



Developing and Deploying Siebel Business Applications

Siebel Innovation Pack 2015, Rev. A
September 2015

ORACLE®

Copyright © 2005, 2015 Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Developing a Siebel Application

About Developing a Siebel Application	11
Stages of Developing a Siebel Application	12
Documents You Use During the Design Stage	12
Testing You Perform When You Develop a Siebel Application	14
Overview of the Development Environment	17
About the Local Development Environment	18
About the Siebel Object Architecture	20
Typical Team Members of a Development Project	21
About The Business Case This Book Uses	22
Overview of the NREC Business Case	22
NREC Business Objectives	23
NREC User Profiles	24
NREC Access Requirements	24
NREC Business Requirements	25
NREC Development Strategy	27
Siebel CRM Functionality That NREC Plans to Use	28
NREC Design Strategy	28
Roadmap for Developing a Siebel Application	31

Chapter 3: Installing Software, Adding Users, and Creating the Company Structure

Installing Software for a Siebel Application	33
Process of Adding Users	36
Creating Database Users	36
Modifying the Configuration File for the Siebel Client	37
Creating Responsibilities	38
Adding Users to a Siebel Application	41
Process of Creating the Company Structure	42
Creating an Organization	43
Creating Positions and Position Skills	44

Creating a Partner Organization 48

Chapter 4: Setting Up the Development Environment

Process of Setting Up the Development Environment 51

Setting Up Local Databases 51

Setting Up Siebel Tools 60

Chapter 5: Configuring the User Interface

Process of Configuring the User Interface Layer 65

Removing Screens from a Siebel Application 65

Removing Views from a Screen 67

Removing Fields from an Applet 68

Adding Fields to Applets 70

Verifying Your Configuration 74

Chapter 6: Configuring the House and Opportunity Entities

Process of Configuring the House Detail View 75

Examining Business Requirements for the House Detail View 75

Adding Columns to the Products Table 77

Configuring the Internal Product Business Component 79

Configuring Text for Labels and Other User Interface Elements 80

Modifying the Product List Applet 83

Modifying the Product Form Applet 85

Creating the House Detail View 86

Creating the Houses Screen 87

Compiling, Registering, and Testing the House Detail View 90

Process of Configuring the House Detail - Appraisals View 90

Examining Business Requirements for the House Detail - Appraisals View 90

Creating the Appraisals Business Component 92

Creating a Link Between Houses and Appraisals 94

Associating the Appraisals Business Component with a Business Object 95

Creating the Appraisals List Applet 96

Creating the House Detail - Appraisals View 97

Adding the House Detail - Appraisals View to the Houses Screen 97

Compiling, Registering, and Testing the House Detail - Appraisals View 98

Process of Configuring the Opportunity Details View 99

Examining Business Requirements for the Opportunity Details View 99

Adding Columns to the Opportunity Table 100

Adding Fields to the Opportunity Business Component 101

Modifying the Opportunity List Applet 101

- Modifying the Opportunity Form Applet 103
- Compiling, Registering, and Testing the Opportunity Details View 105

Chapter 7: Configuring Picklists, Drilldowns, and MVGs

- About Picklists and Drilldowns 107
 - About Picklists 107
 - About Drilldowns 108
- Process of Configuring Picklists, Drilldowns, and MVGs 109
 - Creating Static Picklists 109
 - Creating a Dynamic Picklist 111
 - Constraining a Picklist 113
 - Creating a Dynamic Drilldown 114
 - Configuring a Multi-Value Group 115

Chapter 8: Configuring Logos, Banners, and Colors

- Overview of Configuring Logos, Banners, and Colors 119
 - About User Interface Elements 120
 - About Developing Web Templates 122
- Process of Configuring Logos, Banners, and Colors 122
 - Examining Business Requirements for Configuring Logos, Banners, and Colors 123
 - Adding a Custom Logo to the Banner 124
 - Modifying the Banner Color 126
 - Modifying the Screen Bar and View Bar Colors 127
 - Modifying Applet Colors 128

Chapter 9: Configuring Siebel CRM to Use Data From an External Data Source

- Process of Configuring Siebel CRM to Use Data From an External Data Source 131
 - Examining Business Requirements for Using Data from an External Data Source 131
 - Creating a Business Service 132
 - Creating a Virtual Business Component 135
 - Creating a Link Between Houses and House Renovations 136
 - Modifying the Business Object 137
 - Displaying Custom Fields in the Siebel Client 137
 - Compiling and Testing Your Changes 139
 - Documenting Your Configuration in an Entity Relationship Diagram 140

Chapter 10: Assigning Objects and Automating Business Processes

- Customizing How Siebel CRM Assigns Objects 143

- Process of Automating a Business Process 146
 - Examining Business Requirements for Automating a Business Process 147
 - Configuring Siebel Communications Server 147
 - Creating an Email Template 147
 - Creating and Deploying a Workflow Process 149
 - Creating a Workflow Policy 151
 - Running the Generate Triggers Server Component 153
 - Defining and Starting a New Workflow Monitor Agent 154
- Filtering Records According to a User Profile 155

Chapter 11: Configuring Siebel Remote

- About Siebel Remote 161
- Process of Configuring Siebel Remote 162
 - Setting Up the Siebel Remote Server 162
 - Setting Up a New Siebel Remote User 162
 - Registering a Siebel Mobile Web Client 163
 - Extracting, Initializing, and Synchronizing the Remote Database 164

Chapter 12: Setting Up the Test Environment

- Process of Moving Data to the Test Environment 167
 - Moving Repository Data 167
 - Moving Web Templates, Image Files, and Cascading Style Sheets 169
 - Moving Nonrepository Data 170
- Process of Loading CRM Data in the Test Environment 170
 - Loading the Interface Tables 171
 - Adding the EIM Configuration File 172
 - Disabling Transaction Logging 175
 - Running the Enterprise Integration Manager 175
- Administering the Test Environment 177

Chapter 13: Common Software Development Tasks

- Using Siebel Tools 179
 - Displaying the Object Explorer 179
 - Displaying the Object List Editor 180
 - Displaying the Properties Window 180
 - Opening a Wizard 180
 - Opening the Web Layout Editor 180
 - Setting Options in Siebel Tools 182
- Process of Modifying Siebel Objects 183

Checking Out Projects from the Siebel Server	183
Making Configuration Changes	184
Compiling Projects	184
Logging In as the Siebel Administrator	185
Registering a New View	186
Verifying Your Changes	187
Checking Projects In to the Siebel Server	187
Options for Managing Objects	188

Chapter 14: Guidelines for Testing a Deployment

Guidelines for Integration Testing	191
Guidelines for User Acceptance Testing	192
Guidelines for Performance Testing	193
Guidelines for Continuous Improvement	194
Benefits of Functional Test Automation	195
Benefits of Load Test Automation	196

Appendix A: Code for NREC Business Services

Code for the General Function	199
Code for the Service Function	205

Index

1

What's New in This Release

What's New in Developing and Deploying Siebel Business Applications, Siebel Innovation Pack 2015, Rev. A

Notes have been added to this guide regarding a change in the software used with the local database for Siebel Remote.

NOTE: Siebel Innovation Pack 2015 is a continuation of the Siebel 8.1/8.2 release.

What's New in Developing and Deploying Siebel Business Applications, Siebel Innovation Pack 2015

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

NOTE: Siebel Innovation Pack 2015 is a continuation of the Siebel 8.1/8.2 release.

What's New in Developing and Deploying Siebel Business Applications, Siebel Innovation Pack 2014

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

NOTE: Siebel Innovation Pack 2014 is a continuation of the Siebel 8.1/8.2 release.

2

Overview of Developing a Siebel Application

This chapter describes an overview of developing a Siebel application. It also describes a fictitious company who develops a Siebel application. It includes the following topics:

- [About Developing a Siebel Application on page 11](#)
- [About The Business Case This Book Uses on page 22](#)
- [Roadmap for Developing a Siebel Application on page 31](#)

Goals and Scope of Developing a Siebel Business Application

This book describes how to develop Oracle's Siebel Business Applications at a fictitious company. The goal is to learn by doing in the context of a real-world case study. If you follow the procedures in this book, then you will have a functioning Siebel application that includes some typical customizations that are easy to complete. Note that this Siebel application does not include CRM data. These customizations provide you an overview of how to customization a Siebel application. Although every topic does not describe every possible customization, many topics do include cross-references to more information.

This book describes a basic strategy to plan, design, and deploy a Siebel application for your specific requirement. You can apply this strategy during the case study, and then generalize it to your company.

This book does not describe a solution design or a detailed development methodology. It presents techniques in a real-world context that you can generalize to your environment.

About Developing a Siebel Application

This topic describes an overview of developing a Siebel application. It includes the following topics:

- ["Stages of Developing a Siebel Application" on page 12](#)
- ["Documents You Use During the Design Stage" on page 12](#)
- ["Testing You Perform When You Develop a Siebel Application" on page 14](#)
- ["Overview of the Development Environment" on page 17](#)
- ["About the Local Development Environment" on page 18](#)
- ["About the Siebel Object Architecture" on page 20](#)
- ["Typical Team Members of a Development Project" on page 21](#)

Stages of Developing a Siebel Application

Figure 1 illustrates the stages of developing a Siebel application. This book describes the configure and deploy stages.

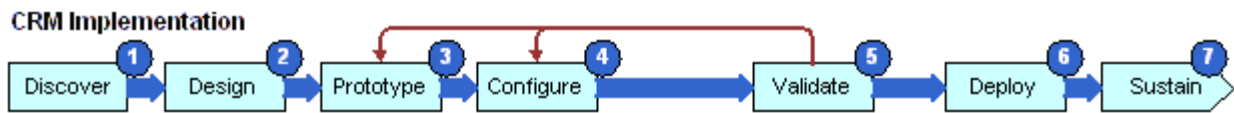


Figure 1. Stages of Developing a Siebel Application

Developing a Siebel application includes the following stages:

- 1 **Discover.** Identify and document functional and technical requirements that support the business goals.
- 2 **Design.** Design the project approach and scope, and incorporate project management controls. For more information, see [Documents You Use During the Design Stage on page 12](#).
- 3 **Prototype.** To develop application screen flows and design layouts, use the requirements from the discovery stage to create a prototype of the solution.
- 4 **Configure.** Configure the application, extensions, and external interfaces that are required to support the custom Siebel application.
- 5 **Validate.** Perform a full test of the functionality, including using production data to perform a user acceptance test of the application.
- 6 **Deploy.** Do the following:
 - Perform a production test that tests and revises the custom Siebel application, user training, technical infrastructure, the network, and the help desk.
 - Deploy the solution from the testing environment to the production environment.
- 7 **Sustain.** Evaluate the product and perform continuous improvement. Measure how well the solution meets the business goals.

Documents You Use During the Design Stage

This topic describes documents that describe business requirements and detailed design. These documents are outputs of the design stage of developing a Siebel application. They describe the scope of the solution and document the specifications that developers require during configuration.

You can use the following design documents:

- **Entity relationship diagram.** Illustrates relationships between entities. Entities include object definitions, such as business components or business objects. For more information, see [Figure 7 on page 29](#).
- **Business component design template.** Defines the properties of business components, including user properties, joins, single-value fields, multi-value fields, and multi-value links.

- **Business object design template.** Groups business components and identifies links between business components, which indicates the parent and associated child records in the Siebel client. You use the business component design template and then the business object design template.
- **Applet design template.** A reference document you create for each applet. It documents the properties of the applet, the fields that are available on the applet, and the properties that are associated with each field. It must include a mock-up or screen shot of the applet. You create an applet design after you decide what entities you must track in your Siebel application.
- **View design template.** Describes what applets are part of the view definition and identifies any special visibility associated with a view. For example, a *My Team's View* or an *All View*. It must also document how the user uses drilldowns to navigate through the views, tabs, and other user interface elements.
- **Screen design template.** Lists the views available from each screen and indicates the default view for each screen.
- **Report design template.** Describes information about reports. Includes the business component, whether the report is dynamic or static, and lists any subreports that must be available.
- **Workflow policy template.** Describes information required to create workflow rules.
- **Assignment manager template.** Describes information required to create assignment rules.
- **Testing strategy document.** Describes project objectives and provides a structure for tracking progress. Without proper planning you might perform too much or too little testing.
- **Test cases document.** Describes test cases according to approved technical designs. A written test case can also serve as a blueprint to develop an automated test script.

Getting Help from Oracle

Design documentation is an important part of a development effort. If you require help from Oracle you can create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support. You can also contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

Testing You Perform When You Develop a Siebel Application

Figure 2 illustrates testing activities. These activities help to make sure you use proper project management and control techniques during the project.

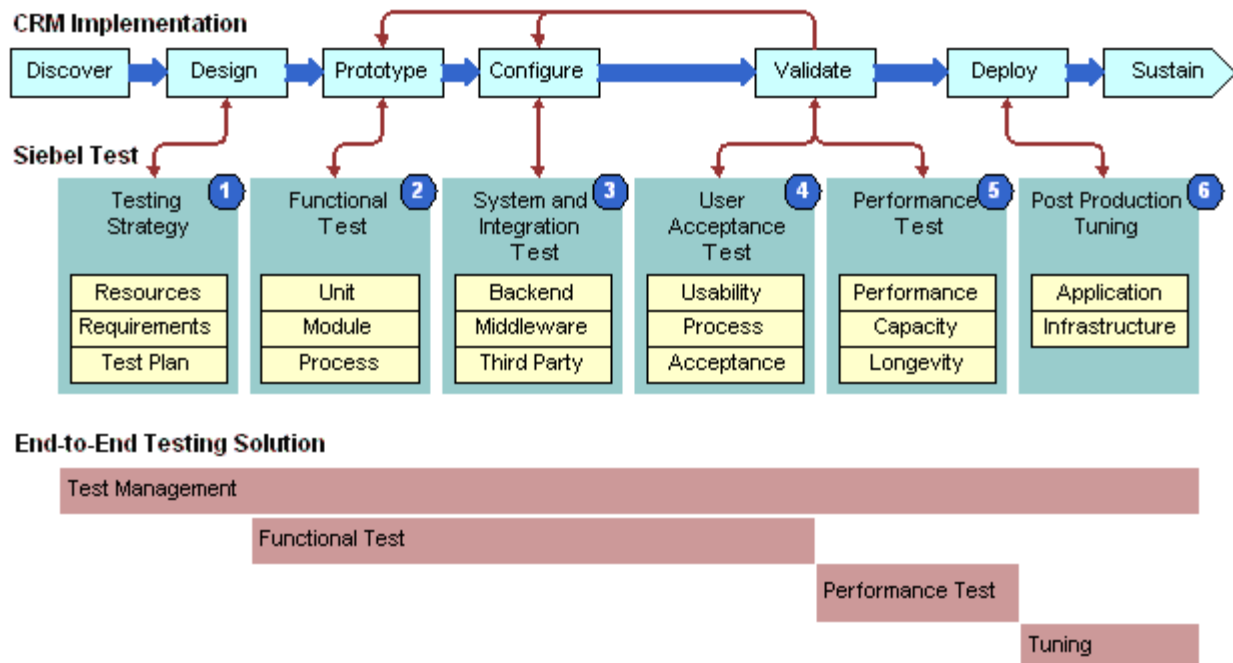


Figure 2. Testing You Perform When You Develop a Siebel Application

Developing a Siebel application includes the following parts:

- 1 Testing strategy.** Makes sure that testing provides useful information, minimizes risk, and provides a way to track progress. Proper planning helps to make sure you perform the correct amount of testing. You do the following work:
 - Identify project objectives and develop plans according to those objectives.
 - Develop a testing strategy and coordinate project stakeholders
 - Develop test cases. Developers and testers finalize the test cases according to approved technical designs. The written test cases can also serve as a blueprint to develop automated test scripts. To make sure test cases are realistic, you must include input from business analysts.

- Evaluate the design. Business analysts and developers verify that the design meets the business unit requirements. Do not start development work until team members agree that the designed solution meets the project objectives. It is more cost effective to prevent problems or omissions during the design stage than addressing them later. If a design is flawed from the beginning, then the cost to redesign later can be high.

You begin testing when you start to configure the Siebel application. You develop a testing strategy to meet the following goals:

- Defines environmental simulation requirements
- Identifies testing methods
- Establishes priorities
- Defines and creates functional and load test scripts
- Provides a comprehensive test plan

2 Functional test. Validates units of the Siebel application. Begins the same time as prototyping and continues during unit testing. Developers and the test team do the following work:

- Perform functional tests progressively on units, modules, and business processes to verify that the Siebel application functions correctly. Developers test units and resolve problems.

The unit test makes sure an individual unit of the overall application can function correctly, and prevents introducing unstable units in the larger application. The developer tests their code against a design specification. A unit test is an isolated test that is often the first test that the developer performs in their own environment before checking changes in to the configuration repository.

- Move units to the testing environment and combine units to form a module.
- Verify module functionality. For example, the team makes sure the module includes the correct layout, such as list menus and text fields, and returns the correct values. The team develops features and tracks problems to manage quality.
- To verify modules work together correctly, performs functional tests that use business processes and scenarios.

3 System and integration test. Validates that the Siebel application works correctly with new modules or other applications and interfaces. Tests the Siebel application in a test environment. This environment allows the Siebel application to interoperate with other required applications. You must verify integration with backend, middleware, and third-party applications. You perform integration testing on the entire system to make sure that the Siebel application functions properly when connected to related applications and other infrastructure components.

4 User Acceptance test. Tests Siebel application usage with business owners and users. You perform this testing on the complete system. You perform usability tests, which test how well the application works during user interactions with the Siebel client when the user attempts to complete specific tasks. You validate how the application supports business processes in the user community in the lines of business and the IT organization. You make sure the application meets performance requirements and that there are no user task failures or prohibitive response times. This is typically a very busy time in the project, when people, process, and technology prepare for the rollout.

- 5 Performance test.** Determines if the infrastructure performs and scales to your requirements. You perform a *stress test*, which is a type of test that identifies the maximum load the hardware configuration can handle. Test scenarios simulate expected peak loads. You usually use an automation tool to measure client and server response time. This stage requires an image of the full database and all interfaces with the Siebel application, such as computer telephony integration, middleware, or email. You must perform this test in an environment that uses realistic test data. You do the following:
- Complete a performance test to establish a performance benchmark.
 - Complete a capacity test. You add users until you reach the number of users that you expect will use the Siebel application during peak usage over the lifetime of the application.
 - Perform a longevity test, which tests the Siebel application over an extended period of time. This test determines application durability, and it identifies problems that become visible over time.
- 6 Post production tuning.** Testing is not complete when you roll out the application. After you complete the initial deployment, Oracle delivers regular maintenance and major software releases that you might apply. You review the test strategy, test objectives, test plans, and test cases to identify any areas for improvement. You must update test cases to include scenarios that you discovered during testing.

You also perform *regression tests*, which is a type of test that makes sure predefined and custom features function correctly together. Code additions or changes can unintentionally introduce errors or regressions. You must periodically perform integration and regression testing. These tests make sure your changes do not prevent predefined features from working, and make sure that the Siebel application can interact with other applications and products. If a test fails, then you must locate the faulty configuration and make the required corrections. If you perform these tests too infrequently, then troubleshooting might become problematic.

Reasons to Test Software

Testing is an important part of any application development project. The testing process determines the readiness of the application. Testing determines if application development meets design requirements. Testing helps to make sure the project team makes informed decisions and does not expose the business to undue risk. Thorough planning and testing can provide the following benefits:

- **Reduce deployment cost.** Identifies problems early in the project to reduce total cost of ownership. The cost of resolving a problem increases significantly in later development stages. A problem discovered during the discover stage can be significantly less expensive to address than the same problem discovered after you deploy the application. When in production, a serious problem can result in lost business and affect the success of the project.
- **Improve user acceptance.** User perception of quality is important to the success of a deployment. Functional testing, usability testing, and performance testing can identify problems early enough so that you can correct these problems before you release the application to the larger user community.

- **Improve deployment quality.** An application that performs reliably is critical in delivering consistent service to users and customers. Inadequate hardware and software resources can cause an outage and lead to lost business. Performance, reliability, and stress testing can determine how the application handles a production load and allows your information technology organization to plan accordingly.

Performing testing early and often can lower the total cost of ownership. A software project that attempts to save time and money by lowering the initial investment in testing often results in a higher cost of ownership.

Overview of the Development Environment

Figure 3 illustrates an overview of the development environment.

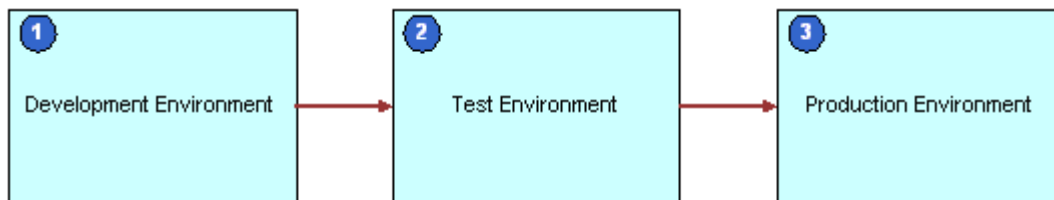


Figure 3. Overview of the Development Environments

The development environment includes the following environments:

- 1 Development.** You use this environment to perform the initial configuration. It typically includes a server computer, several client computers, development tools, and a small set of data that includes repository data, seed data, and sample transactional data. A *development server* is the server computer in the development environment. For more information, see [“About the Local Development Environment” on page 18](#).
- 2 Test.** You use this environment to test your custom Siebel application with data that simulates the production environment. A separate computer, or several computers, contain business data but no development tools. You copy the configuration from the development environment to the test environment. To simulate the production environment, you configure rules and set up actual data for your organization. After you complete the setup, you test customizations, patches, and version upgrades.
- 3 Production.** Similar to the test environment that exists at the end of testing, but the production environment includes hundreds or thousands of users and live business data. You move the configuration and user data from the test environment to the production environment, and then deploy the product to users.

