings	7-4
Exporting Strings and Other Locale-Specific Attributes	7-5
Importing Strings and Other Locale-Specific Attributes	7-6
Identifying Modified Objects	7-8

About This Chapter

This chapter covers basic information about localization, including the tasks and functionality related to the Locale Management utility.

The Localization Process

Localizing an application includes translating the user interface and modifying other attributes to meet locale-specific requirements. It typically includes the following steps:

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

The LMU exports the objects to a flat file. For more information, see [“Exporting Strings and Other Locale-Specific Attributes” on page 7-5](#).

- 3 Translate strings by modifying the flat file directly or by importing the file into a separate localization development environment, modifying the locale-specific attributes, and then exporting the localization result to another flat file.
- 4 Import modified string and/or modified object definitions into the repository using the LMU.

For more information, see [“Importing Strings and Other Locale-Specific Attributes” on page 7-6](#).

- 5 If necessary, search for strings or locale-specific attributes that have been modified since the last export, and update the string translation or attributes localization for these changed objects.

For more information, see [“Identifying Modified Objects” on page 7-8](#).

- 6 Compile the modified projects into a repository file (`.srf`).
- 7 Distribute the repository file to the appropriate Siebel Servers, Mobile Web Clients, and Dedicated Web Clients.

What Can You Localize?

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

Using the Locale Management Utility

The Locale Management Utility (LMU) in Siebel Tools helps you manage the process of localizing text strings, such as applet titles, field labels, and tab labels. Localizing text strings includes exporting the strings to a file, which is then translated and imported back into the repository. The LMU provides the export and import tools to do this.

You use the Locale Management Utility to:

- Find strings that have not been translated.
- Export strings to a file for localization.
- Import strings from a file back into the repository.
- Search for strings that have changed since the last export.
- Compare object definitions in the repository to the object definitions stored in the export file.

Finding Untranslated Text Strings

You can use the Locale Management Utility to find text strings in the repository that have not been translated.

NOTE: The LMU performs search and comparison functions at the object level, not the attribute level. Therefore, if a locale object contains multiple string attributes, the search function will return all strings contained in the locale object, even if only one of them is untranslated.

To find and export untranslated strings

- 1** Choose Tools > Utilities > Locale Management.

The Locale Management Utility appears.

- 2** Select the source language and the target language.
- 3** Select the application that you want to localize.

- 4 Select the Untranslated Strings tab.
- 5 Select the Redo checkbox if you want to see strings that have been marked as redo.

The Redo flag is marked when a record in the repository has been changed since the last time export occurred and therefore may need to be retranslated.

See [“Identifying Modified Objects” on page 7-8](#) for more information about Redo.
- 6 Click Start.

The Locale Management Utility searches through the string attributes of objects in the selected application and displays the ones that have not been translated.
- 7 After you find untranslated strings you can perform the following tasks:
 - Find the views that strings belong to by selecting a string, and then clicking the Find View button.
 - Go to the parent object of the string in the Object Explorer by selecting a string, and then clicking the Go To button.
 - Export the untranslated strings to a text file by selecting a string and then clicking the Export button.

Exporting Strings and Other Locale-Specific Attributes

You use the Locale Management Utility to export strings and other locale-specific attributes to an external file.

To export strings and other locale-specific attributes

- 1 Choose Tools > Utilities > Locale Management.

The Locale Management Utility appears.
- 2 Select the Source and Target Languages.
- 3 If you want to localize a specific application, select the Application.
- 4 Select the Export Tab.

- 5 Select whether you want to export Strings only or all localizable attributes.

All localizable attribute includes Strings and other layout attributes, such as the positions of controls. These attributes may be different for different languages.

- 6 Select one of the following:

- To export objects from specific projects, select the Object present in projects radio button and then select the project or projects.
- To export objects from the application you selected under the Options tab (see [Step 3 on page 7-4](#)), select the Object present in chosen application radio button.

- 7 Click Export.

When you click export the objects that you selected are exported to a file. The default directory for the file is the Tools/Objects directory of your Siebel installation. If you have chosen to export Strings Only, the strings are exported to a text file called “results.txt.” If you have selected to export All Localizable Attributes, Strings and other UI attributes are exported to a file called “results.slf.”

Importing Strings and Other Locale-Specific Attributes

You use the Locale Management Utility to import translated strings and other locale-specific attributes back into the repository. Use the preview functionality to see the results of the import process before you actually import them into the repository.

To preview the results of the import process

- 1 Choose Tools > Utilities > Locale Management.

The Locale Management Utility appears.

- 2 Select a source language and a target language.
- 3 Select the Import tab.
- 4 Enter the directory path and name of the file you are going to import.
- 5 Enter the path and name of the file where you want to store the results for previewing.

The default file name is “preview.txt.”

6 Click Preview.

The Locale Management Utility writes the results of the import process to the log file rather than to the repository.

NOTE: LMU will not mark changed records with a Redo flag when running in Preview mode.

To import strings and other locale-specific attributes into the repository**1** Choose Tools > Utilities > Locale Management.

The Locale Management Utility appears.

2 Select a source language and a target language.**3** Select the Import tab.**4** Enter the file name of the file from which you want to import locale-specific attributes.

You can also use the Browse button to find and select the file. The default file name is:

- `Results.txt` if the file contains strings only
- `Results.slf` if the file contains all locale-specific attributes

5 Select whether you want to mark records in the repository with the Redo flag that have changed since the export occurred.

When the import occurs, the LMU compares the source language records in the repository with the source language records in the import file. If the records in the repository have changed since the export occurred they are marked with the Redo flag. This helps you identify records that may need to be retranslated.

6 Click Import.

The locale-specific attributes are imported into the repository.

Identifying Modified Objects

You can use the Locale Management Utility to identify objects that have been modified in the repository since the last time you exported strings. This is useful when your development and localization efforts occur simultaneously. It helps you keep strings in the repository in sync with the strings that have been exported to a file for localization.

You can search for modified objects using the following two methods:

- Base your search on a specific date.
- Compare objects in the repository with objects in a source file, such as `results.txt`.

NOTE: When you base your search on a specific date, and run the search by clicking the Start button, all records returned for a modified project are marked as “Redo,” regardless of whether a particular locale attribute has changed. This is because the LMU searches for changes at the object level (the base record), not the attribute level.

To identify modified objects

- 1** Choose Tools > Utilities > Locale Management.

The Locale Management Utility appears.

- 2** Select a source language and a target language.
- 3** Select the Modified Objects tab.
- 4** Define the search criteria you want to use:
 - Select the Changed since checkbox and then specify a date after which you want to find modified objects.
 - Select the Different from file checkbox and then specify the file to compare the repository against.

- 5 Do one of the following:
 - Click Start to find records that match the search criteria, display the results, and flag records returned in the search as Redo. Redo indicates that a record has been changed since the last time export occurred and therefore may need to be retranslated.
 - Click Preview to find records that match the search criteria and display the results. Preview does not mark records as Redo.
- 6 After you have identified modified objects, you can perform the following tasks:
 - Click the Save button to save a result set in a .log file.
 - Click the Go To button to open the Object Explorer and go to the parent object of the string or attribute.

NOTE: The Load button allows you to import a result set from a previously saved file. After loading the result set in the display window, you can perform save or goto operations on those records.

Localization

Using the Locale Management Utility

Siebel Tools and Multilingual Development

8

Multi-Locale Repository	8-2
Setting the Tools Language Mode	8-4
Checking In and Out Locale-Specific Attributes	8-5
Compiling an .srf File for a Specific Language	8-6
Working with Visual Editors	8-7

Multi-Locale Repository

Siebel Tools allows you to store multiple translations of text strings and other locale-specific attributes in the Siebel Repository. Siebel Tools distinguishes between the locale-neutral and locale-specific objects that define a Siebel application.

- Locale-neutral objects have attributes that are common among all language versions of the application.
- Locale-specific objects have attributes that are distinct for different language versions of the application. Locale-specific objects contain additional child object types that are used to store the locale-specific attributes, such as translatable text strings. For example, the Applet object type has a child object called Applet Locale. The Applet Locale object type contains the locale-specific versions of the Applet Title. Locale-specific objects also allow you to inactivate the display of controls by locale.

Locale objects also store locale-specific configuration information such as visibility. These types of attributes will show themselves as “*_language override” attributes in the parent base object. “*_language override” attributes will display the locale value of the current tools language mode.

The benefits of having a single repository for locale-neutral and locale-specific objects are:

- Editable locale-specific attributes, such as strings, in Siebel Tools
- Reuse of base objects (locale-neutral) across multiple languages
- Easier maintenance and upgrades
- Parallel development and localization