Each environment corresponds to a stage in the application rollout. Each environment contains separate hardware, software, and data. Using this structure prevents the activities you perform in one environment from interfering with activities you perform in another environment. For example, configuration work in the development environment does not interfere with testing activities in the test environment.

About the Local Development Environment

A *local database* is a snapshot of the server database that is stored on a remote computer. A developer who uses Siebel Tools works on a local database that the developer installs on their personal computer. The data in each local database is a subset of the server database, as determined by the visibility rules that you assign to each developer.

Figure 4 illustrates the local development environment.

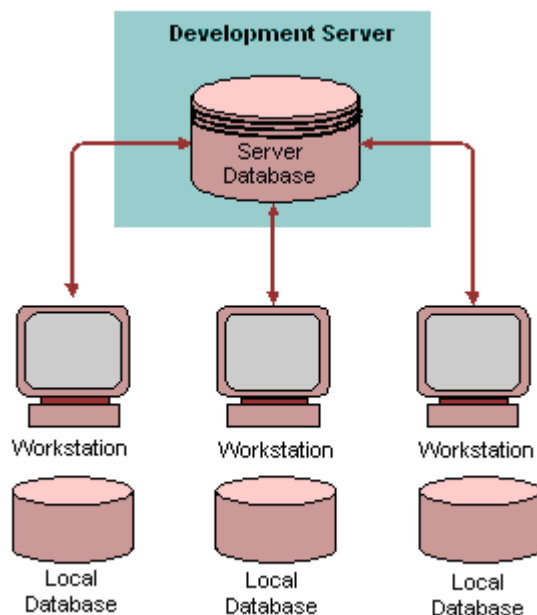


Figure 4. Local Development Environment

A developer checks out object definitions from the server database, modifies these definitions, and then checks them back in to the server database. The check-out and check-in process does the following:

- **Check out.** Copies projects from the server database to the local database and locks the projects on the server database. Locking the project on the server database prevents multiple developers from simultaneously checking out the same object.
- **Check in.** Copies projects from the local database to the server database and releases the lock on the server database.

Working on a local database rather than working directly on the server database provides the following benefits:

- Allows you to remove unwanted changes without overwriting the work of other developers. Working on a local database provides you the option of not checking in changes to the server database. Until you check in changes, the server database remains a clean backup.
- Allows several developers to concurrently use a single repository.

- Allows developers to prototype configuration changes by locking a project locally rather than checking out and locking the project on the server database.
- Allows you to work remotely because you are not required to be connected to the development server.

Siebel CRM does not update Siebel Web templates during project check-in and check-out. To manage Siebel Web template files, you must use a third-party source file control application or a manual process. For more information, see [“Moving Web Templates, Image Files, and Cascading Style Sheets” on page 169](#).

For more information, see [“Checking Out Projects from the Siebel Server” on page 183](#) and [“Checking Projects In to the Siebel Server” on page 187](#).

About the Siebel Object Architecture

The metadata that defines Siebel CRM objects and other files, such as web templates and style sheets, is divided in several architectural layers. Figure 5 illustrates the architecture.

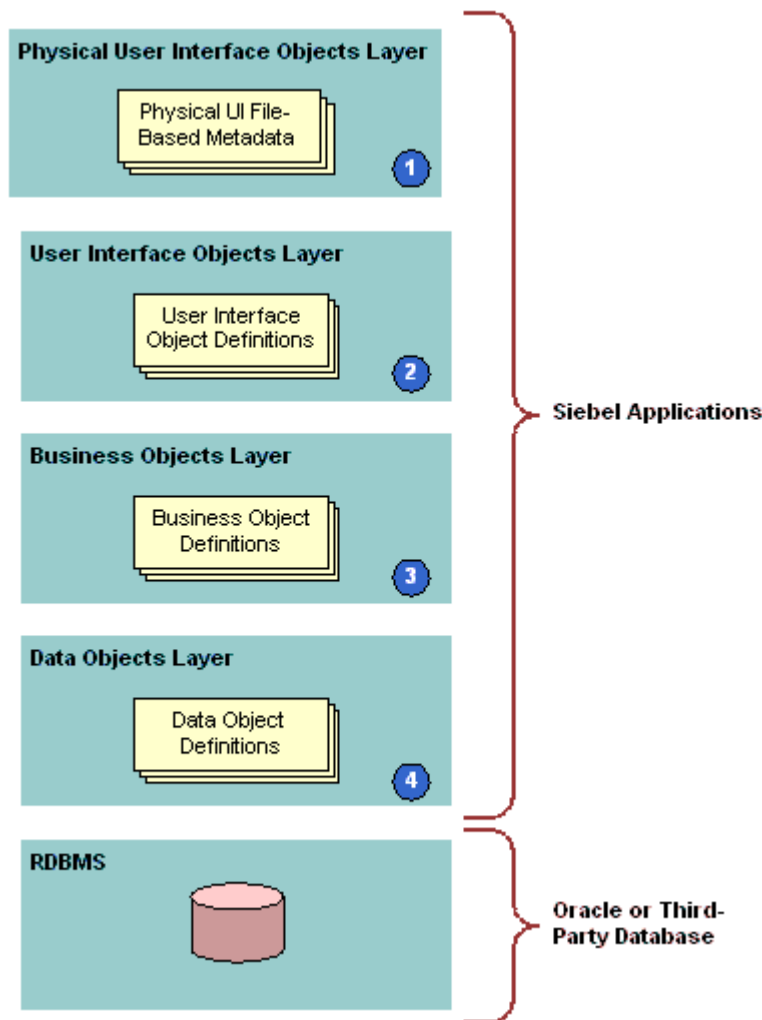


Figure 5. Siebel Object Architecture

The architecture includes the following layers:

- 1 **Physical user interface objects layer.** Includes the physical files, templates, style sheets, and other metadata that reference files that render the user interface in the Siebel client.
- 2 **Logical user interface object layer.** Includes object definitions for the user interface in the Siebel client. These objects define the visual elements that Siebel CRM displays to the user and with which the user interacts in a Web browser. A user interface object references a business object.

- 3 **Business object layer.** Includes objects that define business logic and organizes data from underlying tables into logical units.
- 4 **Data objects layer.** Includes object definitions that directly map the data structures from the underlying relational database to Siebel CRM. These objects provide access to those structures through object definitions in the business object layer.

Siebel CRM insulates objects in each layer of the Siebel object architecture from the layers above and below each layer, including the RDBMS (Relational Database Management System) at the bottom of the architecture. This allows you to change or customize Siebel objects in one layer without affecting other layers. Because Siebel CRM separates the database in the RDBMS from the object architecture, you can make database changes with only minimal affect on Siebel CRM.

For more information, see *Configuring Siebel Business Applications*.

About the Sequence You Use to Define Objects

It is useful to think of configuration tasks in terms of the Siebel object architecture. You might work from the bottom up, starting with data objects first, then business objects, and then user interface objects. Or you might work in one layer only, modifying objects as needed. For the NREC example in this book, you modify objects in the user interface layer first.

Typical Team Members of a Development Project

Table 1 describes typical team members that you require during a development project.

Table 1. Typical Team Members of a Development Project

Resource	Development Stages					
	Define	Discover	Design	Configure	Validate	Deploy
Project Manager	Yes	Yes	Yes	Yes	Yes	Yes
Lead Business Analyst	Yes	Yes	Yes	Yes	Yes	Yes

Table 1. Typical Team Members of a Development Project

Resource	Development Stages					
	Define	Discover	Design	Configure	Validate	Deploy
Lead Configuration Specialist	Not applicable	Yes	Yes	Yes	Yes	Yes
Lead Architect		Not applicable	Yes	Yes	Yes	Yes
Configuration Specialist			Not applicable	Yes	Yes	Yes
Architect		Not applicable		Yes	Yes	
Training Specialist				Yes	Yes	
Documentation Specialist		Yes	Yes			

About The Business Case This Book Uses

This topic describes the business case this book uses. It includes the following topics:

- [“Overview of the NREC Business Case” on page 22](#)
- [“NREC Business Objectives” on page 23](#)
- [“NREC User Profiles” on page 24](#)
- [“NREC Access Requirements” on page 24](#)
- [“NREC Business Requirements” on page 25](#)
- [“NREC Development Strategy” on page 27](#)
- [“Siebel CRM Functionality That NREC Plans to Use” on page 28](#)
- [“NREC Design Strategy” on page 28](#)

Overview of the NREC Business Case

National Real Estate Clearinghouse (NREC) is a fictitious company that participates in the residential real estate industry. It is an agent for people who sell houses. NREC maintains a database of houses for sale and makes this information available to a nationwide network of partner real estate agencies. These real estate agencies represent buyers. Real estate agents use NREC to find houses that match buyer requirements. Through prenegotiated contracts with partner real estate agencies, NREC can close sales with fewer costs than a traditional real estate agency.

Figure 6 illustrates how NREC solicits and routes interest directly from a buyer to an agent. To inquire about a house, a buyer calls a toll-free number. NREC records the buyer information and, rather than handling the lead, NREC sends it to an agency in the NREC distribution network.

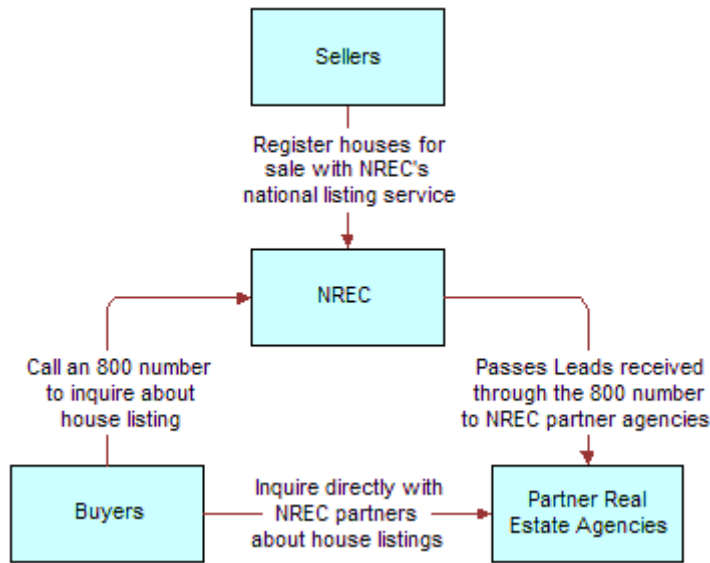


Figure 6. How NREC Interacts with Buyers, Sellers, and Partners

NREC Business Objectives

NREC uses a proprietary solution that it developed to manage houses and buyer leads. It regularly faxes information about available houses to partner real estate agencies and dispatches leads when needed. However the company realizes that this business process costs money and makes it difficult to attract new, independent real estate agencies to join the NREC distribution network.

To address these problems, NREC decides to develop a Siebel business solution. This solution is employee and partner facing. It provides participants with access to the information they require to do their jobs. Rules in the Siebel application streamline inefficiencies. For example, the Siebel application automatically dispatches buyer leads to partners according to assignment rules that all parties have agreed upon. The Siebel application can automatically assigns an opportunity to an agent, and can automatically send an email notification to the agent regarding the new assignment.

The employee applications in this deployment are modeled on Siebel Sales and Siebel Partner Manager, while the partner application is modeled on Siebel Partner Portal. Because each Siebel application shares a common back end, all users view the most up-to-date information.

Because this book assumes that the configuration for the employee applications proceeds successfully, it focuses on the configuration of Siebel Partner Portal, which is the partner application. All examples in this book that describe the Siebel client use Siebel Partner Portal.

NREC User Profiles

NREC includes the following user profiles:

- **NREC application administrator.** NREC operates the Partner Portal Web site for NREC partner real estate agencies and their agents. Administrators add houses and leads as they become available. The Partner Portal Web site automatically sends these leads to a partner agency. Administrators perform administration tasks, such as managing partner information.
- **Real estate agent.** To manage opportunities and contact information, browse houses for sale, and track activities related to each opportunity, real estate agents of the partner agencies use the NREC Partner Portal site.
- **Real estate agent manager.** Managers at partner real estate agencies use the NREC Partner Portal site to track opportunities and activities for every agent in the agency. They also use the NREC Partner Portal to examine houses for sale, and to track their own opportunities, contact information, activities, and so forth.

NREC Access Requirements

After analyzing access requirements, NREC determines that it has the following data access requirements:

- **Access through low-bandwidth Internet connections.** NREC partner agents must access data over low-bandwidth connections without installing the Siebel application on their personal computers. A dial-up connection is an example of a low-bandwidth connection.
- **Access through corporate intranet.** NREC administrators and employees must perform administration tasks, such as creating new users, assigning responsibilities, and so forth. They must also maintain data, such as information about houses, opportunities, contacts, and so forth. They must perform this work through a graphical user interface connected to the corporate intranet.
- **Mobile access.** NREC employees must access and manipulate data on their laptops while they are not connected to a corporate database. They must synchronize with the corporate database, when required.

To meet these requirements, NREC plans to use the Siebel Web Client and Siebel Mobile Web Client:

- **Siebel Web Client.** The user uses their Internet browser to connect to the Siebel server. The Siebel Web Client is an HTML, Web interface. It meets the NREC requirements to access a Siebel application through the Internet or an intranet.
- **Siebel Mobile Web Client.** Agents can use a Siebel application and work offline in a local database that resides on the personal computer that the agent uses. The user synchronizes their local database with the server database, when necessary. The Siebel Mobile Web Client provides the same HTML interface as the Siebel Web Client, but the local Siebel application serves the Web pages rather than a server. It meets the NREC mobile access requirement. For more information about deploying the Siebel Mobile Web Client, see *Siebel Installation Guide for Microsoft Windows*.

NREC Business Requirements

The project team identifies and documents requirements during the discovery and design stages of developing a Siebel application. Predefined Siebel CRM functionality satisfies some of these requirements. The team identifies requirements that are not satisfied as gaps. These gaps require configuration work that modifies the predefined Siebel application.

NOTE: As a general rule, a gap analysis results in less than 25% of the functionality that requires customization. If the gap is more than 25%, it might indicate that the team is not employing predefined functionality appropriately. For more information, see [“Getting Help from Oracle” on page 13](#).

Manage the List of Houses for Sale

NREC must manage the list of houses that is currently for sale. This includes storing certain characteristics, such as price, number of bedrooms, and square footage. It also must incorporate data from an external file that a third-party vendor supplies. This data includes appraisal and renovation history.

[Table 2](#) describes requirements for houses.

Table 2. Requirements for Houses

Requirement	Configuration Work You Must Perform
Track houses for sale.	You create a new screen for Houses and add it to the Siebel eChannel application. For more information, see “Creating the Houses Screen” on page 87 .
Track the features of each house, such as square footage, number of bedrooms, and so forth.	To store more attributes and display them in the Siebel client, you add columns to the S_PROD_INT table. For more information, see “Process of Configuring the House Detail View” on page 75 .
Enter and edit records for houses.	To allow updates and add necessary controls to the Product Form Applet, you modify the Internal Product business component. For more information, see “Configuring the Internal Product Business Component” on page 79 and “Modifying the Product List Applet” on page 83 .
Provide the user with predefined values to choose from when the user enters house features.	You configure static picklists for fields. For more information, see “About Picklists” on page 107 .
Track and display previous appraisal information for each house. Example information includes date, amount, and assessor.	To store appraisal data and display fields in the Siebel client, you use a predefined one-to-many table. For more information, see “Process of Configuring the House Detail - Appraisals View” on page 90 .

Table 2. Requirements for Houses

Requirement	Configuration Work You Must Perform
Display renovation information for each house.	To store and display data from flat file, you create a virtual business component. You create a new view to display the data. For more information, see “Creating a Virtual Business Component” on page 135.
Display houses to partner users according to ZIP code.	Use Siebel Personalization to filter the list of houses according to ZIP code. For more information, see “Filtering Records According to a User Profile” on page 155.

Manage Opportunities

NREC must provide NREC internal employees and partner agents a way to manage opportunities. Uses record opportunities for potential buyers and include contact information and information about the type of house in which the buyer is interested. NREC sends these opportunities to a partner agency according to location.

[Table 3](#) describes requirements for opportunities.

Table 3. Requirements for Opportunities

Requirement	Configuration Work You Must Perform
For each opportunity, store house features that the buyer seeks and buyer characteristics.	You add columns to the base S_OPTY opportunity table. For more information, see “Process of Configuring the House Detail View” on page 75.
Users might choose values for house attributes from a list rather than enter them. Example attributes include square footage, price range, and so forth.	You add picklists for each field requires predefined values. For more information, see “Creating Static Picklists” on page 109.
Assign opportunities to partner agents according to ZIP code.	You use Assignment Manager to assign opportunities to partner agencies. For more information, see “Customizing How Siebel CRM Assigns Objects” on page 143.
Send an email notification to partner agents when Siebel CRM assigns a new opportunity to the agent.	You use the Business Process Designer to configure Siebel CRM to automatically send email notification. For more information, see “Process of Automating a Business Process” on page 146.

Manage Activities

NREC must manage activities for each opportunity and contact. The predefined Siebel CRM activity management functionality meets most NREC requirements.

Table 4 describes NREC requirements that the predefined Siebel CRM activity management functionality does not meet.

Table 4. Requirements for Activities

Requirement	Configuration Work You Must Perform
Display the opportunity associated with each activity.	You add the opportunity to the Activity List and Form Applets. For more information, see “Adding Fields to Applets” on page 70 .
When entering an activity, allow the user to associate an opportunity to an activity.	<p>You do the following work:</p> <ul style="list-style-type: none"> ■ Create a dynamic picklist that allows the user to update the Opportunity field in the Account Form Applet. ■ Constrain the values in the pick applet so it only displays the opportunities for the account that are associated with the activity. <p>For more information, see “Creating a Dynamic Picklist” on page 111 and “Constraining a Picklist” on page 113.</p>
<p>Allow users to navigate from the Opportunity Detail view to one of the following views, depending on the sales stage:</p> <ul style="list-style-type: none"> ■ Account Detail - Activities (SCW) ■ Account Detail - Contacts (SCW) 	You configure a dynamic drilldown. For more information, see “Creating a Dynamic Drilldown” on page 114 .

Manage Contacts

The predefined Siebel CRM contact management meets NREC requirements.

NREC Development Strategy

NREC uses a specific methodology to develop a Siebel application. This methodology helps NREC minimize risk and realize an incremental return on investment, which increases the probability of project success. It provides the following benefits:

- NREC users and partners can realize the benefits of the project as early as possible.
- The project team can use feedback gathered during the early project stages and incorporate it in the final solution.

This scenario assumes that a design team performed the discover, define and prototype stages, and communicated this information to developers. For more information, see [“About Developing a Siebel Application” on page 11](#).

Siebel CRM Functionality That NREC Plans to Use

NREC plans to use the following functionality:

- Account Management
- Activity Management
- Assignment Manager
- Business Process Designer
- Contact Management
- Opportunity Management
- Partner Management
- Personalization
- Product Information

NREC Design Strategy

This topic describes the NREC design strategy.

Design Strategy for the Data Layer

Figure 7 includes an entity relationship diagram that NREC uses to describe entities and relationships between entities.

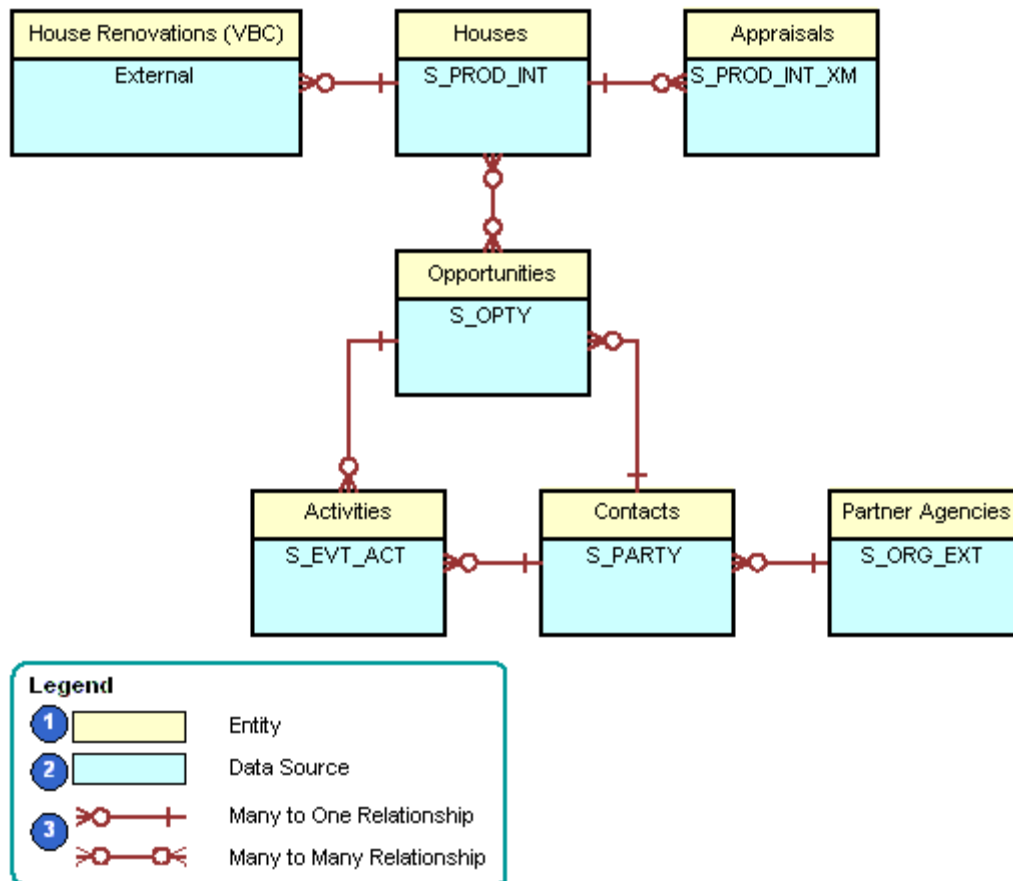


Figure 7. NREC Entity Relationship Diagram

This diagram includes the following information:

- 1 Entity.** A category of data.
- 2 Data source.** Provides the source of data for each entity. Siebel CRM stores most data in tables. For example, house information resides in the S_PROD_INT table. Siebel CRM can derive data from an external source. For example, the House Renovations virtual business component derives data from an external source.
- 3 Relationship.** Indicates the nature of the relationship between entities, such as one-to-one, one-to-many, or many-to-many.