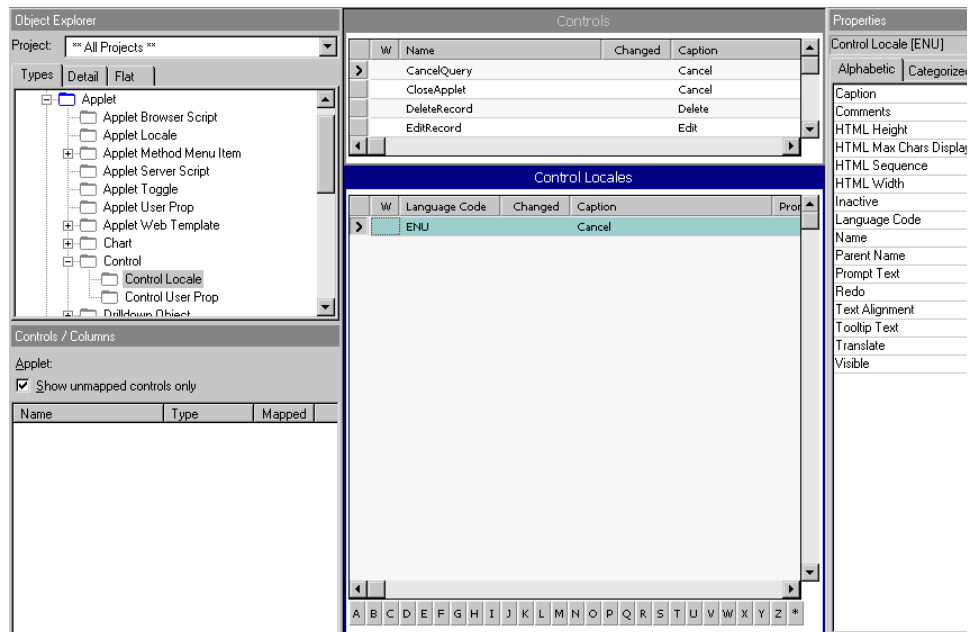


Figure 8-1. Controls and Control Locales Object Types

Each instance of Siebel Tools runs in a language mode defined by the user. The language mode determines the locale-specific objects with which the user is able to work. It also controls the checkout and locking process. Users can only check out and lock the language version of projects that is associated with their current language mode. However, users can check in other language versions of a project, regardless of the current language mode.

See [“Setting the Tools Language Mode” on page 8-4](#) for more information.

Setting the Tools Language Mode

Your working language allows you to work with locale-specific attributes that match your language mode. For example, suppose you are working in a base language of English, to access the Spanish version of locale-specific objects, you would simply change your working language to Spanish. This would allow you to view and edit the Spanish attributes of locale-specific objects. It would also allow you to check out Spanish versions of objects from the server database, modify them, and then compile a Spanish repository (`.srf`) file.

To change your working language

- 1** Choose View > Options.
- 2** Select the Language Settings Tab.
- 3** In the Language field, click the drop-down arrow and select your working language.
- 4** Select the Enable language override check box if you want to be able to edit the locale-specific layout in language override mode.
- 5** Click OK.

Your working language is displayed in the Tools Status bar.

Checking In and Out Locale-Specific Attributes

The Siebel Tools current language mode affects the Get process and the check in/check out and locking processes.

- You can perform a full Get that includes all the locale-neutral object definitions and the locale-specific object definitions associated with your current language mode.
- You can perform a Get that includes only locale-specific objects.
- You can only check out projects in the language defined by your current language mode.
- You can check in previously locked projects (of any language), regardless of your current language mode.
- Projects are locked on the server in a specific language determined by the user's current language mode.
- Projects on the server can only be checked by a single user in one language, at which point they become locked.

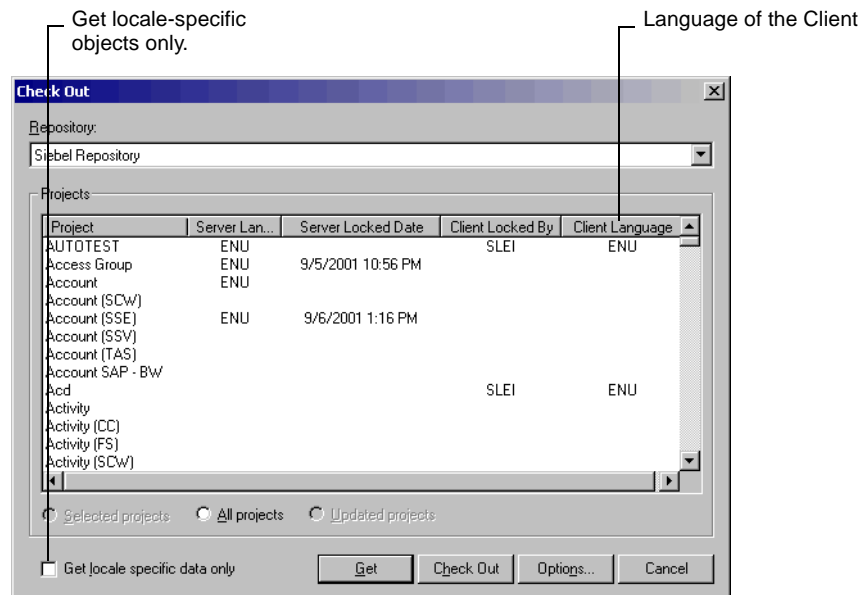


Figure 8-2. Check Out Dialog Box

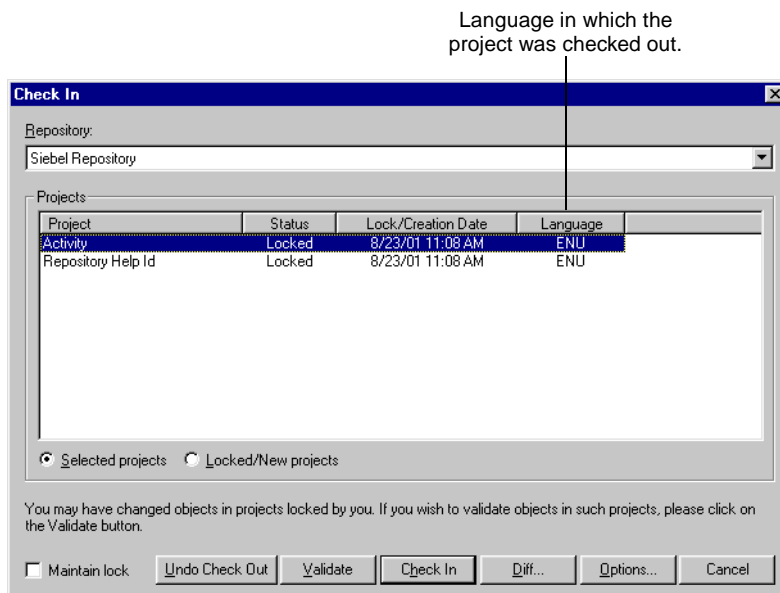


Figure 8-3. Check In Dialog Box

Compiling an .srf File for a Specific Language

You compile a separate repository file (.srf file) for each language you are supporting. For example, if your deployment includes an English and a Spanish language version, you must compile a repository file for each language. You do this by defining the language mode and then compiling. The repository file is compiled for the language you selected.

Working with Visual Editors

Visual editors in Siebel Tools have a base mode and a language override mode.

- **Base mode.** Allows you to edit the locale-neutral object attributes. When working in base mode, changes are stored in the base table.
- **Language override mode.** Allows you to edit language-specific object attributes. When working in language override mode, changes are stored in a child locale table.

You can switch between modes by selecting from the Mode drop-down list on the Configuration toolbar.

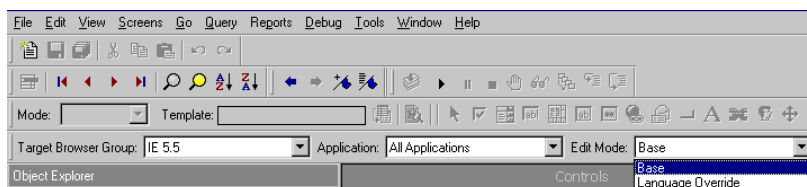


Figure 8-4. Edit Mode

NOTE: To be able to switch modes, the Language override setting in the Development Tools Options dialog box must be selected. See [“Setting the Tools Language Mode”](#) on page 8-4.

When modifying object definitions through visual editors in Siebel Tools in Language Override mode, keep the following key points in mind.

- Objects created or modified in English will store all attributes in the base object table only.
- Objects created in languages other than English will store locale-specific attributes in both the base table and the language overrides table.
- For objects that are modified in languages other than English, Tools will check whether the language specific attribute that is being updated is already in the language overrides table. If the attribute is in the language overrides table, Tools will update the entry. If the attribute is in the base object table but not the overrides table, Tools will create a new entry in the overrides table.

When modifying object definitions through visual editors in base mode, keep the following in mind:

- String translations are always stored in locale objects.
- All other locale-specific attributes other than English are stored in base objects.
- ENU strings are stored in locale objects.

NOTE: Base mode and language override mode are only applicable for visual editors. They do not apply to properties you enter in the Property Window or Object List Editor.

Index

A

- action arguments, enabling 4-27
- active language
 - defined 4-6
 - Siebel Clients, about determining 6-4
- applets
 - Dynamic Toggle Applets,
 - configuring 4-36
 - language-independent code, creating for Workflow Policy Column 4-26
 - language-independent code, creating for Workflow Policy Program
 - Argument 4-28
 - toggle applets, coding guidelines 4-23
- application-wide data
 - correspondence templates, about 6-6
 - literature, about including in Literature tab 6-6
 - sales cycles, about 6-6
 - SmartScript, about and campaigns 6-7
- architecture of Siebel Server, parameters and diagram 1-9
- ASCII and email, support of 5-9
- Assignment Criteria and Skills, configuring for MLOV 4-30
- attributes, locale-specific
 - exporting 7-5 to 7-6
 - importing 7-6 to 7-7
- audience for guide Intro-2

B

- base mode, about editing locale-neutral object attributes 8-7 to 8-8

business components

- dual currency display,
 - configuring 1-7 to 1-8
- dual currency display, configuring
 - example 1-8
- dynamic toggle, configuring 4-36
- MLOV implementation 4-23