Design Strategy for the Business Object Layer

NREC plans to use the following business objects:

- Accounts
- Activities
- Contacts
- Internal Products
- Opportunities

NREC plans to use the following business components:

- Activity
- Appraisals (new business component)
- Contacts
- Internal Product
- Opportunity
- Partner Company
- Renovations VBC (new virtual business component)

Design Strategy for the User Interface Layer

NREC plans to use several screens for the Partner Portal. Each view contains one or more applets. The Opportunities, Contacts, and Activities screens are predefined in the Partner Portal application. NREC must create a new screen for houses. The views that make up each screen are master-detail views or list-form views. The applets are list applets or form applets.

Figure 8 illustrates part of the user interface layer. NREC will also use other screens, such as Partner Management and Application Administration.

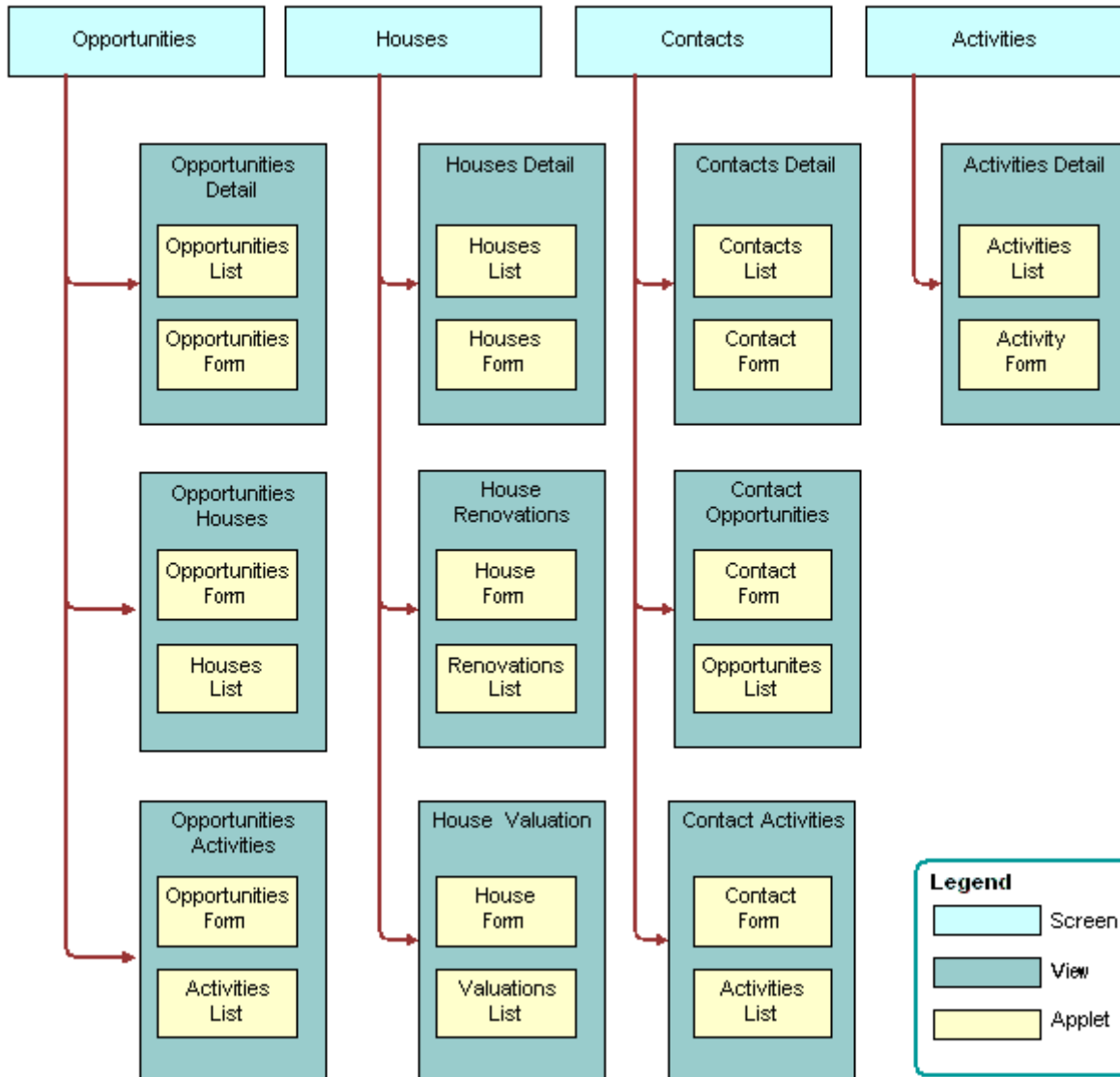


Figure 8. NREC Screens, Views, and Applets

Roadmap for Developing a Siebel Application

To develop a Siebel application, perform the following processes and tasks:

- 1 "Installing Software for a Siebel Application" on page 33
- 2 "Process of Adding Users" on page 36

- 3 "Process of Creating the Company Structure" on page 42
- 4 "Process of Setting Up the Development Environment" on page 51
- 5 "Process of Configuring the User Interface Layer" on page 65
- 6 "Process of Configuring the House Detail View" on page 75
- 7 "Process of Configuring the House Detail - Appraisals View" on page 90
- 8 "Process of Configuring the Opportunity Details View" on page 99
- 9 "Process of Configuring Picklists, Drilldowns, and MVGs" on page 109
- 10 "Process of Configuring Logos, Banners, and Colors" on page 122
- 11 "Process of Configuring Siebel CRM to Use Data From an External Data Source" on page 131
- 12 "Customizing How Siebel CRM Assigns Objects" on page 143
- 13 "Process of Automating a Business Process" on page 146
- 14 "Filtering Records According to a User Profile" on page 155
- 15 "Process of Configuring Siebel Remote" on page 162
- 16 "Process of Moving Data to the Test Environment" on page 167
- 17 "Process of Loading CRM Data in the Test Environment" on page 170

The final step to develop a Siebel application is to move your customizations from the test environment to the production environment. To do this move, you do the following:

- Use Oracle's Siebel Application Deployment Manager (ADM). For more information, see *Siebel Application Deployment Manager Guide*.
- Deploy Siebel CRM to Siebel users. For more information, see *Siebel Installation Guide for Microsoft Windows*.

3

Installing Software, Adding Users, and Creating the Company Structure

This chapter describes how to install software, add users, and create the company structure. It includes the following topics:

- [Installing Software for a Siebel Application on page 33](#)
- [Process of Adding Users on page 36](#)
- [Process of Creating the Company Structure on page 42](#)

Installing Software for a Siebel Application

This task is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

In this topic you install software you use in the development environment. For more information, see [“Overview of the Development Environment” on page 17](#).

To install software for a Siebel application

- 1 Install software you use in the development environment.
For more information, see [“Software You Use in the Development Environment” on page 33](#).
- 2 Install software you use in the test environment.
For more information, see [“Software You Use in the Test Environment” on page 35](#).
- 3 Install software you use in the production environment.
For more information, see [“Software You Use in the Production Environment” on page 36](#).

Software You Use in the Development Environment

[Table 5](#) describes software you use in the development environment. NREC uses Siebel Tools on two computers, one for each developer who configures the Siebel application. There is no limit to the number of developers who can work simultaneously. For more information, see [“Setting Up Local Databases” on page 51](#).

The NREC example uses a Windows environment. For more information, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Table 5. Software You Use in the Development Environment

Product	Description	Installation Instructions
Siebel database schema	Installing and running a database script creates a database schema and enters seed data into some tables. The computer name is DEV_DB_server.	For more information, see <i>Siebel Installation Guide for Microsoft Windows</i>
Siebel File System	A shared directory tree that stores files that the database software does not manage. The computer name is DEV_server.	
Siebel Gateway Server	Software that handles connections between Siebel servers and Siebel clients. The computer name is DEV_server.	
Siebel Server	Application server software. The computer name is DEV_server.	
Siebel Developer Web Client	A client where application layers, except the user interface layer, reside on the Siebel Server. The computer name is DEV_server.	
Siebel Mobile Web Client	A client where application layers reside on the local personal computer. The computer names are DEV_tools_A and DEV_tools_B.	
Sample Database	The Sample database that you use to test, evaluate, and configure your Siebel application. The computer names are DEV_tools_A and DEV_tools_B.	
Database software. For example, IBM DB2, Oracle, or SQL Server.	Third-party relational database software. The computer name is DEV_DB_server.	

Table 5. Software You Use in the Development Environment

Product	Description	Installation Instructions
Web server. For example, Microsoft Internet Information Server or IBM HTTP Server.	Web server that the Siebel Web architecture requires. The computer name is DEV_server.	For more information, see the product documentation for the Web server.
Siebel Tools	An integrated development environment you use to configure a Siebel application. The computer names are DEV_tools_A and DEV_tools_B.	For more information, see <i>Using Siebel Tools</i> .
Microsoft Internet Explorer	The Web browser that displays the interface for the Siebel Web Client or the Siebel Mobile Web Client. The computer names are DEV_client_A and DEV_tools_B.	For More Information, see the Microsoft Web site.

Software You Use in the Test Environment

Table 6 describes software you use in the test environment. The NREC example assumes a Windows environment. For more information, see *Siebel System Requirements and Supported Platforms* on Oracle Technology Network.

Table 6. Software You Use in the Test Environment

Product	Computer Name	Installation Instructions
Siebel database schema	TEST_DB_server	For more information, see <i>Siebel Installation Guide for Microsoft Windows</i>
Siebel File System	TEST_server	
Siebel Gateway Server	TEST_server	
Siebel Server	TEST_server	
Siebel Mobile Web Client	TEST_client_A, and other client computers that you use to test access.	
Database software. For example, IBM DB2, Oracle, or SQL Server.	TEST_DB_server	For more information, see the product documentation for the database software.

Table 6. Software You Use in the Test Environment

Product	Computer Name	Installation Instructions
Web server. For example, Microsoft Internet Information Server or IBM HTTP Server.	TEST_server	For more information, see the product documentation for the Web server.
Microsoft Internet Explorer	TEST_client_B, and other client computers that you use to test access.	For more information, see the Microsoft Web site.

Software You Use in the Production Environment

The software you use in the production environment is typically the same as the software you use in the test environment. When you deploy your Siebel application to users, you copy the contents of one of the client computers to the Siebel Mobile Web Client computers. Also, you send the application URL to users. For more information, see [“Software You Use in the Test Environment” on page 35](#).

Assume that NREC installs the production software on computers with names that it prefixes with PROD. For example, it installs server software on the PROD_server computer.

Process of Adding Users

This process is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

To add users, perform the following tasks:

- 1 [“Creating Database Users” on page 36](#)
- 2 [“Modifying the Configuration File for the Siebel Client” on page 37](#)
- 3 [“Creating Responsibilities” on page 38](#)
- 4 [“Adding Users to a Siebel Application” on page 41](#)

Creating Database Users

This task is a step in [“Process of Adding Users” on page 36](#).

Your database administrator creates database accounts and optionally encrypts user passwords. Depending on your user authentication environment, each user might possess an individual database account, or multiple users might share accounts. If you deploy an external authentication application, then you must configure user authentication including associating each user with a database account.

For NREC, assume the following requirements:

- Set up each user with an individual database account.
- Do not use an external authentication application.

For more information about configuring user authentication, see *Siebel Security Guide*. For more information about creating users, see *Siebel Applications Administration Guide*.

To create database users

- 1 Create a database account for each user listed in “NREC Employees and Users” on page 42:
 - Add the database account on the appropriate database.
 - Add each account to the SSE_ROLE group.
 - For the password, use the same value as the User ID. For example, for Patricia Bosh, use PBOSCH for the User ID and PBOSCH for the password. This technique simplifies development because the developer can enter the same value for the User ID and the password during testing. In an actual deployment, you must set the password according to your security requirements.

The steps to create a database account depends on the database software that your organization uses. Work with your database administrator to perform these tasks.

Types of Users

A Siebel application recognizes several types of users:

- **Employees.** Individual internal users. Employees typically possess login privileges.
- **Partners.** Individual users in partner companies to whom your organization might grant access to data. For more information, see “Creating a Partner Organization” on page 48.
- **Users.** Internal and external users who possess login privileges. These users might include customers who require access to your company data. A user does not require a position to access views.
- **Persons.** Includes employees, partners, users, and contacts. Persons might or might not possess login privileges.

Modifying the Configuration File for the Siebel Client

This task is a step in “Process of Adding Users” on page 36.

You use the Siebel client to add users. To use the Siebel client, you must modify the configuration file for the Siebel client so that it references the correct Siebel database.

To modify the configuration file for the Siebel client

- 1 Use a text editor, such as Notepad, to open the following configuration file:
`C:\ORACLE_HOME\release number\Client_1\BIN\ENU\configuration file`

For example:

```
C:\Siebel\8.2\client_1\BIN\ENU\pmanager.cfg
```

The Siebel client for each Siebel application references a specific configuration file. In the NREC example, you modify the Partner Portal application. The pmanager.cfg file is the configuration file for the Partner Portal application.

2 To set the data source, do one of the following:

- To modify the local database, in the [Siebel] section, make sure the DataSource parameter is set to Local.

In most development environments, you modify the local database, which is located in the following directory:

```
C:\ORACLE_HOME\release_number\client_1\local\sse_data.dbf
```

For example, C:\Siebel\8.2\client_1\local\sse_data.dbf.

- If you are prototyping and prefer not to modify the local database, then you can use the Sample database. Do the following:

- In the [Siebel] section, set the DataSource parameter to Sample.
- In the [Sample] section, set the ConnectString parameter to reference the Sample database.

For example, assume you install the Sample database in the following directory:

```
C:\Siebel\8.2\client_1\
```

In this situation, you set the ConnectString to the following value:

```
C:\Siebel\8.2\client_1\sample\UTF8\sse_samp.dbf -q -m -x NONE -gp 4096 -c40m -ch60m
```

For more information, see [“Prototyping with the Sample Database” on page 63](#).

3 Save and then close the configuration file.

Creating Responsibilities

This task is a step in [“Process of Adding Users” on page 36](#).

A responsibility determines the views a user can access. For example, the Siebel Administrator responsibility allows the user to access all views. Creating responsibilities lets you limit user access to views. If you do not use a predefined responsibility, then you must create a new one.

In this topic, you create responsibilities for NREC employees and partners. You do not create a responsibility for the developer. To perform development tasks, the developer typically must access all views. The predefined Siebel Administrator responsibility provides access to all views. You assign the Siebel Administrator responsibility to the developer.

To create responsibilities

- 1 Log in to the Partner Manager application as an administrator.
For more information, see [“Logging In as the Siebel Administrator”](#) on page 185.
- 2 Navigate to the Administration - Application screen, and then choose Responsibilities.
- 3 Create a responsibility for the NREC CEO:

- a In the Responsibilities list, click Query, enter CEO in the Responsibility field, and then click Go.
You cannot edit a predefined responsibility. You must copy it and then modify it. The CEO predefined responsibility includes most views that the NREC CEO requires. The NREC CEO responsibility must access all views except application administration views.
- b Click Menu and then choose the Copy Record menu item.
Siebel CRM creates a new responsibility. It copies the views from the predefined responsibility, but it does not copy the users from the predefined responsibility.
- c Enter values using information from the following table.

Field	Value
Responsibility	NREC CEO
Description	Leave empty

- 4 Create a responsibility for the NREC Sales Manager:
 - a In the Responsibilities list, click Query, enter Sales Manager in the Responsibility field, and then click Go.
 - b Click Menu and then choose the Copy Record menu item.
 - c Enter values using information from the following table.

Field	Value
Responsibility	NREC Sales Manager
Description	Leave empty

- 5 Create a responsibility for the NREC Agent:
 - a In the Responsibilities list, click Query, enter Universal Agent in the Responsibility field, and then click Go.
The predefined Universal Agent responsibility contains many of the views that the NREC Agent requires. In an actual development project, you must customize the views that each responsibility can access.
 - b Click Menu and then choose the Copy Record menu item.

- c Enter values using information from the following table.

Field	Value
Responsibility	NREC Agent
Description	Leave empty

- 6 Create a responsibility for the NREC Partner Agent:
 - a In the Responsibilities list, click Query, enter Partner Sales Rep in the Responsibility field, and then click Go.
 - b Click Menu and then choose the Copy Record menu item.
 - c Enter values using information from the following table.

Field	Value
Responsibility	NREC Partner Agent
Description	Leave empty

Guidelines for Creating a Responsibility

If you create a responsibility, then it is recommended that you use the following guidelines:

- Define each responsibility to correspond to a major job function in your organization. For example, you might create a separate responsibility for each of the following job functions:
 - NREC CEO
 - Sales manager
 - Partner manager
 - Field sales representatives
- Copy a predefined responsibility and then customize it. Note that you cannot edit a sample responsibility.
- For a developer, add new views to the administrator responsibility. The administrator responsibility provides access to all views. A developer typically helps troubleshoot the Siebel application and requires access to all views.
- Do not provide end users with access to the following views and screens:
 - **System Preferences view.** This view controls server logic and processing. Provide access to the System Preferences to only a particular group of administrators.
 - **Administration - Applications screens.**
 - **Employees, Master Forecasts, Mobile Clients, Responsibilities, Views, and Territories screens.** These screens can affect the entire application.

- **License key view.** To discourage an unauthorized user from attempting to change a license key, you can limit access to the license key view. To do this, you use the License Keys view in the Administration - Applications screen. To hide the license key view, disable the Siebel License Key view in the responsibility for the user.

Adding Users to a Siebel Application

This task is a step in [“Process of Adding Users” on page 36.](#)

In this topic, you add users to a Siebel application.

To add users to a Siebel application

- 1 Navigate to the Administration - User screen, and then the Employees view.
- 2 Add a new record for the CEO:
 - a In the Employees list, click New.
 - b Enter information using values from the following table.

Last Name	First Name	Job Title	User ID	Responsibility
Bosch	Patricia	CEO	PBOSCH	NREC CEO

Siebel CRM automatically adds the Siebel Administrator responsibility. When you add the NREC CEO responsibility, remove the Siebel Administrator responsibility.

For more information, see [“About the User Id Field” on page 42.](#)

- c To add the position, do the following:
 - In the Position field, click the selection icon.
 - In the Positions Occupied dialog box, in the Find list, choose Position.
 - In the Starting With window, enter Siebel Administrator, and then click Go.
 - In the Available window, make sure the Siebel Administrator position is chosen, click Add, and then click OK.

Siebel CRM requires that you add a position. You change this position later. For more information, see [“Creating Positions and Position Skills” on page 44.](#)

- 3 Repeat [Step 2](#) for each of the other users listed in [“NREC Employees and Users” on page 42.](#)

Make sure you do the following:

- For each user, set the Position to Siebel Administrator. For some users, you change this position later. For more information, see [“Creating Positions and Position Skills” on page 44.](#)
- For the sales manager and agent users, make sure you remove the Siebel Administrator responsibility.

About the User Id Field

The value you enter in the User Id field is used to do the following:

- Log in to Siebel Tools
- Identify who checked out a project in Siebel Tools.

If a developer locks a project in Siebel Tools, then Siebel Tools displays the user ID for the account in the Locked By Name property. Make sure the user name clearly identifies the developer. It is recommended that you use a convention for the user name, such as the first initial plus the last name.

The login name must match the database account name for the user. To maintain compatibility across databases, it is recommended that you use all uppercase letters. For example, LCHENG.

NREC Employees and Users

Table 7 lists NREC employees and users.

Table 7. NREC Employees and Users

Last Name	First Name	Job Title	User ID	Responsibility
Bosch	Patricia	CEO	PBOSCH	NREC CEO
Brown	Christine	Technology Manager	CBROWN	Siebel Administrator
Cheng	Li	Developer	LCHENG	Siebel Administrator
Bolivar	Sandy	Administrator	SBOLIVAR	Siebel Administrator
Smith	Lee	Sales Manager	LSMITH	NREC Sales Manager
Castro	Dana	Agent	DCASTRO	NREC Agent

Process of Creating the Company Structure

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To create the company structure, perform the following tasks:

- 1 [“Creating an Organization”](#) on page 43
- 2 [“Creating Positions and Position Skills”](#) on page 44
- 3 [“Creating a Partner Organization”](#) on page 48

Overview of Creating the Company Structure

Because NREC does not use Access Groups or Catalogs, this chapter does not describe how to create them. For more information, see *Siebel Applications Administration Guide* and *Siebel Security Guide*.

Creating the organizational structure for your company includes creating the following items:

- **Organizations and organization skills.** An organization represents broad divisions of internal or external groups of users. For example, internal divisions or partners. An organization controls access to data.
- **Divisions.** A division belongs to an organization but does not affect visibility. A division groups positions and record addresses, and maintains default currencies.
- **Positions and position skills.** A position represents a specific job type. It determines the records a user can access.
- **Responsibilities.** A responsibility controls the screens and views a user can access. For more information, see [“Creating Responsibilities” on page 38](#).

The company structure affects the records and views that Siebel CRM displays to a user. A many-to-many relationship exists between a person and the positions you associate with that person. For example, you can associate a person with the sales representative position for the X territory and the sales manager position for the XYZ region. You can also associate more than one person to a single position. For example, when two people share a single job.

If you associate a user with one responsibility and multiple positions, then that user can access the same set of views regardless of the position the user logs in with. If the user changes position, then the user can access different data. For example, a user who logs in as a sales representative can only access data that pertains to the sales representative position. The user can still access the My Team’s Accounts view, but because no other positions report to the sales representative position, Siebel CRM only displays data that pertains to the sales representative position. In this situation, it does not display data that pertains to the sales manager position. If the user changes to the sales manager position, and if the user is the primary on the account, then the My Team’s Accounts view displays user information for the positions that report to that user.

NOTE: Changing your company structure can cause routers to reevaluate visibility for every object that is related to the objects that you change. This reevaluation can affect performance. For more information, see *Siebel Remote and Replication Manager Administration Guide*.

For more information, see *Siebel Applications Administration Guide* and *Siebel Security Guide*.

Creating an Organization

This task is a step in [“Process of Creating the Company Structure” on page 42](#).

Creating an organization is optional. If you do not create an organization, then Siebel CRM automatically assigns objects to a default organization. For the NREC example, you create one parent organization for NREC and then a child organization for each partner real estate agency. This structure lets NREC logically group different user groups and partition data.

To create an organization

- 1 Navigate to the Administration - Group screen, and then the Organizations view.

- 2 Add a new record using values from the following table.

Field	Description
Name	Enter NREC Organization. This value must be a unique name for the organization. To provide a visual clue, it is recommended that you include the word <i>organization</i> in the name.

Siebel CRM automatically creates a division when you create an organization. For example, when you create the NREC Organization, Siebel CRM automatically creates the NREC division. Note that you cannot delete an organization that you create.

- 3 Examine the division that Siebel CRM created for your organization:
 - a Click the Internal Divisions tab.
 - b In the Divisions list, click Query, enter NREC Organization in the Division Name field, and then click Go.

You created NREC Organization in [Step 1](#).
 - c Examine the values defined for the division.

A division belongs to an organization. It stores addresses, maintains default currencies, and defines the country of operation. A Siebel application must include at least one division. NREC only requires one division.

Creating Positions and Position Skills

This task is a step in [“Process of Creating the Company Structure” on page 42](#).

A *position* represent an actual job position in your organization. It determines the records that Siebel CRM displays to each user. A position can be a formal job title, such as the actual job title that the developer uses. It can also be any descriptive text. For example, Developer 1, Developer 2, and so forth.

To log in to a Siebel application, an employee must be assigned a position. When you create positions and position skills, it is recommended that you do the following:

- To determine reporting relationships between positions and parent positions, refer to the organization chart for your company.
- Because you choose parent positions as you create new positions, start at the top of the organization chart and work your way down.

Note that there is typically one position that does not include a parent position. For example, the CEO position typically does not report to a higher level.

Siebel CRM includes predefined positions that ship with the Siebel product. However, none of the predefined positions meet the requirements of the developer role in the NREC example, so you must create a new position for the developer. For more information about positions, see *Siebel Applications Administration Guide*.

Figure 9 illustrates the NREC position hierarchy.

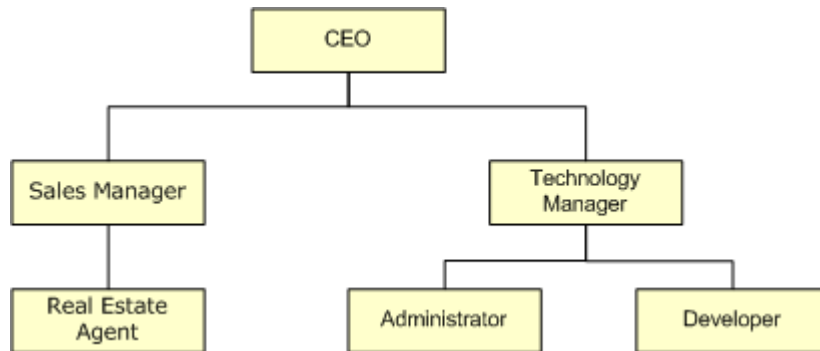


Figure 9. NREC Position Hierarchy

To create positions and position skills

1 Create positions:

- a In the Siebel client, navigate to the Administration - Group screen, and then the Positions view.
- b In the Positions list, add a new record using values from the following table. For all positions that you add in this step, enter NREC Organization in the Division field.

Field	Value
Position	NREC CEO
Parent Position	Leave empty
Last Name	Bosch
First Name	Patricia

- c In the Positions list, add a new record using values from the following table.

Field	Value
Position	NREC Technology Manager
Parent Position	NREC CEO
Last Name	Brown
First Name	Christine

- d In the Positions list, add a new record using values from the following table.

Field	Value
Position	NREC Developer
Parent Position	NREC Technology Manager
Last Name	Cheng
First Name	Li

- e In the Positions list, add a new record using values from the following table.

Field	Value
Position	NREC Administrator
Parent Position	NREC Technology Manager
Last Name	Bolivar
First Name	Sandy

- f In the Positions list, add a new record using values from the following table.

Field	Value
Position	NREC Sales Manager
Parent Position	NREC CEO
Last Name	Smith
First Name	Lee

- g In the Positions list, add a new record using values from the following table.

Field	Value
Position	NREC Sales Agent
Parent Position	NREC Sales Manager
Last Name	Castro
First Name	Dana

- 2 In the employee record, adjust the position for the CEO:

- a Navigate to the Administration - Users screen, and then the Employees view.
- b In the Employees list, click Query, enter Bosch in the Last Name field, and then click Go.
- c In the Position field, click the selection icon.

- d In the Selected window, click the NREC CEO position, and then make sure the Primary field for the NREC CEO position contains a check mark.
 - e In the Selected window, click the Siebel Administrator position.
 - f Click Remove, and then click OK.
- 3 Repeat [Step 2](#) for each of the following employees:
- Christine Brown
 - Li Cheng
 - Sandy Bolivar
 - Lee Smith
 - Dana Castro
- 4 Log out of the Siebel client.
- 5 Verify your work:
- a Log in to the Siebel client. In the login screen, enter values using information from the following table.

Field	Value
User ID	LCHENG
Password	LCHENG

To make sure you possess system administrative access, navigate to an administrative view, such as Administration - User, and then the Employees view.

- b Log out of the Siebel client, and then log back in using values from the following table.

Field	Value
User ID	PBOSCH
Password	PBOSCH

- c To make sure you do not possess system administrative access, attempt to navigate to an administrative view, such as Administration - User, and then the Employees view.

You must not be able to view an administrative view.

- 6 Create position skills:
- a Navigate to the Administration - Group screen, and then the Positions view.
 - b In the Positions list, click the position to which to add skills.
 - c Click the Assignment Skills tab.
 - d In the Assignment Skills list, add a new record.

- e Choose a skill in the Item field.
You must use Siebel Tools to add skills before you can add them to a position.
- f Add comments, if required.
- g Save the record.

Creating a Partner Organization

This task is a step in “[Process of Creating the Company Structure](#)” on page 42.

For the NREC example, you register only one partner, San Francisco Real Estate. You create an organization and set up the company structure for this partner. For more information, see *Siebel Partner Relationship Management Administration Guide*.

To create a partner organization

- 1 Add employees for the partner organization:
 - a Navigate to the Administration - User screen.
 - b In the Users list, click New.
 - c Enter information using values from the following table.

Last Name	First Name	Job Title	User ID	Responsibility
Singh	Maya	NREC Partner Agent	MSINGH	NREC Partner Agent

Siebel CRM automatically adds the Siebel Administrator responsibility. When you add the NREC CEO responsibility, remove the Siebel Administrator responsibility.

- 2 Register the partner and promote it to an organization:
 - a Navigate to the Administration - Partner screen, and then the Qualified Partner screen.
 - b In the Qualified Partners list, click New, and then enter values from the following table.

Field	Value
Name	San Francisco Real Estate

- c Click Register.

- d In the Register Partner form, set values using information from the following table, and then click Save.

Field	Value
Organization	Contains a check mark
Parent Organization	NREC Organization

If you click Save, then Siebel CRM adds the partner to the Registered partners list and creates an organization for the partner as a child of the parent NREC Organization.

- 3 In the Registered Partners list, in the Name field, click the San Francisco Real Estate link.
- 4 In the Responsibilities list, click Menu, choose the New Record menu item, and then set values using information from the following table.

Field	Value
Responsibility	NREC Partner Agent

- 5 Click the Positions tab, click New, and then set values using information from the following table.

Field	Value
Division	San Francisco Real Estate
Position	NREC Partner Agent

- 6 Click the User Assignments tab, click New, and then set values using information from the following table.

Field	Value
Last Name	Singh
First Name	Maya
Job Title	NREC Partner Agent

4

Setting Up the Development Environment

This chapter describes how to set up the development environment. It includes the following topic:

- [Process of Setting Up the Development Environment on page 51](#)

Process of Setting Up the Development Environment

This process is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

To set up the development environment, perform the following tasks:

- 1 [“Setting Up Local Databases” on page 51](#)
- 2 [“Setting Up Siebel Tools” on page 60](#)

Setting Up Local Databases

This task is a step in [“Process of Setting Up the Development Environment” on page 51](#).

The NREC example requires that each developer use a local database. The process to set up a local database includes tasks that an administrator performs and tasks that the developer performs. For example, to generate a local database, an administrator creates the database user and runs Siebel Remote server tasks. To initialize the local database, each developer downloads repository data from the Siebel Server.

For more information about each step in this task, see *Siebel Remote and Replication Manager Administration Guide*.

NOTE: To complete the examples in [Chapter 5, “Configuring the User Interface”](#) through [Chapter 8, “Configuring Logos, Banners, and Colors”](#) instead of setting up and using a local database you can use the Sample database. For more information, see [“Prototyping with the Sample Database” on page 63](#).

NOTE: For Siebel Innovation Pack 2015, the local database for the Siebel Mobile Web Client uses SAP SQL Anywhere. SAP SQL Anywhere is not available for new deployments after September 2015. For more detailed information on how this change affects Siebel Tools and Siebel Remote, see *Siebel Release Notes* on Siebel Support for Innovation Pack 2015 (Doc ID 1996273.1).

To set up local databases

- 1 Log in to the Partner Manager application as an administrator.
For more information, see [“Logging In as the Siebel Administrator” on page 185](#).
- 2 Make sure the Partner Manager application is connected to a Siebel server.

- 3** Register the developer as a new Siebel Mobile Web Client:
 - a** Navigate to the Administration - Siebel Remote screen, and then the Mobile Clients view.

- b In the Mobile Clients list, click New, and then complete fields using values from the following table.

Field	Description
Mobile Client	<p>Enter LCHENG.</p> <p>It is recommended that you enter the Mobile Client name in uppercase letters, with a maximum of eight characters. It is recommended that you use the User ID.</p> <p>CAUTION: The Mobile Client name and the User ID must contain only Roman characters, alphanumeric characters, the underscore (_) or the dash (-). It cannot include spaces, periods, or other special characters. For more information, see the topic about format of the remote client name, database name, login id, and windows password in <i>Siebel Remote and Replication Manager Administration Guide</i>.</p>
User ID	<p>Choose the LCHENG user ID that is assigned to Li Cheng.</p> <p>The user uses the User ID to log in to the local database during initialization. Siebel Remote uses the User ID to access the local database during initialization and synchronization.</p>

Field	Description
Routing Model	<p>Choose MOBILE CLIENT - EXTRACT ONLY.</p> <p>Siebel Remote associates each mobile user with one Routing Model. This model includes a set of routing rules that determines if Siebel CRM routes a record to a Siebel Mobile Web Client.</p> <p>For a developer, it is recommended that you use the MOBILE CLIENT - EXTRACT ONLY routing model. It does not allow synchronization. It provides only a snapshot of data, which is appropriate for a developer.</p>
Language(s)	<p>Do the following:</p> <ul style="list-style-type: none"> ■ In the Language(s) field, click the select button. ■ In the Languages dialog box, click New. ■ In the Language field, click the select button. ■ In the Pick Language Name dialog box, Choose a language, and then click OK. For example, choose English-American. ■ Click Save, and then click OK. <p>Siebel CRM uses a language preference for the mobile user to download data in a preferred language for the following dock objects. These objects contain translation tables. This technique helps to optimize the size of the local database:</p> <ul style="list-style-type: none"> ■ LOV ■ Product ■ Literature ■ Catalog ■ Catalog Category <p>If you leave the Language field empty for a Mobile Client, then Siebel CRM routes all language translations for these dock objects to the local database that resides on the Mobile Client. If you specify a language, then you must add English-American - ENU, and then each of the other languages that your deployment requires.</p>

Although Siebel CRM does not use Siebel Remote to check in and check out projects, you must register each developer as a Siebel Mobile Web Client. This registration provides Siebel Remote the information it requires to create a local database file for the developer and to enter appropriate data in the file.

- 4 Make sure the Remote Server component group is enabled:
 - If you enabled Siebel Remote when you installed the Siebel Server, then the Siebel Remote component group is enabled.

- If you did not enable Siebel Remote when you installed the Siebel Server, then you must enable it. For more information, see [“Enabling the Siebel Remote Component Group” on page 56](#).

5 Generate a new database:

- a** Navigate to the Administration - Server Management screen, and then the Jobs view.
- b** In the Jobs list, click New.
- c** In the Component/Job field, choose Generate New Database.
- d** In the Requested Server field, enter the name of the Siebel Server on which to run the GenNewDb job.
- e** In the Job Parameters list, click New.
- f** In the Name field, click the select button, locate the SQL Anywhere Database job parameter, and then click OK.

The Siebel Server defaults the Value field to sse_utf8.dbf. To use an encrypted template file, replace this value with sse_encr.dbf.

- g** (Optional) To modify the values of other optional parameters, repeat [Step e](#) through [Step f](#) for the other parameter.

For more information, see [“Parameters for the Generate New Database Server Component” on page 58](#).

- h** In the Jobs list, click Submit Job.

The Siebel Server generates a new database file, which typically requires a few minutes.

6 Make sure the Generate New Database task finishes:

- a** Click Tasks.
- b** In the Tasks list, click Query, enter Generate New Database in the Component field, and then click Go.
- c** Make sure the value in the State field is Completed.
- d** If the value in the State field is not Completed, then wait a few minutes, and then repeat [Step b](#).
If necessary, repeat [Step d](#) until the value in the State field is Completed.

7 Extract the local database:

- a** Click Jobs.
- b** In the Jobs list, click New.
- c** In the Component/Job field, choose Database Extract.

- d In the Job Parameters list, add a new parameter for each row in the following table.

Parameter	Value
Client Name	Enter LCHENG. This is the name of the Mobile Client you entered in Step b on page 53 .
Encrypt client Db password	Enter Fal se.
Client Database encryption method	Enter none.

The Job Parameters list is located below the Jobs list and below the Job Detail form. The encryption parameters allow you to extract a local database without encryption. For more information, see 475398.1 (Article ID) on My Oracle Support.

- e In the Jobs list, click Start.

Siebel CRM extracts the database for the Siebel Mobile Web Client. This might take a few minutes
- 8 Verify that the database extraction created the required files:
 - a On the Siebel Server computer, open Windows Explorer, and then navigate to the following directory:

```
si ebel \docki ng\l cheng
```
 - b Verify that the following items exist in the si ebel \docki ng\l cheng directory:
 - The dobjinst.dbf file
 - Inbox and outbox directories
 - c Open the outbox directory, and then verify that at least one file exists for each of the following file extensions:
 - .uaf
 - .dat
 - .toc

For more information, see [“How the Database Extract Server Component Extracts Data” on page 59](#).
- 9 Log out of the Siebel client.

Enabling the Siebel Remote Component Group

You must make sure the Siebel Remote component group is enabled. For more information, see *Siebel System Administration Guide*.

To enable the Siebel Remote component group

- 1 Enable the Siebel Remote component group:

- a** Navigate to the Administration - Server Configuration screen, Enterprises, and then the Component Groups view.
- b** In the Component Groups list, click Query, enter Siebel Remote in the Component Group field, and then click Go.
- c** If the value in the Enable state field is not Enabled, then click Enable.
- d** In the Component Group Assignments list, make sure the Enabled on Server option contains a check mark.
- e** If you clicked Enable in [Step c](#), or if you added a check mark in [Step d](#), then log out of the Siebel application, restart the Siebel Server, and then log back in.

For more information, see [“Enabling the Siebel Remote Component Group” on page 56](#).

2 Make sure the Siebel Remote server components are online:

- a** Navigate to the Administration - Server Management screen, and then the Components view.
- b** In the Components list, click Query, enter Database Extract in the Component field, and then click Go.
- c** Make sure the value in the State field is Online.
- d** If the server component is not online, then use the buttons in the Components list to restart the server component.
For example, click Startup, or Resume.
- e** Repeat [Step b](#) for the Generate New Database server component.
- f** Repeat [Step b](#) for the Synchronization Manager server component.

3 Synchronize the server components:

- a** Navigate to Administration - Server Configuration screen, and then the Enterprises view.
- b** Click the Synchronize tab.
- c** In the Synchronize list, click Synchronize.
- d** Wait for Siebel CRM to complete the synchronization.

This might require a few minutes.

Parameters for the Generate New Database Server Component

Table 8 lists the parameters and default values for the Generate New Database server component.

NOTE: For Siebel Innovation Pack 2015, the local database for the Siebel Mobile Web Client uses SAP SQL Anywhere. SAP SQL Anywhere is not available for new deployments after September 2015. For more detailed information on how this change affects Siebel Tools and Siebel Remote, see *Siebel Release Notes on Siebel Support for Innovation Pack 2015 (Doc ID 1996273.1)*.

Table 8. Parameters for the Generate New Database Server Component

Name	Alias	Description
SQL Anywhere Database	DbfFile	The SQL Anywhere database filename to initialize. The default value is sse_utf8.dbf, which is a Unicode file that supports all languages. Alternatively, you can use sse_encr.dbf, which provides Sybase encryption of the local database template. The SQL Anywhere Database parameter is a required parameter.
DBA Password	DbnPwd	Password for the DBA account. The default value for SQL Anywhere is SQL. Set the DBA Password parameter to the password for the DBA user ID in the empty database template file.
New DBA Password	NewDbnPwd	The password assigned to the local database administrator account on the Siebel Mobile Web Client. The Siebel Server uses the parameter value you specify only for the current run of the Generate New Database component. If you do not specify a value for the New DBA Password parameter, then the Siebel Server does the following: <ul style="list-style-type: none"> ■ Uses a value specified in the administrative view. ■ If you do not specify a value in the administrative view, then the Siebel Server sets the value for the New DBA Password to the first 8 characters of the enterprise name. If the enterprise name includes less than 8 characters, then the Siebel Server pads the name with consecutive digits. For example, If the enterprise name includes ABCD, then the Siebel Server sets the name to ABCD1234, where 1234 are consecutive digits.
Table Space	TSpace	Space name in the database template that stores Siebel tables. Do not specify the Table Space parameter unless you intend to build a custom empty database file that uses this table space.
Index Space	ISpace	Space name in the database template that stores Siebel indexes. Do not specify the Index Space parameter unless you intend to build a custom empty database file that uses this index space.
Use Transaction Log File	UseTxnLog	Use if you create a new template file. The default value is TRUE.

Table 8. Parameters for the Generate New Database Server Component

Name	Alias	Description
Use DDL File	UseDdlFile	<p>Use this parameter if you create a new template file. Note the following values:</p> <ul style="list-style-type: none"> ■ FALSE. The Siebel Server reads the schema directly from the server database. This is the default value. ■ TRUE. The Siebel Server reads the schema from a DDL file. <p>If a schema change occurs in your environment, then set the Use DDL File parameter to FALSE. This value directs the Generate New Database server component to read the latest schema from the database, and not the DDL file. Set the Use DDL File parameter to FALSE only if you run the Generate New Database component after a schema change.</p>
Interface Tables	IFaceTbls	Creates interface tables and indexes. The default value is FALSE.
Warehouse Tables	WarehouseTbls	Creates warehouse tables and indexes. The default value is FALSE.
Client Db Type	ClientDbType	Sets the client database engine type. The default value is SQL Anywhere.

How the Database Extract Server Component Extracts Data

The database extract process retrieves data that is visible to the Remote user from the server database. It retrieves data according to routing rules that define the level of access for the user. It creates compressed files that contain data that Siebel CRM loads in the local database when the user initializes the laptop. Note that, for the database extract to be successful, the mobile user must be associated with a valid position. For more, see *Siebel Applications Administration Guide*.