C

- character set. *See* code pages
- characters, about undisplayable character conversion 2-5
- code pages
 - See also* deployment scenarios
 - about 2-2
 - common code pages and languages, table of 2-3
 - database support for 2-4
 - substitute characters 2-5
 - using third-party software 6-6
- coding guidelines for MLOV 4-23
- collation, definition and
 - example 2-6 to 2-7
- Communication Outbound Manager (COM), and Siebel eMail Response 5-4
- communication profile, about creating 5-7
- communication server, configuring 5-7
- concurrent multiple language support,
 - definition 1-2
- configuration
 - Applet Toggle objects in Siebel Tools 4-36
 - communication server 5-7
 - definition 1-12

- dual currency display 1-7 to 1-8
 - dual currency display example 1-8
 - fields for global time zones 4-43
 - MLOV configuration compatibility 4-23
 - MLOV implementation 4-23
 - multilingual target columns,
 - enabling 4-13
 - response groups 5-7 to 5-8
 - Siebel Anywhere for MLOV-enabled
 - fields 4-31
 - Siebel Assignment Manager for MLOV-enabled
 - fields 4-29
 - Siebel Field Service for MLOV-enabled
 - fields 4-31
 - target columns for multilingual use 4-12
 - visibility rules and MLOV 4-8
 - Workflow Manager for MLOV 4-25
 - correspondence templates, application-wide
 - data 6-6
 - Criteria Values and Skills, configuring 4-29
 - currencies
 - conversion, about 1-6
 - dual currency display 1-6 to 1-8
 - Euro support 1-6 to 1-8
 - Custom extension columns 4-8
- D**
- data
 - EIM and imported data 4-22
 - exporting 4-22
 - formatting based on locales 1-5
 - historical, converting to global time zone-enabled 4-48
 - historical, preparing for conversion to global time zone 4-43
 - historical, upgrading for global time zone 4-41
 - importing 4-22
 - MLOV upgrade log file,
 - about 4-20 to 4-21
 - operating system, affected by
 - settings 6-2
 - upgrading bounded columns 4-15
 - data formats, about setting regional
 - settings 6-3
 - Database Extension Designer, about using
 - to edit picklists 4-11
 - database, about collation sequences and
 - example 2-6 to 2-7
 - date formats, examples of
 - internationalization 1-5
 - dedicated Web clients
 - active language, about 4-6
 - active language, about determining 6-4
 - language packs, about distributing
 - additional 5-10
 - deployment scenarios
 - monolingual deployment, about and
 - example (diagram) 3-3
 - multilingual deployment, about and
 - example (diagram) 3-4
 - single-code page model, about and
 - requirements 3-2
 - deployment, global. *See* global deployment.
 - designated contact, becoming Intro-4
 - Display Value
 - field described 4-34
 - language-independent code, about
 - columns enabled for MLOV 4-25
 - queries, about using to run 4-24
 - displays, multilingual, special cases 4-12
 - DOS command prompt
 - MLOV Upgrade Utility, starting
 - from 4-18
 - Siebel Upgrade Wizard (development
 - environment), manually
 - launching 4-50, 4-53
 - drilldown, dynamic 4-23
 - driver_utc.ucf file
 - editing 4-44
 - parameters, editing 4-45
 - dynamic picklists, about and multilingual
 - example 4-3
 - Dynamic Toggle Applets, configuring 4-36

E

- elements, localizable and non-localizable (list of) 7-3
- email
 - See also *individual Siebel eMail Response entries*
 - ASCII, support of 5-9
 - Communication server, about 5-4
 - response templates, using to reply 5-4
- Enterprise Integration Manager, about using to import and export data 4-22
- errors
 - LOV type messages 4-20 to 4-21
 - UTC, in conversion files 4-52
- Euro currency, support of 1-6 to 1-8
- executables, distributing in multilingual environments 5-10 to 5-11
- export process 4-22

F

- features, new in this release Intro-3
- fonts
 - global deployments, importance in 6-5
 - localized Siebel application, recommended fonts (table) 6-5

G

- global deployment
 - definition 1-12
 - development strategy 1-13 to 1-14
 - in Siebel applications 1-2
 - terms defined 1-3 to 1-11
- Global Time Zone
 - See also *Universal Time Zone*
 - about 4-37
 - upgrading, about and example 4-41 to 4-48

H

- help
 - contacting technical support Intro-4 to Intro-5
 - online help, location of 6-4