Each registered Siebel Mobile Web Client requires a separate directory on the Siebel Remote server. The Database Extract job creates these directories for each Siebel Mobile Web Client. It also places a directory named `txnproc` in the docking directory in the Siebel server root directory. Do *not* modify the contents of this directory under any circumstances.

The following example displays a portion of the server directory tree after you run Database Extract for Siebel Mobile Web Clients and developers named LCHENG and SSCOTT.

```

siebel
  docking
    lcheng
      inbox
      outbox
    sscott
      inbox
      outbox
    txnproc

```

Running the Database Extract job creates a database snapshot for a given user, which includes multiple files. These files contain the data the Siebel CRM requires to initialize the local database for the user. Siebel CRM places them in the following directory on the development server:

```
siebsrvr_root\docki ng\user\outbox
```

To create a local database and local file system, each mobile user downloads these files.

Setting Up Siebel Tools

This task is a step in “[Process of Setting Up the Development Environment](#)” on page 51.

In this topic you set up Siebel Tools. For more information, see *Using Siebel Tools*.

To set up Siebel Tools

1 Modify the Siebel Tools configuration file:

a Open the tools.cfg file in a text editor.

The tools.cfg file is located in the following directory:

```
C:\ORACLE_HOME\BI\N\ENU
```

For example, C:\Siebel\8.2\Tools_1\BI\N\ENU.

b Edit parameters using values from the following table.

Parameter	Description
EnableToolsConstrain	Set to FALSE. For more information, see “ Configuring Text for Labels and Other User Interface Elements ” on page 80.
DataSource	Do one of the following: <ul style="list-style-type: none">■ If you use a local database, then set to Local.■ If you use the Sample database, then set to Sample.
ConnectionString	Set to the same database that the Siebel client references. For more information, see “ Modifying the Configuration File for the Siebel Client ” on page 37.
ClientConfigurationMode	Make sure the ClientConfigurationMode parameter is not All. You cannot use the Form Applet Wizard, List Applet Wizard, View Wizard, or set the HTML Sequence if the ClientConfigurationMode parameter is All.
EncryptApplicationPassword	Set to FALSE. Password encryption must be disabled when you check projects in or out. For more information, see “ Process of Modifying Siebel Objects ” on page 183.
EncryptCredentialsPassword	

c Save and then close the tools.cfg file.

2 Initialize the local database:

a Establish a LAN connection between the Siebel Remote server and the remote computer.

Because the amount of information that Siebel CRM must download from the Siebel server to initialize the database for Siebel Mobile Web Client is substantial, it is recommended that you use a LAN connection rather than a modem or WAN between the server and the Siebel Mobile Web Client. Alternatively, you can initialize the local database from a CD-ROM or other media.

b On the local computer that the developer uses, start Siebel Tools.

In the Siebel Tools login dialog box, in the Connect to window, choose Local.

Siebel Tools displays the following message:

The local Siebel database was not found. Would you like to connect to the Siebel Remote server to initialize the local database?

c Click Yes.

Siebel Tools connects to the Siebel Remote server and initializes the local database. If initialization is successful, then Siebel Tools creates a local database file named sse_data.dbf, and stores it in the following directory:

C:\ORACLE_HOME\local directory

where:

- ORACLE_HOME is the root directory of where you installed Siebel Tools. For example, c:\siebdev.

If you use the Sample database, then it is not necessary to initialize the local database. For more information, see ["Prototyping with the Sample Database" on page 63](#).

3 Perform a full get:

a Choose the Tools menu, and then the Check Out menu item.

b In the Check Out dialog box, choose the name of your development repository from the Repository list.

c Click All Projects, and then click Options.

d In the Development Tools Options dialog box, make sure the following settings exist:

Option	Value
Server Data Source	References the server development database.
Client Data Source	References the local database you initialized in Step 2 .

These data sources must match the values displayed in the Control Panel under ODBC Data Sources.

- e In the Check Out dialog box, click Get.

For more information, see [“About the Full Get” on page 63](#).

4 Set the target browser:

- a Choose the View menu, Toolbars, and then the Configuration Context menu item.
- b In the Target Browser list of the Configuration Context toolbar, choose Target Browser Config.
- c In the Target Browser Configuration dialog box, move the target browser to the Selected pane window, as necessary.

For example, move MSIE 7.0.

- d Click OK.

You use the Target Browser feature to configure your Siebel application for different browsers, such as Internet Explorer or Firefox. Although you do not configure for different browsers in this book, Siebel Tools requires that you set a target browser. Otherwise, you can not open the Web Layout Editor. For more information, see [“Opening the Web Layout Editor” on page 180](#).

5 Make sure the Application list on the Configuration Context toolbar is set to All Applications.

6 Display object types you use to configure Siebel CRM:

- a Open Siebel Tools.
- b Choose the View menu, and then the Options menu item.
- c Click the Object Explorer tab.
- d Scroll down through the Object Explorer Hierarchy window until you locate the object you must display.
- e Make sure the tree for the object you must display and all child objects of the tree include a check mark. For the example in this book, display the following object types:

- Bitmap Category
- Business Component User Prop
- EIM Interface Table
- Symbolic String
- Workflow Process

If all child objects of an object type are displayed, then Siebel Tools displays a black check mark with a white background for the tree.

- f Click OK.

7 Create a new project:

- a In Siebel Tools, in the Object Explorer, click Project.
- b In the Projects list, right-click, and then choose the New Record menu item.

- c Enter a new record using values from the following table.

Property	Value
Name	NREC Configuration
Locked	TRUE

You use this project during development to group object definitions that you create or modify. This grouping simplifies the task of finding your work and exporting and importing the object definitions.

- 8 Set the default HTML Editor:
 - a If an HTML editor is not installed on your computer, then install one now.
To allow you to edit HTML code, Siebel Tools uses a third party HTML editor. This HTML editor is a third-party application that you must install on your computer. Notepad++ is an example of a third-party HTML editor.
 - b Choose the View menu, and then the Options menu item.
 - c Click the Web Template Editor tab.
 - d In the Executable Full Path window, enter the full path to the executable file for your HTML editor.
For example, C:\Program Files\Notepad++\notepad++.exe.
- 9 Choose the View menu, Toolbars, and then make sure the WF/Task Editor Toolbar menu item contains a check mark.

About the Full Get

A *full get* is the process of populating a local database with a read-only copy of the projects and object definitions that are stored in the server database. When you do a full get, Siebel Tools copies all objects from the server repository to the local repository. The local repository that is currently open includes the same data as the server repository from which you performed the full get.

A full get is equivalent to checking out every project from the Siebel Server, however the projects are not locked and you cannot modify the object definitions until you check out a project from the server database or lock a project locally. Having a read-only copy of the full repository on your local computer allows you to compile a Siebel repository file for your local database. This is required to test changes locally.

If you use the Sample database, then it is not necessary to perform a full get. For more information, see [“Prototyping with the Sample Database” on page 63](#).

Prototyping with the Sample Database

The Sample database contains example data that you can to quickly prototype various design ideas. To use the Sample database, it is not necessary to set up a local database. Note that the Sample database cannot receive a checked out object, and you cannot check in objects from the Sample database to the server database.

5

Configuring the User Interface

This chapter describes how to configure the user interface. It includes the following topics:

- [Process of Configuring the User Interface Layer on page 65](#)

Process of Configuring the User Interface Layer

This process is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

To configure the user interface layer, perform the following tasks:

- 1 [“Removing Screens from a Siebel Application” on page 65](#)
- 2 [“Removing Views from a Screen” on page 67](#)
- 3 [“Removing Fields from an Applet” on page 68](#)
- 4 [“Adding Fields to Applets” on page 70](#)
- 5 [“Verifying Your Configuration” on page 74](#)

To begin developing the NREC example, you configure the user interface layer. Completing tasks in this topic helps you understand the hierarchy of applications, screens, views, and applets, and provides you with experience in performing configuration tasks.

Removing Screens from a Siebel Application

This task is a step in [“Process of Configuring the User Interface Layer” on page 65](#).

NREC deployment of the Siebel Partner Portal application does not require every screen that comes with the predefined product, such as the Service, Solutions, and Campaigns screens. These screens are not part of the NREC requirements, and NREC does not plan to use them in the future.

Siebel CRM displays screens in the Siebel client as screen tabs and as links on the Site Map. The following objects control these tabs and links:

- **Page Tab.** Controls screen tabs in screens in the Siebel client.
- **Screen Menu Item.** Controls links in the Site Map.

To remove these screens in the NREC example, you disable the Page Tab and Screen Menu Item objects.

As an alternative, you can create a responsibility that does not include any of the views in the screen, and then assign users to that responsibility. This technique removes the screens from the Siebel client but leaves the screen objects active in the repository. It allows you to add the screens back in the Siebel client without having to recompile the repository file. For more information, see [“Creating Responsibilities” on page 38](#).

To remove screens from a Siebel application

- 1 In the Object Explorer, click Application.
- 2 In the Applications list, locate the Siebel eChannel application.

The Siebel eChannel application object represents the Partner Portal application. The employee application is Siebel Partner Manager.
- 3 Choose the Tools menu, and then the Lock Project menu item.

Siebel Tools locks the Siebel eChannel project, which is the project to which the application belongs. Siebel Tools displays a pencil icon in the W column. This icon indicates that you can modify the record.
- 4 Change the value in the Project property to NREC Configuration.

This step associates the application object with the NREC project, which simplifies how you manage changes.
- 5 In the Object Explorer, expand the Application tree, and then click Page Tab.
- 6 In the Page Tabs list, make sure the check box in the Inactive property contains a check mark for all page tabs, except for the Opportunities Screen (SCW) page tab.

Siebel Partner Portal uses the SCW designation for many screens, views, applets, and so forth. Siebel Tools uses red font to indicate inactive Page Tabs.
- 7 In the Object Explorer, in the Application tree, click Screen Menu Item.
- 8 In the Screen Menu Items list, enter the following query in the Screen property:

```
<> "Accounts Screen (SCW)" AND <> "Activities Screen (SCW)" AND <> "Contacts Screen (SCW)" AND <> "Opportunities Screen (SCW)"
```

This query returns only those screens that NREC does not require.
- 9 Make sure the Inactive property contains a check mark for each record that Siebel Tools displays in the Screen Menu Items list.
- 10 Compile your changes.

For more information, see [“Compiling Projects” on page 184](#).
- 11 To test your changes, open the Partner Portal application and make sure Siebel CRM only displays screen tabs and links for the following screens:
 - Accounts
 - Activities
 - Contacts

- Opportunities

Removing Views from a Screen

This task is a step in [“Process of Configuring the User Interface Layer” on page 65](#).

Each screen in a Siebel application includes a set of views. The screen view object type is a child of the screen object type, and it creates an association between a view and a screen. Siebel CRM displays views in the Siebel client below the screen tabs. Some views include data or contain functionality that NREC does not require. For example, NREC does not require certain views that are associated with the Opportunities Screen, such as Attachments, Presentations, Proposals, Quotes, and Sales Teams. To remove these views in the NREC example, you make sure the Inactive property for each view contains a check mark.

To remove a view from a Siebel application, you can also do one of the following:

- Create a responsibility for NREC that includes only the views that NREC requires. For more information, see [“Creating Responsibilities” on page 38](#).
- Use Siebel Personalization to display or hide views according to user profile attributes. For more information, see *Siebel Personalization Administration Guide*.

To remove views from a screen

- 1 In the Object Explorer, click Screen.
- 2 In the Screens list, locate the Accounts Screen (SCW) screen.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 In the Object Explorer, expand the Screen tree, and then click Screen View.
- 6 In the Screen Views list, make sure the Inactive property contains a check mark for each of the following views:
 - Account Detail - Account Team View (SCW)
 - Account Detail - Assets View (SCW)
 - Account Detail - Orders View (SCW)
 - Account Detail - Projects View (SCW)
 - Account Detail - Quotes View (SCW)
 - Account Detail - Revenue Schedule Chart View (SCW)
 - Account Detail - Revenue Schedule View (SCW)
 - Account Detail - Revenue Schedule View DC (SCW)
 - Account Detail - Service Requests View (SCW)
 - Account Presentations View (SCW)

- Account Proposals View (SCW)
 - SI Com Account Briefing View (eApps)
- 7 Repeat [Step 2](#) through [Step 6](#), but with the following differences:
- In the Screens list, locate the Activities Screen (SCW) screen.
 - In the Screen Views list, make sure the Inactive property contains a check mark for the FS Activity Part Movements view.
- 8 Repeat [Step 2](#) through [Step 6](#), but with the following differences:
- In the Screens list, locate the Contacts Screen (SCW) screen.
 - In the Screen Views list, make sure the Inactive property contains a check mark for the Contact Detail - Service Request View (SCW) view.
- 9 Repeat [Step 2](#) through [Step 6](#), but with the following differences:
- In the Screens list, locate the Opportunities Screen (SCW) screen.
 - In the Screen Views list, make sure the Inactive property contains a check mark for each of the following views:
 - Opportunity Detail - Attachments View (SCW)
 - Opportunity Detail - Products View (SCW)
 - Opportunity Detail - Quotes View (SCW)
 - Opportunity Detail - Revenue Schedule Chart View (SCW)
 - Opportunity Detail - Revenue Schedule View (SCW)
 - Opportunity Detail - Revenue Schedule View DC (SCW)
 - Opportunity Presentations View (SCW)
 - Opportunity Proposals View (SCW)
 - Oppty Chart View - Campaign Analysis
 - Oppty Chart View - Campaign Pipeline Analysis
 - Oppty Chart View - Current Opportunity Analysis
 - Oppty Chart View - Pipeline Analysis
 - Oppty Chart View - Quality Analysis by Rep
- 10 Compile and test your changes.

For more information, see [“Compiling Projects” on page 184](#), and [“Registering a New View” on page 186](#).

Removing Fields from an Applet

This task is a step in [“Process of Configuring the User Interface Layer” on page 65](#).

You use the Web Layout Editor to remove fields from an applet. For more information, see [“Opening the Web Layout Editor” on page 180](#).

To remove fields from an applet

- 1 In the Object Explorer, click Applet.
- 2 In the Applets list, locate the Activity Detail Applet (SCW) applet.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 Make sure you set up the target browser and the application on the Configuration Content toolbar.
For more information, see [“Setting Up Siebel Tools” on page 60](#).
- 6 In the Applets list, right-click the Activity Detail Applet (SCW) applet, and then choose the Edit Web Layout menu item.
- 7 In the Controls/Columns window, make sure the mode is set to Edit.
- 8 In the Web template, delete the Account field:
 - a Right-click the label for the Account field, and then choose the Delete menu item.
 - b Right-click the control for the Account field, and then choose the Delete menu item.
For more information, see [“About Removing a Field” on page 69](#).
- 9 Repeat [Step 8](#), but this time remove the Duration field.
- 10 Repeat [Step 8](#), but this time remove the % Complete field.
- 11 Move the End field to the position that the Duration field formerly occupied. To determine where to place the field, see the mock-up in [Figure 10 on page 71](#).
- 12 Choose the File menu, and then the Save menu item.
- 13 (Optional) To view an approximation of how Siebel CRM renders the applet in the Siebel client, right-click anywhere in the Web template, and then choose the Preview menu item. To turn off the preview, right-click and then choose Preview again.
- 14 Choose the File menu, and then the Close menu item.

About Removing a Field

To remove a field from an applet, you must remove the following objects for the field:

- **Label.** References a symbolic string.
- **Control.** References a field in the Siebel database.

When you remove a field from the Web template, Siebel Tools unbinds the list column or control from the Web template. The objects remain intact as child objects of the applet, but Siebel CRM does not render them in the Siebel client.

Adding Fields to Applets

This task is a step in [“Process of Configuring the User Interface Layer”](#) on page 65.

To add fields to applets, you do the following:

- [“Adding a Field to a List Applet”](#) on page 72
- [“Adding a Field to a Form Applet”](#) on page 73

NREC requires the following applets in the Activities view:

- Activity List Applet
- Activity Detail Applet

You must modify these applets to meet the following NREC business requirements:

- Display fields in a particular order.
- Provide partner agents the ability to track which opportunity is associated with each activity. This functionality is not available in the predefined application. Therefore, you must add the Opportunity field to the Activity List Applet and the Activity Detail Applet applet.

Figure 10 illustrates a view mock-up. You will develop a view that resembles this mock-up.

The mock-up is titled "Activities Detail View" and is enclosed in a rectangular border. It is divided into two main sections:

- Activities Section:** A tab labeled "Activities" is positioned above a table. The table has seven columns: "New", "Type", "Description", "Agency", "Assigned to", "Opportunity", and "Assigned to". There are four empty rows below the header.
- More Info Section:** A tab labeled "More Info" is positioned above a form. The form contains several input fields:
 - Row 1: "New" (checkbox), "Agency" (text box), "Status" (text box)
 - Row 2: "Type" (text box), "Assigned to" (text box), "Due" (text box)
 - Row 3: "Description" (text box), "Opportunity" (text box)

Figure 10. Mock-up of the Activity Detail View

Because the Opportunity field is predefined on the Action business component that these applets reference, you are only required to add the field to each applet. It is not necessary to add a field to the business component.

Each object that Siebel CRM displays in the Siebel client includes a display name or caption. Example objects include controls, fields, views, and screens. The following properties determine the text label that Siebel CRM displays in the Siebel client:

- The Display Name property for a list applet
- The Caption property for a form applet

To determine the text that the property displays, you use the drop-down list for the property to identify a predefined symbolic string. You do not define text directly in the property.

For more information, see *Configuring Siebel Business Applications*.

Adding a Field to a List Applet

You use the Web Layout Editor to add a field to a list applet. For more information, see [“Opening the Web Layout Editor” on page 180](#).

To add a field to a list applet

- 1 In the Object Explorer, click Applet.
- 2 In the Applets list, locate the Activity List Applet (SCW) applet.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 Right-click the Activity List Applet (SCW) applet, and then choose the Edit Web Layout menu item.
- 6 In the Controls/Columns window, set the Mode to Edit List.
- 7 To make room for the Opportunity field, rearrange predefined controls in the Web template.
To determine the required layout, see [Figure 10 on page 71](#). To move a control, drag and drop it.
- 8 Drag a Field control from the Palettes window, and drop it in the space you made in [Step 7](#).
- 9 In the Properties window, set properties for the control you added in [Step 8](#), using values from the following table.

Property	Description
Name	Enter Opportunity.
Display Name	<p>Do the following:</p> <ul style="list-style-type: none"> ■ Click the arrow in the Display Name property. ■ In the Display Name - String Reference dialog box, type Opportunity in the <i>Starting with</i> window, and then click Go. ■ In the Current String Value column, click the record that includes only Opportunity. It is approximately the fourth record in the list. Do not choose the record that includes a colon, such as <i>Opportunity:</i>. ■ Click Pick.
Field	<p>In the Field property, click the arrow, and then choose Opportunity.</p> <p>Note that the values that Siebel Tools includes in the list of values in the Field property are fields that are defined on the parent business component of the Applet. In this example, this is the Action business component.</p>
Runtime	<p>Set to TRUE.</p> <p>If the Runtime property is TRUE, then Siebel CRM performs a run-time check to determine if it must display a picklist, calculator, calendar, or multi-value group (MVG).</p>

- 10 To approximate how Siebel CRM renders the applet in the Siebel client, right-click anywhere in the Web template, and then choose the Preview menu item.

You can also export the preview to an HTML file. To do this, choose the File menu, and then the Export menu item. Choose a file name and location in the Save As dialog.

- 11 Make adjustments as necessary.
- 12 Choose the File menu, and then the Save menu item.

Adding a Field to a Form Applet

You use the Web Layout Editor to add a field to a form applet. For more information, see [“Opening the Web Layout Editor” on page 180](#).

To add a field to a form applet

- 1 In the Object Explorer, click Applet.
- 2 In the Applets list, locate the Activity Detail Applet (SCW) applet.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 Right-click the Activity Detail Applet (SCW), and then choose the Edit Web Layout menu item.
- 6 To make room for the Opportunity field, rearrange controls in the Web template that are already mapped to the Web template.
To determine the required layout, see [Figure 10 on page 71](#). To move a control, drag and drop it.
- 7 Drag a Field control from the Palettes window, and drop it in the space you made in [Step 6](#).
- 8 In the Properties window, set properties for the control you added in [Step 7](#). Use the values described in [Step 9 on page 72](#), with the following differences:
 - Set the Caption property instead of the Display Name property.
 - Set the HTML Type property to Field.
 - Do not set the Runtime property.
- 9 Drag a Label control from the Palettes window, and drop it immediately to the left of the Opportunity control you added in [Step 7](#).
- 10 In the Properties window, set properties for the control you added in [Step 9](#). Use the same values you used for the Opportunity field you defined in [Step 8](#).
- 11 To approximate how Siebel CRM renders the applet in the Siebel client, right-click anywhere in the Web template, and then choose the Preview menu item.
- 12 Choose the File menu, and then the Save menu item.