I

- I18N. See *internationalization*
- internationalization
 - correspondence templates, about using 6-6
 - date formats 1-5
 - defined and use 1-4
 - dual currency display, configuring 1-7 to 1-8
 - dual currency display, configuring example 1-8
 - Euro currency support 1-6
 - features of 1-4 to 1-5
 - locale settings, formatting of 1-5
 - multi-currency support 1-6
 - number formats 1-5
 - phone number formats, examples 1-5
 - software architecture design, parameters and diagram 1-9
- ISO character sets, about and code pages 2-3

L

- Language column, described 4-34
- language detection. See *Siebel Smart Response*
- language field, viewing 5-5
- language packs
 - international languages, about support of 5-6
 - note, about uninstalling previous versions 5-10
 - for Web clients 5-10
- Language parameter
 - active language setting 4-6
 - active language, about using to change 6-4
- language-independent code
 - applet, creating for Workflow Policy Column 4-26
 - applet, creating for Workflow Policy Program Argument 4-28
 - avoiding in queries 4-24
 - deleted LOV records 4-35

- display of 4-23
- and Display Value 4-25
- display value translation 4-14
- exporting data 4-22
- List of Values use 4-34
- and MLOV 4-6
- modification caution 4-7
- picklist, creating for a Workflow Policy
 - Column 4-26
- picklist, creating for Workflow Policy
 - Program argument 4-27
 - and querying 4-24
 - and S_LST_OF_VAL 4-7
 - sorting and searching 4-23
 - target column display 4-23
- languages
 - See also* code pages
 - correspondence templates, about
 - using 6-6
 - language override mode,
 - about 8-7 to 8-8
 - mode defined by user 8-3
 - monolingual deployment scenario, about
 - and example (diagram) 3-3
 - multiple, working with 1-2
 - and object definitions 8-7
 - Siebel Language, about and
 - example 1-11
 - working language, changing 8-4
- List of Values. *See individual LOV entries*
- literature, about including in Literature
 - tab 6-6
- LMU. *See* Locale Management Utility
- local database, about collation sequences
 - and example 2-6 to 2-7
- Locale Management Utility
 - See also* localization
 - about using 7-4
 - modified objects, identifying 7-8 to 7-9
 - note, about modified records for project
 - marked as Redo 7-8
 - note, about not marking changed records
 - in Preview mode 7-7
 - strings and other locale-specific
 - attributes, exporting 7-5 to 7-6
 - strings and other locale-specific
 - attributes, importing 7-6 to 7-7
 - untranslated text strings,
 - finding 7-4 to 7-5
- locale object, about finding untranslated
 - text strings 7-4
- locale, defined, use, and example 1-3
- locale-neutral objects
 - attributes, about and storing in
 - repository 8-2
 - base mode, about using to
 - edit 8-7 to 8-8
- locale-specific attributes
 - checking in and out 8-5 to 8-6
 - exporting 7-5 to 7-6
 - importing 7-6 to 7-7
 - working language, changing 8-4
- locale-specific objects, about and storing in
 - the repository 8-2
- localization
 - See also* Locale Management Utility
 - applets localized 4-36
 - correspondence templates, about
 - using 6-6
 - defined 1-10
 - elements, localizable (list of) 7-3
 - fonts, about choosing 6-5
 - fonts, recommended (table) 6-5
 - process, steps 7-2
 - string expansion 6-5
- log file 4-20 to 4-21
- log space, allocating 4-48
- LOV table
 - See* language-independent code
 - administration columns 4-33
 - constraints on entries 4-35
 - display value 4-34
 - EIM validation 4-35
 - enabling for multilingual LOV
 - (MLOV) 4-8
 - entries, constraints on 4-35

- entries, displaying 4-32
- Language column 4-34
- multilingual integration
 - considerations 4-22
- records, adding 4-34
- records, deactivating 4-35
- records, deleting 4-35
- Translatable column 4-34
- LOV Type
 - See also* LOV table
 - enabling multilingual support 4-12
 - enabling target columns for MLOV 4-13
 - entries, displaying 4-32
 - error messages about 4-20 to 4-21
 - fixing from log file 4-20
 - multilingual designation 4-33
 - Multilingual property 4-12
 - querying 4-10
 - and translation 4-9
- LOV_Upgrade.log 4-20 to 4-21
- LOVs
 - about and example LOVs (table) 4-4
 - domains not in S_LST_OF_VAL table 4-21
 - Explorer view, about and screen example 4-33
 - flat view, about and screen example 4-32
 - inconsistently bounded columns 4-11, 4-20
 - monolingual, about 4-5
 - multilingual, about 4-5 to 4-7
 - multilingual, about enabling 4-7
- M**
- mailboxes, setting up 5-6
- Microsoft Windows,
 - regional settings 2-5
- MLOV
 - about 4-5 to 4-7
 - Assignment Criteria and Skills, configuring 4-30
 - configuration and coding guidelines 4-23
 - configuration for data read by server programs 4-8
 - drilldown, dynamic 4-23
 - enabling, about 4-7
 - and language-independent code 4-6
 - querying 4-24
 - search specifications 4-24
 - and Siebel Anywhere 4-31
 - and Siebel Assignment Manager 4-29
 - and Siebel Field Service 4-31
 - toggle applets, coding guidelines 4-23
 - and Translate property 4-10
 - upgrade utility, about 4-15
 - VB functionality, about 4-24
 - visibility rules, checking 4-9
 - Workflow Manager, configuring for 4-25
- MLOV table
 - administering MLOV entries 4-32
 - administration columns 4-33
 - columns, identifying which to enable 4-8 to 4-9
 - configuration compatibility 4-23
 - displaying columns, warning about 4-12
 - EIM validation 4-35
 - multilingual designation 4-33
 - multilingual integration
 - considerations 4-22
 - Multilingual property 4-12
 - records, adding 4-34
 - records, deactivating 4-35
 - records, deleting 4-35
 - sample administration screen 4-7
 - target columns 4-13
 - translating display values 4-14
 - visibility rule changes 4-8
- MLOV Upgrade Utility, starting from DOS Prompt 4-18