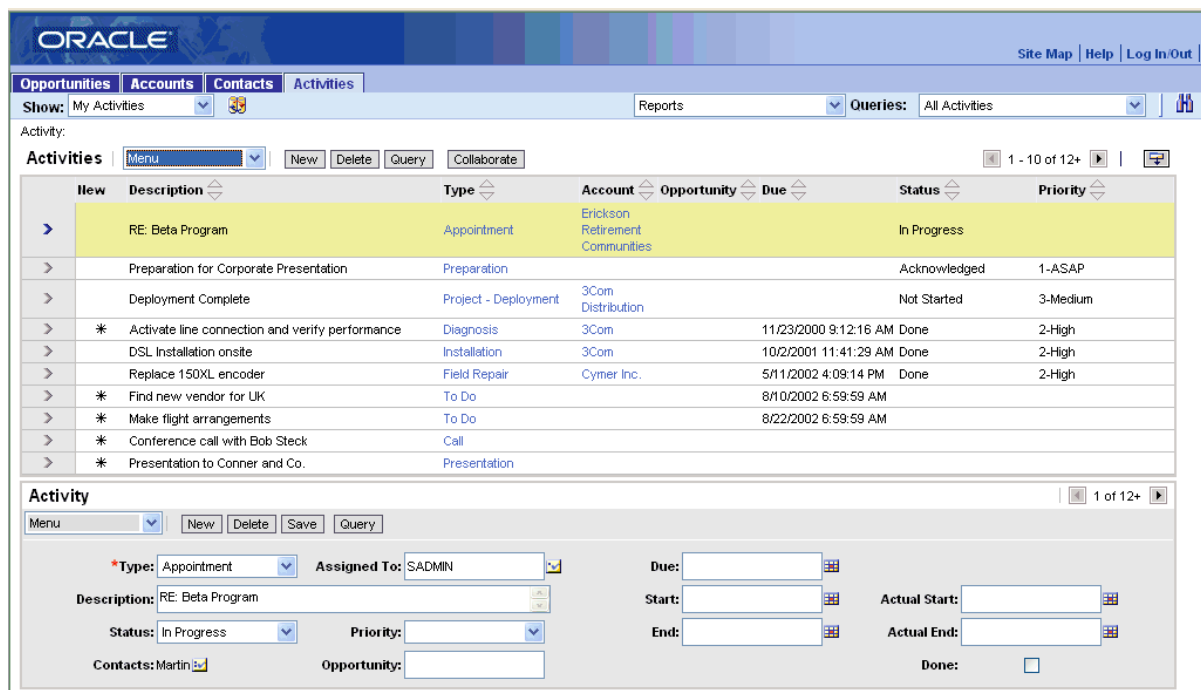
Verifying Your Configuration

This task is a step in “Process of Configuring the User Interface Layer” on page 65.

After you finish configuring screens, views, and applets, you must verify your changes.

To verify your configuration

- 1 In Siebel Tools, compile your changes.
For more information, see “Compiling Projects” on page 184.
- 2 Open the Siebel client, and then verify that the Activities screen is similar to the following screen capture. It must include the following:
 - An Opportunity column between the Account and Due columns in the list applet.
 - An Opportunity field under the Priority field in the form applet.



6

Configuring the House and Opportunity Entities

This chapter describes how to configure the house and opportunity entities. It includes the following topics:

- [Process of Configuring the House Detail View on page 75](#)
- [Process of Configuring the House Detail - Appraisals View on page 90](#)
- [Process of Configuring the Opportunity Details View on page 99](#)

Process of Configuring the House Detail View

This process is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

To configure the House Detail View, perform the following tasks:

- 1 [“Examining Business Requirements for the House Detail View” on page 75](#)
- 2 [“Adding Columns to the Products Table” on page 77](#)
- 3 [“Configuring the Internal Product Business Component” on page 79](#)
- 4 [“Configuring Text for Labels and Other User Interface Elements” on page 80](#)
- 5 [“Modifying the Product List Applet” on page 83](#)
- 6 [“Modifying the Product Form Applet” on page 85](#)
- 7 [“Creating the House Detail View” on page 86](#)
- 8 [“Creating the Houses Screen” on page 87](#)
- 9 [“Compiling, Registering, and Testing the House Detail View” on page 90](#)

Examining Business Requirements for the House Detail View

This task is a step in [“Process of Configuring the House Detail View” on page 75](#).

The NREC House Detail view must display the current list of houses for sale and the details for each house, such as square feet and number of bedrooms. The product entity in the Siebel data model meets some of the NREC requirements. However, some information, such as square feet and number of bedrooms, are not part of the Siebel data model.

You can extend the predefined data model to meet unique business requirements. For example, assume NREC decides to add columns to the internal products base table to store the following information for each house:

- Address
- City
- Number of bathrooms
- Number of bedrooms
- Price
- Square feet
- State
- ZIP code

Although List Price exists as a predefined field in the Internal Product business component, assume NREC decides to add a column to the S_PROD_INT table to keep the house attributes stored in one table.

For more information about extending the data model, see *Configuring Siebel Business Applications*.

Figure 11 illustrates a mock-up of the House Detail View. This is a new view that uses predefined applets.

House Detail View

Houses

Address	City	State	Zip Code	Bedrooms	Bathrooms	SQFT	List Price

More Info

Address <input style="width: 80%;" type="text"/>	Bedrooms <input style="width: 80%;" type="text"/>	Stories <input style="width: 80%;" type="text"/>
City <input style="width: 80%;" type="text"/>	Bathrooms <input style="width: 80%;" type="text"/>	Region <input style="width: 80%;" type="text"/>
State <input style="width: 80%;" type="text"/>	Square Feet <input style="width: 80%;" type="text"/>	List Price <input style="width: 80%;" type="text"/>
Zip Code <input style="width: 80%;" type="text"/>	Neighborhood <input style="width: 80%;" type="text"/>	Year Built <input style="width: 80%;" type="text"/>

Figure 11. Mock-Up of the House Detail View

Adding Columns to the Products Table

This task is a step in [“Process of Configuring the House Detail View”](#) on page 75.

In this topic, you add new columns to the base table and then apply these changes to your local database.

To add columns to the Products table

- 1 In the Object Explorer, click Table.
- 2 In the Tables list, locate the S_PROD_INT table.

- 3 Choose the Tools menu, and then the Lock Project menu item.

Note the following:

- The Project property is read-only for tables. You cannot change the value in the project property.
- The Type property for the S_PROD_INT table is Data (Public). You can only extend a public table.

- 4 Add columns to the table:

- a In the Object Explorer, expand the Table tree, and then click Column.
- b In the Columns list, add a new column for each row listed in the following table.

Name	Physical Type	Length
X_ADDRESS	Varchar	30
X_BATHROOMS	Varchar	30
X_BEDROOMS	Varchar	30
X_CITY	Varchar	30
X_PRICE	Varchar	30
X_STATE	Varchar	30
X_SQFT	Varchar	30
X_ZIP_CODE	Varchar	30

The name of an extension column begins with the X_ prefix. For example, X_ADDRESS. The User Name of an extension column ends with the Ext suffix. For example, X_ADDRESS Ext. Siebel Tools automatically applies these formats.

- 5 Apply schema changes to the local database:

- a In the Tables list, Click Apply/DLL.
- b In the Choose option dialog box, accept the Apply default, and then click OK.
- c In the Warning dialog box, click OK.
- d In the Apply Schema dialog box, in the Tables window, choose Current Row.

This setting updates the database to reflect the schema change to the row that is currently chosen in the Tables list. In this situation, that row is for the S_PROD_INT table only.

- e Do not specify a table space or index space.
- f Verify that the information in the ODBC Data Source text box is correct for your environment.

You cannot apply a schema change to any database other than the one to which you are currently connected. If you are connected to the Sample database, then do the following:

- Enter SI EBEL in the Database User window.
- Enter SI EBEL in the Database User Password window.

- Enter SEAW Samp Db default instance in the ODBC Data Source window.

For more information, see ["Prototyping with the Sample Database" on page 63](#).

- g Click Apply.

Siebel Tools applies the schema changes. The new columns you added now physically exist in your local database. They are available to use in your configuration.

Applying Schema Changes to the Server Database

Typically, after you test changes in the local database environment, you apply these changes to the server database. Checking in a project copies configuration changes to the Siebel Server, but it does not apply changes you make to the physical database. You must perform this work separately. For more information, see *Configuring Siebel Business Applications*.

Configuring the Internal Product Business Component

This task is a step in ["Process of Configuring the House Detail View" on page 75](#).

In this topic, you modify properties of the business component to allow the user to update records, and you add custom business component fields.

To configure the Internal Products business component

- 1 In the Object Explorer, click Business Component.
- 2 In the Business Components list, locate the Internal Product business component.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 To allow the user to update records, change business component properties using values from the following table.

Property	Value
No Delete	FALSE
No Insert	FALSE
No Merge	FALSE
No Update	FALSE

- 6 Add fields to the business component:
 - a In the Object Explorer, expand the Business Component tree, and then click Field.

- b In the Fields list, add a record for each of the rows in the following table.

Name	Column
Address	X_ADDRESS
Bathrooms	X_BATHROOMS
Bedrooms	X_BEDROOMS
City	X_CITY
Price	X_PRICE
Square Feet	X_SQFT
State	X_STATE
Zip Code	X_ZIP_CODE

The column property references the columns you created in [Step 4 on page 78](#).

Configuring Text for Labels and Other User Interface Elements

This task is a step in [“Process of Configuring the House Detail View” on page 75](#).

In this topic you configure text for labels and other user interface elements.

To configure text for labels and other user interface elements

- 1 Make sure Siebel Tools is set up to configure symbolic strings.
For more information, see [“Setting Up Siebel Tools” on page 60](#).
- 2 Check out the NREC Configuration project.
The Symbolic Strings project includes predefined symbolic strings. It is recommended that you create all custom symbolic strings in a custom project. For example, the NREC Configuration project.
- 3 In the Object Explorer, click Symbolic Strings.

- 4 In the Symbolic Strings list, create new record using values from the following table.

Name	Current String Value
X_Bathrooms	Bathrooms
X_Bedrooms	Bedrooms
X_House_ID	House ID
X_Number_of_Bathrooms	Number of Bathrooms
X_Number_of_Bedrooms	Number of Bedrooms
X_Square_Feet	Square Feet

For the Project property, use NREC Configuration for all symbolic strings you create. Some of the customizations you make for the NREC project involve adding new fields or changing the names of predefined fields. To make these changes, you add these new symbolic strings or edit the Symbolic String Locale of a predefined string.

As you create each symbolic string Siebel Tools automatically adds a child Symbolic String Locale object. This object includes ENU as the language, as determined by the language mode in Siebel Tools, which is set by default to ENU.

About Symbolic Strings

Siebel CRM stores the text that it displays on screens and views in the Siebel client. It stores this information in the repository. This includes the names of every screen, view, and field. For example, if the user displays information on accounts, then the view includes fields with names such as Account Name, Account Team, Site, Status, Territory, and so forth. Siebel CRM stores these names in the SRF file. If the user displays a page, then the Siebel Server uses the SRF file to determine which fields to display and the text labels to display with each field. These text labels are strings of text.

Symbolic strings is a model that Siebel CRM uses to keep the text strings separate from the view definitions. Symbolic strings provide the following advantages:

- Reduces redundancy because many objects can reference one symbolic string. For example, you only need to define the term Account Team one time as a symbolic string. When you must use Account Team on multiple views, you can reference the same string rather than defining a separate text label for every view and applet that uses Account Team.
- Results in a more consistent display of objects in the Siebel client. By reusing the same symbolic string, you avoid having slightly different phrasing on similar views. For example Acct. Team, Act. Team, Account Group, Client Team, and so forth.
- Simplifies maintenance because you only have to maintain one string for a given term. If you must change the name from Account Team to Client Group, you can update the symbolic string. Siebel CRM displays your update in all views that use this symbolic string. This technique is more efficient than having to change each view individually.

- Eliminates duplicated translations of the same word, which simplifies translation. Because you reuse symbolic strings, the same term does not exist in multiple different objects that must be translated.
- Reduces localization cost, time, and inconsistency. If the term Account Team is on five views, then Siebel CRM must translate it five times. A symbolic string allows Siebel CRM to translate the term only once. This saves on localization costs and reduces the development time for the project. The use of symbolic strings also helps prevent inconsistent translations. Although the NREC example does not cover localization, many custom deployments are localized.

How Siebel CRM Uses Symbolic Strings

Siebel CRM uses symbolic strings in a top-level object in the Siebel repository named Symbolic Strings and a child object named Symbolic String Locale. Each symbolic string record represents a word or phrase, for example Account or Contact, and is language-independent. Siebel CRM stores all translations of that word or phrase, including English, as child symbolic string locale records. User interface objects reference symbolic string records for text strings. Siebel CRM retrieves the literal display value from one of the several translations stored as symbolic string locale records according to the current Tools language mode.

Strings Not Included in the Symbolic Strings Model

The symbolic strings model includes text strings stored in the repository. User interface objects reference these strings. Example interface objects include applet titles, control captions, and list column display names. The symbolic string model does not include other types of strings that are typically supplied as seed data, such as LOVs, error messages, or predefined queries.

Properties of a Symbolic String

Table 9 describes some of the properties of a symbolic string.

Table 9. Properties of a Symbolic String

Name	The unique name for the symbolic string. Siebel Tools enforces a prefix for the symbolic string name, such as X_. This helps you distinguish a custom symbolic string from a predefined Siebel System string, which is prefixed with SBL_. The SymStrPrefix parameter in the tools.cfg file determines the value that Siebel Tools uses for the prefix. For more information, see "Setting Up Siebel Tools" on page 60 .
Current String Value	The text that Siebel CRM displays in the Siebel client.
Definition	A description of the symbolic string.

Modifying the Product List Applet

This task is a step in [“Process of Configuring the House Detail View”](#) on page 75.

Assume that NREC determines that it is preferable to modify the following two applets rather than creating new applets:

- Product List Applet
- Product Form Applet

These applets reference the Internal Product business component and meet most of the NREC requirements.

To modify the Product List Applet

- 1 In the Object Explorer, click Applet.
- 2 In the Applets list, locate the Product List Applet.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 In the Object Explorer, expand the Applet tree, expand the List tree, and then click List Column.
- 6 Change the display name for predefined list columns using values from the following table.

Name of Predefined List Column	New Value
Part #	House ID
Name	Summary

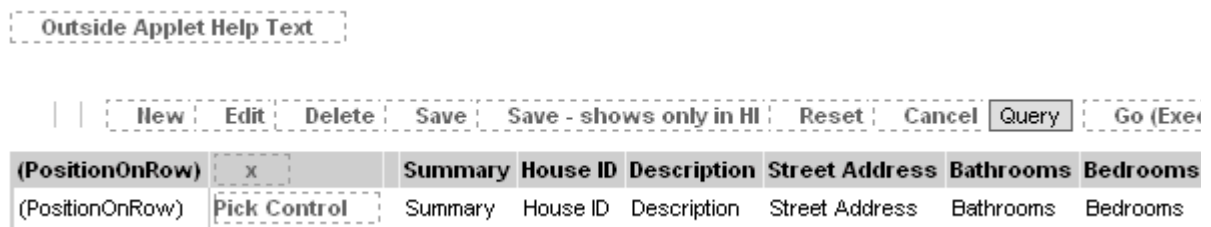
- 7 In the Object Explorer, click Applet.
- 8 In the Applets list, right-click the Product List Applet, and then choose the Edit Web Layout menu item.
- 9 In the Web Layout Editor, in the Controls/Columns window, make sure the mode is set to Base.
- 10 In the Web template, delete the following columns:
 - Class
 - Compensable
 - Customizable
 - Description
 - Effective End Date
 - Effective Start Date
 - Equivalent Product
 - Lead Time
 - Orderable

- Organization
- Product Line
- Revision
- Serialized
- Service Product
- Unit of Measure

11 For each row in the following table, drag a Field control from the Palettes window, and then drop it in the Web template. Each row represents a separate control. Set properties for each control using values from the following table.

Name	Caption	Field	Show In List
Address	Street Address	Address	FALSE
Bathrooms	Bathrooms	Bathrooms	TRUE
Bedrooms	Bedrooms	Bedrooms	TRUE
City	City	City	FALSE
Price	Price	Price	TRUE
Square Feet	Square Feet	Square Feet	TRUE
State	State	State	FALSE
ZIP Code	ZIP Code	ZIP Code	FALSE

The Web template must resemble the following screen capture:



For more information, see [“Adding a Field to a List Applet” on page 72](#).

- 12** Choose the File menu, and then the Save menu item.
- 13** In the Controls/Columns window, set the mode to Edit.
- 14** Repeat [Step 10](#) through [Step 12](#).
- 15** In the Controls/Columns window, set the mode to Query.
- 16** In the Web template, delete the following columns:
 - Class

- Vendor

17 Save your work, and then close the Web Layout Editor.

About the Show In List Property

The Show In List property determines if Siebel CRM displays the field by default in the list applet. If the Show In List property is FALSE, then Siebel Tools displays the field with gray color in the Web template. Siebel CRM hides these fields by default in the Siebel client. To display them, the user can choose Columns Displayed from the applet menu in the Siebel client.

Modifying the Product Form Applet

This task is a step in “[Process of Configuring the House Detail View](#)” on page 75.

In this topic, you modify the Product Form Applet.

To modify the Product Form Applet

- 1 In the Object Explorer, click Applet.
- 2 In the Applets list, locate the Product Form Applet.
- 3 Choose the Tools menu, and then the Lock Project menu item.
- 4 Change the value in the Project property to NREC Configuration.
- 5 In the Object Explorer, expand the Applets tree, and then click Control.
- 6 Modify the Caption property of the following predefined controls:
 - Change the Caption property of the PartNum control to House ID.
 - Change the Caption property of the Name control to Summary.
- 7 In the Applets list, right-click the Product Form Applet, and then choose the Edit Web Layout menu item.
- 8 In the Web Template, delete the controls that NREC does not require.
Use [Figure 11 on page 77](#) to determine the controls to delete.

- 9 For each row in the following table, drag a Field control from the Palettes window, and then drop it in the Web template. Each row represents a separate control. Set properties for each control using values from the following table.

Name	Field	Caption
Address	Address	Street Address
Bathrooms	Bathrooms	Bathrooms
Bedrooms	Bedrooms	Bedrooms
City	City	City
Price	Price	Price
Square Feet	Square Feet	Square Feet
State	State	State
ZIP Code	ZIP Code	ZIP Code

For more information, see [“Adding a Field to a Form Applet” on page 73](#).

- 10 Drag the EditRecord control from the Controls/Columns window and drop it in an appropriate placeholder in the Web template.
- 11 Save your work, and then close the Web Layout Editor.

Creating the House Detail View

This task is a step in [“Process of Configuring the House Detail View” on page 75](#).

You must create a new view to display the Product List and Product Form Applets you modified.

To create the House Detail View

- 1 Choose the File menu, and then the New Object menu item.
- 2 In the General tab of the New Object Wizards dialog box, click View, and then click OK.

- In the New View wizard, provide the information that the wizard requests using the information from the following table.

Property	Value
Project	NREC Configuration
Name	NREC House Detail View
Title	House Detail View
Business Object	Internal Product
Upgrade Behavior	Admin
Web Template	View Detail (Parent with Pointer)
Selected Applets	Choose the following applets: <ul style="list-style-type: none"> ■ Product List Applet ■ Product Form Applet

After you click Finish in the last dialog box, Siebel Tools creates the view and related objects.

- In the Object Explorer, click View.
- In the Views list, locate the NREC House Detail View.
- Right-click the NREC House Detail View, and then choose the Edit Web Layout menu item.
- Right-click anywhere in the Web template, choose Preview, and then examine the layout of the new view.
- Choose the File menu, and then the Close menu item.

NOTE: If you create a view that contains special visibility properties, such as Organization, Manager, or Sales Rep, then set the Visibility Applet Type property of the View object. Do not modify the View Web Template Item object. The only exception to this rule is the Home Page. To set visibility for the Home Page, modify the View Web Template Item object.

Creating the Houses Screen

This task is a step in [“Process of Configuring the House Detail View”](#) on page 75.

After you create a view you typically add it to a screen so Siebel CRM can display it in the Siebel client. For the NREC example, you must create a new screen for houses. This screen groups the views related to the Houses entity.

To create the Houses screen

- 1 Create a symbolic string for the following text:

Houses

For more information, see [“Configuring Text for Labels and Other User Interface Elements” on page 80.](#)

- 2 In the Object Explorer, click Screen.
- 3 In the Screens list, right-click, and then choose the New Record menu item.
- 4 In the Screens list, define properties for the new record using values from the following table.

Property	Value
Name	NREC House Screen
Project	NREC Configuration
Viewbar Text	Houses

- 5 In the Object Explorer, expand the Screen tree, and then click Screen View.
- 6 In the Screen Views list, right-click, and then choose the New Record menu item.
- 7 In the Screen Views list, define properties for the new record using values from the following table.

Property	Value	Description
Type	Aggregate Category	Defines a category that Siebel CRM displays in the Site Map under the Houses link. You group this and other screen views in this category.
Category Name	NREC	Provides a name for the category. When you define a screen view, you choose this category name as the Parent Category for the screen view.
Viewbar Text	More Info	Text that Siebel CRM displays in the Show menu.

- 8 In the Screen Views list, right-click, and then choose the New Record menu item.
- 9 In the Screen Views list, define properties for the new record using values from the following table.

Property	Value	Description
View	NREC House Detail View	Choose the name of associated view.
Parent Category	NREC	Choose the NREC category to place this screen view in the category.

Property	Value	Description
Viewbar Text	More Info	Enter the text. Siebel CRM displays this text in the Show list in the client.
Sequence	20	Enter the number. It determines the sequence in which Siebel CRM displays tabs.