- mobile Web clients
 - active language, about 4-6
 - active language, about determining 6-4
 - global deployment tip 1-14
 - language packs, about distributing
 - additional 5-10
- monolingual deployment, about and example (diagram) 3-3
- monolingual LOVs, about 4-5
- multibyte languages
 - common page codes and languages (table) 2-3
- Multilingual column, described 4-33
- multilingual deployment
 - about and example (diagram) 3-4
 - personalization, about
 - configuring 5-12 to 5-13
- multilingual development
 - locale-specific attributes, checking in and out 8-5 to 8-6
 - multi-locale repository, about 8-2 to 8-3
 - repository file, about compiling for each language 8-6
 - visual editors, working with 8-7 to 8-8
 - working language, changing 8-4
- multilingual displays, special cases 4-12
- multilingual email
 - and COM 5-4
 - mailboxes 5-6
 - response groups, configuring 5-7 to 5-8
 - templates 5-7 to 5-8
- multilingual email environment. *See* Siebel eMail Response
- multilingual environment
 - Siebel eMail Response, about using in 5-4
- multilingual lists of values. *See individual MLOV entries*
- Multilingual property, LOV type 4-12
- multilingual response templates 5-4

N

- number formats 1-5

O

- object definitions
 - caution about changing 4-13
 - working with 8-7
- object managers
 - parameters, table of 5-14 to 5-16
- objects, identifying modified objects 7-8 to 7-9
- online help, location of 6-4
- operating system settings and display 6-2

P

- phone number formats, examples 1-5
- picklists
 - bounded 4-10
 - display value 4-34
 - language-independent code, creating 4-26, 4-27
 - LOVs, about and example (table) 4-4
 - monolingual LOVs, about 4-5
 - multilingual LOVs, about 4-5 to 4-7
 - multilingual LOVs, about enabling 4-7
 - repository, recompiling after changes 4-21
 - types of and multilingual example 4-3
 - unbounded 4-10
 - upgrading bounded columns 4-15
- platform, defined 1-11
- Policy Conditions, enabling 4-25
- profile, about creating communication drive profile 5-7

Q

- queries, about running against MLOVs 4-24

R

- Redo, about Locale Management Utility marking projects 7-8
- Regional Settings
 - about using 2-5

- and object manager parameters (table) 5-14 to 5-16
 - data formats, about setting 6-3
 - operating system, in the 6-2
 - Windows NT, viewing 6-3
 - release, new features Intro-3
 - remote clients, configuring multilingual 4-31
 - repository
 - compiling the .srf file 4-21
 - language, about compiling for each language 8-6
 - multi-locale, about 8-2 to 8-3
 - Siebel Upgrade Wizard, launching if conversion fails 4-53
 - Siebel Upgrade Wizard, manually launching 4-50, 4-53
 - response groups, configuring 5-7 to 5-8
 - response templates for email 5-4
 - results.slf, about 7-6
 - results.txt, about 7-6
- S**
- S_LST_OF_VAL
 - See also *individual LOV entries*
 - about 4-4
 - and language-independent code 4-7
 - LOV domains not in table 4-21
 - sales cycles, about 6-6
 - sample database, about collation sequences and example 2-6 to 2-7
 - Siebel Anywhere
 - and MLOV-enabled fields 4-31
 - executables, distributing in multilingual environment 5-10 to 5-11
 - Siebel Assignment Manager, configuring to use MLOV-enabled fields 4-29
 - Siebel Client
 - active language, about determining 6-4
 - fonts, about file used to change 6-5
 - fonts, choosing 6-5
 - Siebel eMail Response
 - See also email
 - about 5-2
 - deployment, planning 5-5
 - limitations 5-9
 - multilingual environment, using in 5-4
 - overview and architecture diagram 5-3 to 5-4
 - Siebel eMail Response, international deployment
 - See also email
 - communication driver profile, creating 5-7
 - Communication Server, configuring for outbound messages 5-7
 - language packs, installing 5-6
 - mailboxes, setting up 5-6
 - prerequisite tasks 5-6
 - response groups, configuring 5-7 to 5-8
 - templates, creating 5-9
 - Siebel Field Service and MLOV-enabled fields 4-31
 - Siebel Language, about and example 1-11
 - Siebel mobile Web client. See mobile Web clients
 - Siebel Personalization, about configuring multilingual environment 5-12 to 5-13
 - Siebel Report Server
 - about and example 5-13
 - Regional settings and Object Manager parameters 5-14 to 5-16
 - Siebel Repository. See repository
 - Siebel Server architecture, parameters and diagram 1-9
 - Siebel technical support Intro-4 to Intro-5
 - Siebel Tools
 - See also multilingual development
 - Applet Toggle objects, configuring 4-36
 - language mode, setting 8-4
 - multi-locale repository, about 8-2 to 8-3
 - visual editors, working with 8-7 to 8-8

- Siebel Upgrade Wizard, manually launching
 - development environment 4-50, 4-53
- Siebel Workflow Manager, configuring for MLOV-enabled fields 4-25
- single-code page deployment model, about
 - and requirements 3-2
- SmartScript, about and campaigns 6-7
- sort orders, collation definition and
 - example 2-6 to 2-7
- .srf file. *See* repository
- static picklists, about and multilingual
 - example 4-3
- string expansion, about when choosing
 - fonts 6-5
- strings
 - attributes, finding multiple 7-4
 - exporting 7-5 to 7-6
 - importing 7-6 to 7-7
 - untranslated, finding and
 - exporting 7-4 to 7-5
- T**
- tables, partitioning for UTC 4-46
- target columns
 - defined 4-15
 - display values vs. language-independent codes 4-23
 - enabling for multilingual use 4-6
 - LOV Type and multilingual support 4-12
 - Multilingual property 4-12
 - multilingual use, enabling for 4-13
 - translating display values 4-14
 - upgrading bounded columns 4-15
- technical support Intro-4 to Intro-5
- templates
 - acknowledgement template, about 5-8
 - correspondence templates, about
 - using 6-6
 - installed by language packs 5-6
 - language-dependent 5-7 to 5-8
 - multilingual deployments, about
 - creating 5-9
 - response templates, using to reply 5-4
 - Territory Assignment and MLOV, about
 - implementation 4-23
 - third-party software, about 6-6
 - time zones
 - global, about 4-37
 - records, default 4-40
 - setting for users 4-46
 - settings, modifying 4-40
 - user report, saving 4-46
 - Translatable column, described 4-34
 - translatable types, defined 4-9
 - Translate property, about using 4-10
 - translations
 - and LOV Type 4-9
 - custom extension columns 4-8
 - display values, adding 4-14
 - localization, role of 1-10
 - mode, about running MLOV Upgrade Utility 4-15
 - time zones, maintaining 4-41
 - triangulation defined 1-6
 - U**
 - undisplayable character conversion,
 - about 2-5
 - Universal Time Zone
 - See also* Global Time Zone
 - Global Time Zone,
 - upgrading 4-41 to 4-48
 - upgrade utility, about running in two modes 4-15
 - UTC conversion utility
 - conversion log files, reviewing 4-52
 - conversion parameters, table 4-51
 - data conversion to and
 - from 4-37 to 4-39
 - failure of 4-50
 - launching 4-53
 - running 4-50
 - turning on 4-53

V

- validation mode, MLOV Upgrade Utility 4-15
- validation mode, MLOV Upgrade Utility, about 4-15
- values
 - repicking (Workflow Policy Column) 4-27
 - repicking (Workflow Policy Program Argument) 4-28
- VB functionality, about writing VB code 4-24
- visibility rules
 - checking 4-9
 - MLOVs, and configuring 4-8
- visual editors, working with 8-7 to 8-8

W

- WE8ISO8859P1 setting, about Oracle language support 2-4
- Web clients

- active language, about 4-6
- active language, about determining for 6-4
- language packs, about distributing additional 5-10
- Western language, support of 2-4
- Windows
 - code page restrictions 5-9
 - Regional Settings, about using 2-5
 - Regional Settings, table of 5-14 to 5-16
 - Regional Settings, viewing 6-3
- Workflow Policy Column
 - configuring 4-26
 - LIC picklist, creating for 4-26
 - repicking 4-27
- Workflow Policy Program Argument
 - configuring 4-28
 - LIC picklist, creating for 4-27
 - repicking values 4-28
- Workload Rules, configuring 4-30