- 10 In the Screen Views list, locate the NREC category screen view you created in [Step 7](#).
- 11 Set the Category Default View property to NREC House Detail View.
- 12 Make the NREC House Detail View the default view for the House screen:
 - a In the Object Editor, click Screen.
 - b In the Screens list, locate the NREC House Screen.
 - c Set the Default View property to NREC House Detail View.
- 13 Define the page tab and screen menu items:
 - a In the Object Explorer, click Application.
 - b In the Applications list, locate the Siebel eChannel application.
 - c In the Object Explorer, expand the Application tree, and then click Page Tab.
 - d In the Page Tabs list, right-click, choose New Record, and then create a new record using values from the following table.

Property	Value	Description
Screen	NREC House Screen	The screen object that is associated with the page tab.
Sequence	125	Determines the sequence in which Siebel CRM displays the page tabs.
Text	Houses	Text that Siebel CRM displays in the page tab.

Siebel CRM displays page tabs across the top of the client. It displays Screen Menu Items in the Site Map.

- e In the Object Explorer, in the Application tree, click Screen Menu Item.
- f In the Screen Menu Items list, add a new record using values from the following table.

Property	Value	Description
Screen	NREC House Screen	Screen associated with the Screen Menu Item.
Sequence	15	Determines the sequence in which Siebel CRM displays the item.
Text	Houses	Text that Siebel CRM displays as a link on the Site Map.

Compiling, Registering, and Testing the House Detail View

This task is a step in [“Process of Configuring the House Detail View”](#) on page 75.

In this topic, you compile, register, and test the House Detail View.

To compile, register, and test the House Detail View

- 1 Compile your changes.
For more information, see [“Compiling Projects”](#) on page 184.
- 2 Register the House Detail View.
For more information, see [“Registering a New View”](#) on page 186.
- 3 Verify that the House Detail View meets NREC requirements.
The view must resemble [Figure 11](#) on page 77. For more information, see [“Verifying Your Changes”](#) on page 187.

Process of Configuring the House Detail - Appraisals View

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To configure the House Detail - Appraisals View, perform the following tasks:

- 1 [“Examining Business Requirements for the House Detail - Appraisals View”](#) on page 90
- 2 [“Creating the Appraisals Business Component”](#) on page 92
- 3 [“Creating a Link Between Houses and Appraisals”](#) on page 94
- 4 [“Associating the Appraisals Business Component with a Business Object”](#) on page 95
- 5 [“Creating the Appraisals List Applet”](#) on page 96
- 6 [“Creating the House Detail - Appraisals View”](#) on page 97
- 7 [“Adding the House Detail - Appraisals View to the Houses Screen”](#) on page 97
- 8 [“Compiling, Registering, and Testing the House Detail - Appraisals View”](#) on page 98

Examining Business Requirements for the House Detail - Appraisals View

This task is a step in [“Process of Configuring the House Detail - Appraisals View”](#) on page 90.

The House Appraisals view provides a list of appraisal details for a particular house. These details include the appraisal date, appraisal value, and the appraiser. A one-to-many relationship exists between products and appraisals. The products represent houses. A user can associate each house with many appraisals. For more information, see [“Design Strategy for the Data Layer” on page 29](#).

Because the predefined Siebel data model does not include an appraisal entity, NREC must create a new business component for appraisals that stores data in a predefined one-to-many extension table. In the NREC example, you use the S_PROD_INT_XM table, which is a one-to-many extension table of the S_PROD_INT table. For more information about extension tables, see *Configuring Siebel Business Applications*.

NREC determines that it must use a master-detail view to display appraisal data. The view includes a house record in the top applet and a list of appraisals in the bottom applet. For the top applet, you can use the Product Form Applet that you configured in [“Modifying the Product List Applet” on page 83](#). To display the list of appraisals in the bottom applet, you must create a new applet. To view the end result of this configuration work, see [Figure 12 on page 92](#).

About Predefined One-to-Many Extension Tables

A *one-to-many extension table* is a predefined table that includes a one-to-many relationship with a base table. This extension table includes generic columns that you can use to store data. You can extend the data model for your purposes and track entities that are not part of the predefined Siebel data model. Because extension tables are part of the predefined data model, you do not modify the database schema. There are more than 20 one-to-many extension tables in the predefined data model. The name of a one-to-many extension table contains the _XM suffix.

If you use a one-to-many extension table to store data, then you use the TYPE column to group records. You create a new business component for the entity. This business component must include the following items:

- A Type field that defaults to a unique value
- A search specification that finds only those records that contain this unique value

If you use this technique, then the following items apply:

- You can use a single one-to-many extension table to store data for multiple business components.
- If only one business component uses the extension table, then you must use this technique.

For more information, see [“Creating the Appraisals Business Component” on page 92](#).

Screen Capture of the House Detail - Appraisals View

Figure 12 includes a screen capture of the House Detail - Appraisals View you will configure.

The screenshot displays the Oracle Siebel interface for the 'House Detail - Appraisals View'. At the top, there are navigation tabs for 'Opportunities', 'Accounts', 'Contacts', 'Activities', and 'Houses'. Below these, there are utility buttons like 'New', 'Edit', 'Delete', 'Save', 'Query', and 'Create Auction'. The main form area contains several input fields: 'House ID' (D4206), 'Bedrooms' (3), 'Street Address' (23815 Lansdowne Circle), 'Price' (380,000), 'Bathrooms' (2), 'City' (Ashburn), 'Square Feet' (1,600), 'State' (VA), and 'ZIP Code' (20146). A 'Summary' field contains the text '2-level Townhome, 2 marked'. At the bottom of the form, there is a table with the following data:

Appraisal Value	Date	Name
392,000	May 12, 2004	Pat Cmeyla

Figure 12. Screen Capture of the House Detail - Appraisals View

Creating the Appraisals Business Component

This task is a step in “Process of Configuring the House Detail - Appraisals View” on page 90.

To display data from a one-to-many extension table, you must define a new business component that includes the following items:

- Fields that reference the generic columns in the extension table. Examples of generic columns include ATTRIB_01, ATTRIB_02, and so forth.
- The following columns that establish the user key:
 - **PAR_ROW_ID**. Referenced by the foreign key field in the one-to-many link.
 - **NAME**. Referenced by the field tracked in the business component. The value in the name must make the record unique for each parent record.
 - **TYPE**. Referenced by the Type field. It groups records in the extension table. You set a default value for the Type field and then configure the business component to search for those records in the extension table that contains this value.

To satisfy the U1 index of the _XM table, the combination of the NAME, PAR_ROW_ID, and TYPE columns must be unique.

To create the Appraisals business component

- 1 In Siebel Tools, choose the File menu, and then the New Object menu item.
- 2 In the General tab of the New Object Wizards dialog box, click BusComp, and then click OK.

- 3 In the New Business Component dialog box, enter the information from the following table, and then click Next.

Property	Value
Project	Product (SSE)
Name	Appraisals
Table	S_PROD_INT_XM

- 4 In the Single Value Fields dialog box, enter a column and field name for each row from the following table.

Column in Base Table	Field Name
ATTRIB_01	Date
ATTRIB_02	Appraisal Value
ATTRIB_03	Comments
PAR_ROW_ID	Par Row Id
NAME	Name
TYPE	Type

After you enter a column and field, click Add, and then add the next column and field. Note the following:

- The Date field stores the date on which the appraiser performed the appraisal.
 - The Appraised Value field stores the value of the property.
 - The Name field stores the name of the appraiser.
 - The Type field groups records.
- 5 Click Finish.
- The Business Component Wizard creates the business component according to the information you entered.
- 6 In the Business Components list, modify the following property for the Appraisals business component.

Property	Value
Search Specification	[Type]= 'Appraisals'

This search specification instructs the business component to retrieve only those records in which the Type field contains the value Appraisals.

- 7 Set the Predefault Value of the Type field:

- a In the Object Explorer, expand the Business Component tree, and then click Field.
- b In the Fields list, locate the Type field and then set the following property.

Property	Value
Predefault Value	Appraisals

This property instructs Siebel CRM to always default the Type field in the Siebel client to Appraisals.

- 8 In the Fields list, make sure the Required property contains a check mark for each of the following fields:
 - Name
 - Par Row Id
 - Type

Because the combination of the NAME, PAR_ROW_ID, and TYPE columns satisfy the U1 index of the _XM table, you must make sure these fields are required.

Creating a Link Between Houses and Appraisals

This task is a step in [“Process of Configuring the House Detail - Appraisals View”](#) on page 90.

The relationship between Houses and Appraisals is one-to-many. For each house, there can be many appraisals. A link establishes this relationship between the Internal Products parent business component and the Appraisals child business component.

To create a link between houses and appraisals

- 1 Display the Row_Id field in the Internal Product business component:
 - a In the Object Explorer, click Business Component.
 - b In the Business Components list, locate the Internal Product business component.
 - c In the Object Explorer, expand the Business Component tree, and then click Field.
 - d In the Fields list, right-click, choose New Record, and then set the following properties.

Property	Value
Name	Row_Id
Column	ROW_ID

Before you can create the link, you must display the Row_Id field in the Internal Products business component. Siebel CRM uses this field to link with the Par Row Id field in the Appraisals business component.

- 2 Create a link:
 - a In the Object Explorer, click Link.
 - b In the Links list, right-click, and then choose the New Record menu item.
 - c Set the following properties.

Property	Value
Name	Internal Product/Appraisals
Project	NREC Configuration
Parent Business Component	Internal Product
Child Business Component	Appraisals
Source Field	Row_Id
Destination Field	Par Row ID

The Source Field establishes the primary key value in the parent table. The Destination Field establishes the foreign key value in the child table.

Associating the Appraisals Business Component with a Business Object

This task is a step in ["Process of Configuring the House Detail - Appraisals View"](#) on page 90.

A business object groups related business components together, and gathers the link objects that associate two business components with one another. The Internal Products business object provides data for the Houses screen. You must associate the Appraisals business component that you create in ["Creating the Appraisals Business Component"](#) on page 92 to the Internal Products business object.

To associate the Appraisals business component with a business object

- 1 In the Object Explorer, click Business Object.
- 2 In the Business Objects list, locate the Internal Product business object.
- 3 In the Object Explorer, expand the Business Object tree, and then click Business Object Component.
- 4 In the Business Object Components list, enter a new record using values from the following table.

Property	Value	Description
Bus Comp	Appraisals	This is the business component you created in "Creating the Appraisals Business Component" on page 92.
Link	Internal Product/ Appraisals	Defines the master-detail relationship between the parent and child business components.

Creating the Appraisals List Applet

This task is a step in “[Process of Configuring the House Detail - Appraisals View](#)” on page 90.

To display appraisal data in the Siebel client, you create a new list applet that displays a list of appraisal records for a single house record.

To create the Appraisal list applet

- 1 Choose the File menu, and then the New Object menu item.
- 2 In the Applets tab, click List Applet and then click OK.
- 3 In the General dialog box, enter values from the following table, and then click Next.

Property	Value
Project	NREC Configuration
Applet Name	Appraisals List Applet
Display Title	Appraisals
Business Component	Appraisals
Upgrade Behavior	Admin

- 4 In the Web Layout - General dialog box, choose the templates to use, using values from the following table, and then click Next.

Template	Value
Template for Base read-only mode	Applet List (Base/EditList)
Template for Edit List mode	Applet List (Base/EditList)
Template for Edit mode	Applet List Edit (Edit/New/Query)

- 5 In the Web Layout - Fields dialog box, choose the following fields, and then click Next:

- Appraisal Value
- Date
- Name

- 6 In the next Web Layout - Fields dialog box, accept the default values and then click Next.
- 7 In the Finish dialog box, review the information you entered and then click Finish.

The New Applet Wizard creates a new list applet according to the information you entered and displays the applet in Web Layout Editor.

- 8 Choose the File menu, and then the Close menu item.

Creating the House Detail - Appraisals View

This task is a step in [“Process of Configuring the House Detail - Appraisals View”](#) on page 90.

In this topic you create a new view that displays the Product Form applet and the Appraisals List applet.

To create the House Detail - Appraisals View

- 1 Choose the File menu, and then the New Object menu item.
- 2 In the General tab of the New Object Wizards dialog box, click View and then click OK.
- 3 In the New View dialog box, enter information using values from the following table, and then click Next.

Property	Value
Project	NREC Configuration
View Name	House Detail - Appraisals View
View Title	Appraisals
Business Object	Internal Product

- 4 In the View Web Layout - Select Template dialog box, choose the View Detail (Parent with Pointer) Web template, and then click Next.
- 5 In the Web Layout - Applets dialog box, choose the following applets and then click Next:
 - Product Form Applet
 - Appraisals List Applet
- 6 In the Finish dialog box, review the information you entered and then click Finish.

The New View wizard creates the applet, the necessary supporting objects, and then opens the applet in the Web Layout Editor.
- 7 Choose the File menu, and then the Close menu item.

Adding the House Detail - Appraisals View to the Houses Screen

This task is a step in [“Process of Configuring the House Detail - Appraisals View”](#) on page 90.

In this topic, you add the House Detail - Appraisals View to the Houses Screen.

To add the House Detail - Appraisals View to the Houses Screen

- 1 Create a symbolic string that includes the following text:

Appraisals

For more information, see [“Configuring Text for Labels and Other User Interface Elements”](#) on page 80.

- 2 In the Object Explorer, click Screen.
- 3 In the Screens list, locate the NREC House Screen.
- 4 In the Object Explorer, expand the Screen tree, and then click Screen View.
- 5 In the Screen Views list, add a record using values from the following table.

Property	Value
View	House Detail - Appraisals View
Type	Detail View
Parent Category	NREC
Viewbar Text	Appraisals
Menu Text	Appraisals

Compiling, Registering, and Testing the House Detail - Appraisals View

This task is a step in [“Process of Configuring the House Detail - Appraisals View”](#) on page 90.

In this topic, you compile, register, and test the House Detail - Appraisals view.

To compile, register, and test the House Detail - Appraisals view

- 1 Compile your changes.

For more information, see [“Compiling Projects”](#) on page 184.

- 2 Register the House Detail - Appraisals View in the application.

For more information, see [“Registering a New View”](#) on page 186.

- 3 Verify that the new view meets NREC requirements.

The view must resemble [Figure 12](#) on page 92. For more information, see [“Verifying Your Changes”](#) on page 187.

Process of Configuring the Opportunity Details View

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To configure the Opportunity Details View, do the following:

- 1 [“Examining Business Requirements for the Opportunity Details View”](#) on page 99
- 2 [“Adding Columns to the Opportunity Table”](#) on page 100
- 3 [“Adding Fields to the Opportunity Business Component”](#) on page 101
- 4 [“Modifying the Opportunity List Applet”](#) on page 101
- 5 [“Modifying the Opportunity Form Applet”](#) on page 103
- 6 [“Compiling, Registering, and Testing the Opportunity Details View”](#) on page 105

Examining Business Requirements for the Opportunity Details View

This task is a step in [“Process of Configuring the Opportunity Details View”](#) on page 99.

To allow partner agents to enter and display information about opportunities, NREC uses the predefined Opportunity Details view in the Partner Portal application. Many of the attributes that NREC must track for each opportunity are not part of the predefined Siebel data model. For example, the data model does not include the number of bedrooms or bathrooms in which a potential buyer might be interested. This information must include the following items:

- Agent
- Number of Bathrooms
- Number of Bedrooms
- Price Range
- Square Feet

To add these items, you perform work that is similar to configuring the Houses entity, such as adding columns to the base table and then modifying the objects at the business logic layer and the user interface layer.

Figure 13 illustrates a mock-up of the Opportunity Detail View that NREC requires.

The mock-up is titled "Opportunity Detail View" and is divided into two main sections: "Opportunities" and "More Info".

The "Opportunities" section contains a table with the following columns: New, Name, Contact, Price Range, Agency, Agent, Bedrooms, and Bathrooms. There are four empty rows below the header.

The "More Info" section contains a form with the following fields:

- Name
- Bedrooms
- Mortgage Status
- Contact
- Bathrooms
- Contingent on Current Sale?
- Price Range
- Square Feet
- Reason for Buying
- Agency
- Neighborhood
- Stories
- Region

Figure 13. Mock-Up of the Opportunity Detail View

The Opportunity Detail view is the default view for the Opportunities screen. It includes the following applets:

- The Opportunity List Applet is the master applet.
- The Opportunity Form applet is the detail applet.

Adding Columns to the Opportunity Table

This task is a step in "Process of Configuring the Opportunity Details View" on page 99.

In this topic, you add columns to the opportunity table.

To add columns to the Opportunity table

- Perform the same work you perform to add columns to the products table, with the following differences:

Add columns to the S_OPTY table using values from the following table.

Name	Type	Length
X_AGENT	Varchar	50
X_BATHROOMS	Varchar	20
X_BEDROOMS	Varchar	20
X_PRICE_RANGE	Varchar	20
X_SQUARE_FEET	Varchar	20

For more information, see [“Adding Columns to the Products Table”](#) on page 77.

Adding Fields to the Opportunity Business Component

This task is a step in [“Process of Configuring the Opportunity Details View”](#) on page 99.

In this topic, you add fields to the Opportunity business component.

To add fields to the Opportunity business component

- Perform the same work you perform to add fields to the Internal Product business component, with the following differences:

Add fields to the Opportunity business component using values from the following table.

Name	Column
Agent	X_AGENT
Number of Bathrooms	X_BATHROOMS
Number of Bedrooms	X_BEDROOMS
Price Range	X_PRICE_RANGE
Square Feet	X_SQUARE_FEET

For more information, see [“Configuring the Internal Product Business Component”](#) on page 79.

Modifying the Opportunity List Applet

This task is a step in [“Process of Configuring the Opportunity Details View”](#) on page 99.

In this topic, you modify the predefined Opportunity List Applet (SCW). Siebel CRM displays it in the Opportunity Detail View. As an alternative, you could create new applets. However, modifying predefined objects is typically a preferable method because it requires less configuration work and causes fewer problems with future upgrades.

To modify the Opportunity List Applet

- 1 Create a symbolic string for the following text:

Price Range

For more information, see [“Configuring Text for Labels and Other User Interface Elements” on page 80](#).

- 2 In the Object Explorer, click Applet.
- 3 In the Applets list, right-click the Opportunity List Applet (SCW), and then choose Edit Web Layout.
- 4 In the Web template, delete the following list columns:

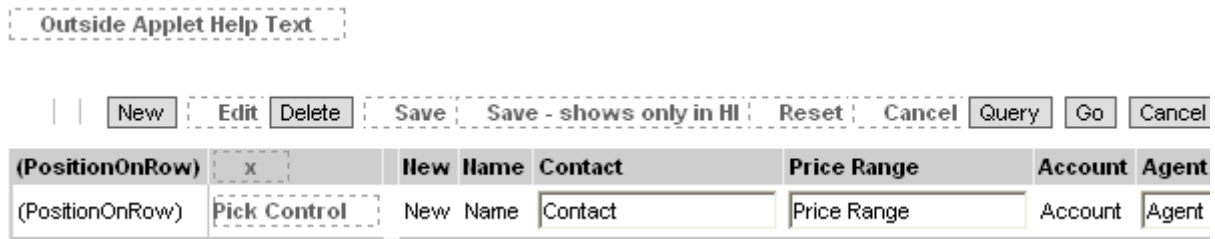
- Revenue
- Close Date
- Lead Quality
- Probability %
- Expected Value
- Status
- Reason
- Sales Method
- Created
- Description
- Site
- Address
- City
- State
- Zip Code
- Country
- Sales Team
- Organization
- Split Revenue

To choose multiple controls, press and hold down the CTRL, and then choose each field in succession. For more information, see [Figure 13 on page 100](#).

- For each row in the following table, drag a Field control from the Palettes window, and then drop it in the Web template. Each row represents a separate control. Set properties for each control using values from the following table.

Name	Field	Display Name	Show in List
Agent	Agent	Agent	TRUE
Contact	Key Contact Id	Contact	TRUE
Number of Bathrooms	Number of Bathrooms	Bathrooms	TRUE
Number of Bedrooms	Number of Bedrooms	Bedrooms	TRUE
Price Range	Price Range	Price Range	TRUE
Square Feet	Square Feet	SQFT	TRUE

The Web template must resemble the following screen capture:



For more information, see [“Adding a Field to a List Applet” on page 72](#) and [“About the Show In List Property” on page 85](#).

- In the Controls/Columns window, set the mode to Edit List.
- Repeat [Step 4](#) and [Step 5](#).
- Save your work, and then close the Web Layout Editor.

Modifying the Opportunity Form Applet

This task is a step in [“Process of Configuring the Opportunity Details View” on page 99](#).

To modify the Opportunity Form Applet

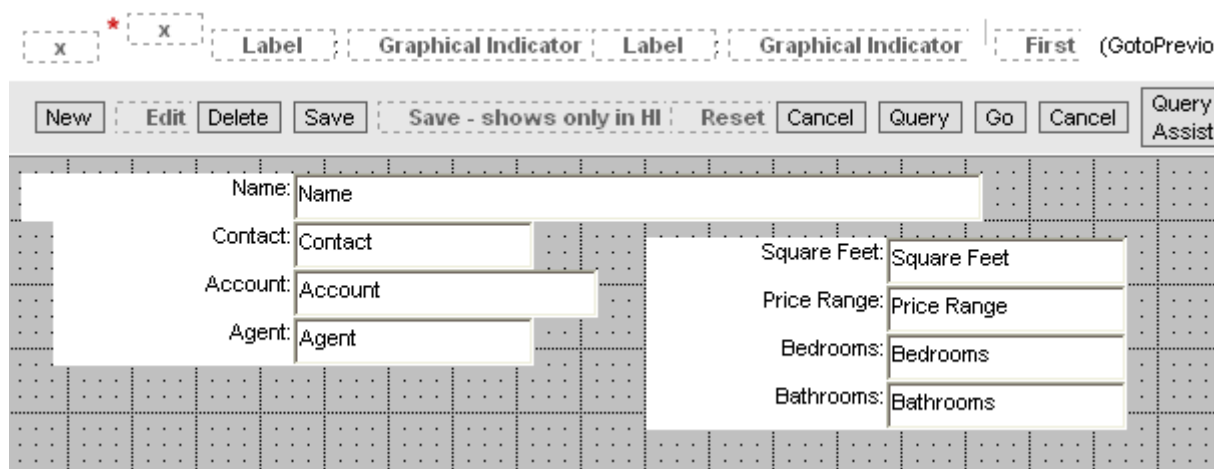
- In the Object Explorer, click Applet.
- In the Applets list, right-click the Opportunity Form Applet (SCW), and then choose the Edit Web Layout menu item.
- In the Web template, delete the following list columns:
 - Description

- Site
- Probability %
- Close
- Organization
- Revenue
- Expected Value
- Committed
- Sales Team

4 For each row in the following table, drag a Field control from the Palettes window, and then drop it in the Web template. Each row represents a separate control. Set properties for each control using values from the following table.

Name	Caption	Field
Agent	Agent	Agent
Contact	Contact	Key Contact Id
Number of Bathrooms	Bathrooms	Number of Bathrooms
Number of Bedrooms	Bedrooms	Number of Bedrooms
Price Range	Price Range	Price Range
Square Feet	SQFT	Square Feet

The Web template must resemble the following screen capture:



- 5 Repeat Step 3 and Step 5 for each active applet mode.
- 6 Save your work, and then close the Web Layout Editor.

Compiling, Registering, and Testing the Opportunity Details View

This task is a step in [“Process of Configuring the Opportunity Details View” on page 99](#).

In this topic, you compile, register, and test the Opportunity Details View.

To compile, register, and test the Opportunity Details View

- 1 Compile your changes.

For more information, see [“Compiling Projects” on page 184](#).

- 2 Register the Opportunity Details View.

For more information, see [“Registering a New View” on page 186](#).

- 3 Verify that the Opportunity Details View meets NREC requirements.

The view must resemble [Figure 13 on page 100](#). For more information, see [“Verifying Your Changes” on page 187](#).

7

Configuring Picklists, Drilldowns, and MVGs

This chapter describes how to configure picklists, drilldowns, and MVGs. It includes the following topics:

- [About Picklists and Drilldowns on page 107](#)
- [Process of Configuring Picklists, Drilldowns, and MVGs on page 109](#)

About Picklists and Drilldowns

This topic describes picklists and drilldowns.

About Picklists

To standardize data entry and minimize mistakes, NREC must use picklists. To enter information in a field, the user can choose a value from a picklist rather than typing the information in the field.

Static Picklist

A *static picklist* is a type of picklist that displays a single-column list of predefined values when the user clicks the arrow that is located to the right of a field. When the user chooses a value from this list, Siebel CRM enters the value in the field. A static picklist derives values from a table that a Siebel administrator maintains. To view an example of a static picklist, do the following:

- 1 In the Siebel client, navigate to the Opportunities screen.
- 2 Click the arrow in the Sales Stage field.

Siebel CRM displays the static picklist.

Dynamic Picklist

A *dynamic picklist* is a type of picklist that is similar to a static picklist, except it gets list values from a table that the user maintains. A field that uses a dynamic picklist is typically a joined field that displays data from a table other than the base table that the business component references. The opportunity field you display on the applet in [“Adding a Field to a List Applet” on page 72](#) is a joined field. A dynamic picklist on this field allows the user to use the opportunity name from the Opportunity table to update the activity. NREC requires a dynamic picklist on the Activity Detail Applet.

The Pick List Wizard creates the following dynamic picklist and related objects:

- **Pick List.** Defines the properties of the picklist, including the originating business component and the pick business component. The *originating business component* is the business component that the dynamic picklist references. The *pick business component* is the business component from which Siebel CRM picks values to display to the user. In the NREC example, the Action business component is the originating business component, and the Opportunity business component is the pick business component.
- **Pick Map.** Child object of a business component field that maps the source field in the pick business component with the target field in the originating business component.
- **Pick Applet.** Pop-up applet that displays the list of records.

Constrained Picklist

A *constrained picklist* is a type of picklist that uses a pick applet to display records that only include field values that match corresponding field values in the originating business component. For example, you can constrain the Opportunity pick applet to display only the opportunities for the account that are associated with the activity.

To constrain a picklist, you create another pick map with the Constrain property set to TRUE. This pick map defines fields on the originating business component and a field on the pick business component. If the values in these fields match, then Siebel CRM displays the record in the pick applet. You can create more than one constrain pick map. A constrain pick map does not copy the values of the field from the originating business component to the pick business component. It provides a filter that Siebel CRM uses to determine which records to display in the pick applet.

About Drilldowns

A *drilldown* is a user interface element that allows the user to click a field and display another view that includes more information about the field. If a field includes a drilldown, then Siebel CRM displays the value in the field as a link. To view an example of a drilldown, do the following:

- 1 In the Siebel client, click the Opportunity screen tab, and then click the Opportunities List link.
- 2 In the Opportunities list, click a name in the Opportunity Name field.

Siebel CRM displays the Opportunity Detail - Contacts View.

You can configure one of the following types of drilldowns:

- **Static.** Always displays the same view.
- **Dynamic.** Can display a view depending on a condition, such as the value of a field.

Dynamic Drilldown

A *dynamic drilldown* is a type of drilldown that displays a view depending on a condition, such as the value of a field. NREC must configure a dynamic drilldown according to the following condition:

- If the opportunity sales stage is not Prospecting, Qualification, or Closing, then display the Opportunity Detail - Contacts view. This is the behavior of the predefined drilldown object.

- If the opportunity sales stage is Prospecting, Qualification, or Closing, then display the Opportunity Detail - Activities view.

A dynamic drilldown requires at least one *dynamic drilldown destination object*. This object defines the conditions that determine when Siebel CRM displays a particular drilldown object. It is a child of the object type known as drilldown object. A drilldown object is a child of an applet.

For a dynamic drilldown, to define each candidate view, you create a drilldown object. To define the condition for each drilldown object, you define one or more dynamic drilldown destinations. A *dynamic drilldown destination* is a child of the drilldown object. If the condition in the dynamic drilldown destination is false, then Siebel CRM uses the parent drilldown as the default.

Process of Configuring Picklists, Drilldowns, and MVGs

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To configure picklists, drilldowns, and MVGs, perform the following tasks:

- 1 [“Creating Static Picklists”](#) on page 109
- 2 [“Creating a Dynamic Picklist”](#) on page 111
- 3 [“Constraining a Picklist”](#) on page 113
- 4 [“Creating a Dynamic Drilldown”](#) on page 114
- 5 [“Configuring a Multi-Value Group”](#) on page 115

Creating Static Picklists

This task is a step in [“Process of Configuring Picklists, Drilldowns, and MVGs”](#) on page 109.

In this topic, you create a static picklist for each of the following fields in the Opportunity business component:

- Price Range
- Bathrooms
- Bedrooms
- Square Feet

For more information, see [“Static Picklist”](#) on page 107.

To create static picklists

- 1 In Siebel Tools, choose the File menu, and then the New Object menu item.
- 2 In the General tab of the New Object Wizards dialog box, click Pick List, and then click OK.

- 3 In the Pick List dialog box, enter values from the following table, and then click Next.

Property	Value
Project	NREC Configuration
Business Component	Opportunity
Field	Price Range

Note that you created this field in [“Adding Columns to the Opportunity Table”](#) on page 100.

- 4 Choose the Static option, and then click Next.
- 5 In the Pick List Definition dialog box, choose the Create New Pick List option, and then click Next.
- 6 In the next Pick List Definition dialog box, enter Price Range for the name, choose the Create New List Of Values option, and then click Next.
- 7 In the List of Values dialog box, enter Price for the name.
- 8 In the Enter a Value window of the List of Values dialog box, enter the following value, and then click Enter:
- 0 - 100,000
- 9 Repeat [Step 8](#) until the Current Values window in the List of Values dialog box contains all of the following values:
- 100,000 - 150,000
 - 150,000 - 200,000
 - 200,000 - 250,000
 - 250,000 - 500,000
 - 500,000 - 750,000
 - 750,000 - 1,000,000
 - More than 1,000,000
- 10 Click Next.
- 11 In the Pick List Definition dialog box, do the following:
- a Leave the Search Specification property empty.
 - b Enter a comment in the Comment field that describes the purpose of this picklist.
 - c Make sure the Bounded Pick List option does not contain a check mark.
- A picklist can be bounded or unbounded:
- Bounded.** Restricts the user to only choose a value from the list.
 - Unbounded.** Allows the user to choose a value from the list or type a value in the field.

12 In the Finish dialog box, review your configuration, and then click Finish.

The wizard does the following:

- Creates the Pick List object and related objects
- Adds the list of values to the database

13 Repeat [Step 1](#) through [Step 12](#) for each of the following picklists:

Pick List Name	Values
Bathrooms	Use the following values: <ul style="list-style-type: none"> ■ 1 ■ 2 ■ 3
Bedrooms	Use the following values: <ul style="list-style-type: none"> ■ 1 ■ 2 ■ 3 ■ 4
Square Feet	Use the following values: <ul style="list-style-type: none"> ■ Under 1000 ■ 1000 to 2000 ■ 2000 to 3000 ■ 3000 to 4000 ■ Over 5000

14 Compile and test your changes.

For more information, see [“Compiling Projects” on page 184](#).

Creating a Dynamic Picklist

This task is a step in [“Process of Configuring Picklists, Drilldowns, and MVGs” on page 109](#).

In this topic, you create a dynamic picklist. For more information, see [“Dynamic Picklist” on page 107](#).

To create a dynamic picklist

- 1** Lock the projects for the originating business component and the pick business component:
 - a** In Siebel Tools, in the Object Explorer, click Business Component.

- b** In the Business Components list, locate the Action business component.
- c** Right-click the Action business component, and then choose the Lock Project menu item.
- d** Change the value of the Project property to NREC Configuration.

You already locked the project that the Opportunity business component references. For more information, see [“Adding Fields to the Opportunity Business Component” on page 101](#).

- 2** Choose the File menu, and then the New Object menu item.
- 3** In the General tab of the New Object Wizards dialog box, click Pick List, and then click OK.
- 4** In the Pick List dialog box, enter the values from the following table, and then click Next.

Property	Value	Description
Project	NREC Configuration	Project to which the picklist will belong.
Business Component	Action	Business component that contains the field for which you are defining the picklist.
Field	Opportunity	Field for which you are defining the picklist.

- 5** In the Pick List Type dialog box, choose the Dynamic option, and then click Next.
- 6** In the Pick List Definition dialog box, choose the Create New Pick List option, and then click Next.
- 7** In the next Pick List Definition dialog box, enter the values from the following table, and then click Next.

Property	Value	Description
Business Component	Opportunity	This is the business component from which you are drawing values to display to the user.
Field	Created	Siebel CRM uses the field you specify to sort records in the pick applet.
Name	Opportunity NREC	Name of the picklist.
Search Specification	Leave empty	This is optional.
Comment	Describe this picklist	Enter a comment that will help you identify this picklist in the future.

- 8** In the Pick List Specifications dialog box, accept the defaults, and then click Next.
- 9** In the Pick Map dialog box, choose the business component fields:
 - a** In the Select a Field in the Originating Business Component list, choose Opportunity, and then click Add.
 - b** In the Select a Field in the Pick Business Component list, choose Opportunity, and then click Add.
 - c** In the Select a Field in the Pick Business Component list, choose Oppty Id, and then click Add.

d Click Next.

Siebel CRM uses the information you enter to define the pick maps for the picklist. These pick maps are the mappings between the source field and the target field.

10 In the Finish dialog box, review the information you entered, and then click Finish.

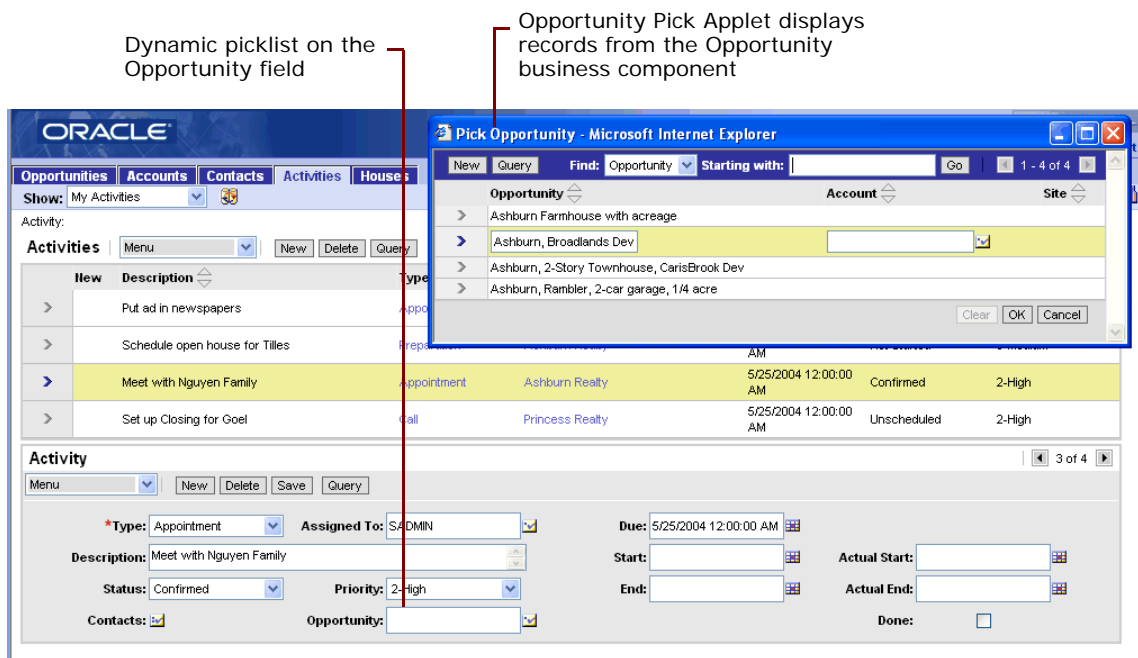
The Pick List wizard creates the picklist, pick maps, and determines if a pick applet already exists that it can use to display the records. For the NREC example, an Opportunity pick applet is already defined, so Siebel Tools does not open the New Applet Wizard.

11 Compile your changes.

Make sure you compile the Activity project. For more information, see [“Compiling Projects” on page 184](#).

12 Test your changes.

The results must resemble the following illustration:



Constraining a Picklist

This task is a step in [“Process of Configuring Picklists, Drilldowns, and MVGs” on page 109](#).

In the NREC example, you constrain the list of records in the Opportunity Pick Applet according to the Account field. For more information see, [“Constrained Picklist” on page 108](#).

To constrain a picklist

1 In the Object Explorer, click Business Component.

- 2 In the Business Components list, locate the Action business component.

In this example, the Action business component is the originating business component, and the Opportunity business component is the pick business component.

- 3 In the Object Explorer, expand the Business Component tree, and then click Field.

- 4 In the Fields list, locate the Opportunity field.

At runtime, Siebel CRM calls the pick applet from the Opportunity field.

- 5 In the Object Explorer, expand the Field tree, and then click Pick Map.

- 6 In the Pick Maps list, create a new pick map using values from the following table.

Property	Value
Field	Account Name
Picklist Field	Account
Constrain	Contains a check mark

At run-time, if the value in the Account Name field in the originating business component matches the Account field in the picklist business component, then Siebel CRM displays the record in the pick applet.

- 7 Compile your changes.

For more information, see [“Compiling Projects” on page 184](#).

- 8 Test your changes.

Enter data for accounts, activities, and opportunities. Choose an activity and then open the Opportunity picklist. The Pick Opportunity list must display only the opportunities for the chosen account.

Creating a Dynamic Drilldown

This task is a step in [“Process of Configuring Picklists, Drilldowns, and MVGs” on page 109](#).

This topic describes how to create a dynamic drilldown. For more information, see [“About Drilldowns” on page 108](#).

To create a dynamic drilldown

- 1 Create a dynamic drilldown object:

- a In the Object Explorer, click Applet.

- b In the Applets list, locate the Opportunity List Applet (SCW) applet.

This applet contains the link that the user clicks to drill down to a view.

- c In the Object Explorer, expand the Applet tree, and then click Drilldown Object.

- d In the Drilldown Objects list, create a new record using values from the following table.

Property	Value
Name	Activities
Hyperlink Field	Name
View	Opportunity Detail - Activities (SCW)

- 2 Create dynamic drilldown destination objects:

- a In the Drilldown Objects list, locate the Original drilldown object.

If the conditions in the dynamic drilldown destinations are false, then Siebel CRM displays the view that you define in the View property of the Original drilldown object.

- b In the Object Explorer, expand the Drilldown Object tree, and then click Dynamic Drilldown Destination.

- c In the Dynamic Drilldown Destinations list, enter new records using values from the following table.

Name	Field	Value	Destination Drilldown Object	Sequence
Prospecting	Sales Stage	01 - Prospecting	Activities	1
Qualification	Sales Stage	02 - Qualification	Activities	2
Closing	Sales Stage	03 - Closing	Activities	3

- 3 Compile your changes.

For more information, see [“Compiling Projects” on page 184](#).

- 4 Test your changes.

In the Siebel client, navigate to the Opportunity List Applet, and then drill down on the name field. Siebel CRM must display the Opportunity Detail - Contacts (SCW) view or the Opportunity Detail - Activities (SCW), depending on the value of the Sales Stage field.

Configuring a Multi-Value Group

This task is a step in [“Process of Configuring Picklists, Drilldowns, and MVGs” on page 109](#).

A *multi-value group* is a user interface element that incorporates child data in an applet. It allows the user to display multiple child records for a single parent record. Instead of using a master-detail view, the user clicks the select button in a multi-value field, and then Siebel CRM displays an MVG applet that includes a set of child records.

For example, to display the contacts associated with each opportunity, NREC must create an MVG on the Opportunity Form Applet (SCW). To view the results of the configuration work, see [Figure 14 on page 118](#).

In this example, you use the Multi Value Group Wizard. This wizard automatically creates the following objects:

- Multi-Value Link
- Multi-Value Field
- Multi-Value Group Applet, if necessary

After you use the wizard, you configure Siebel CRM to display the multi-value field in the Siebel client. The Opportunity Form (SCW) applet already includes a Contact field. You modify it so it displays the Contact MVG.

To configure a multi-value group

- 1 Create the multi-value group:
 - a In Siebel Tools, choose the File menu, and then the New Object menu item.
 - b In the General tab of the New Object Wizards dialog box, click MVG and then click OK.
 - c In the Multi Value Group dialog box, enter values from the following table, and then click Next.

Property	Value	Description
Project	NREC Configuration	Project to which the MVG belongs.
Master Business Component	Opportunity	The name of the parent business component. For example, Opportunity is the parent business component in the Opportunity/Contact link.

- d In the next Multi Value Group dialog box, enter the values from the following table, and then click Next.

Property	Value	Description
Detail Business Component	Contact	The name of the detail business component. For example, Contact is the child business component in the Opportunity/Contact link.
Multi Value Link Name	NREC Contact	Unique name for the multi value link.

- e In the Direct Links dialog box, click the Opportunity/Contact link and then click Next.
This dialog box displays every link that describes a master-detail relationship between the business components you define in [Step c](#) and [Step d](#).
 - f In the Primary ID Field dialog box, accept all default values, and then click Next.
As an option, you can use a primary to improve performance. It reduces the number of SQL queries that Siebel CRM performs to enter items in the MVG for each record.
 - g In the Multi Value Link dialog box, accept all default values, and then click Next.

- h** In the Multi Value Fields dialog box, use the list to choose a field from the detail business component that the wizard uses to create the multi-value field in the master business component:
 - ❑ In the Select a Field in the Destination Business Component of the Multi Value Link list, choose Last Name.
 - ❑ For the name of the multi value field, enter NREC Contact Last Name.
 - ❑ Click Add, and then click Next.

i In the Finish dialog box, review the information you entered and then click Finish.

The Multi Value Group Wizard creates the objects for the multi-value group. It also searches the repository for a predefined multi-value group applet that Siebel CRM can use to display data in the Siebel client. If the wizard does not find an appropriate applet, then Siebel Tools opens the MVG Applet Wizard so you can define one. In the NREC example, the Contacts MVG Applet already exists, so Siebel Tools does not open the MVG Applet wizard.

2 Modify the opportunity form:

- a** In the Object Explorer, click Applet.
- b** In the Applets list, locate the Opportunity Form (SCW) applet.
- c** In the Object Explorer, expand the Applet tree, and then click Control.
- d** In the Controls list, locate the Contact control.
- e** Modify the properties using values from the following table.

Property	Value
Field	NREC Contact Last Name
MVG Applet	Contact MVG Applet
Runtime	Contains a check mark

3 Compile your changes.

For more information, see ["Compiling Projects" on page 184](#).

4 Test your changes.

Figure 14 illustrates how Siebel CRM must display the multi-value field in the applet with a select button next to the field. When you click the select button, Siebel CRM must display the MVG applet.

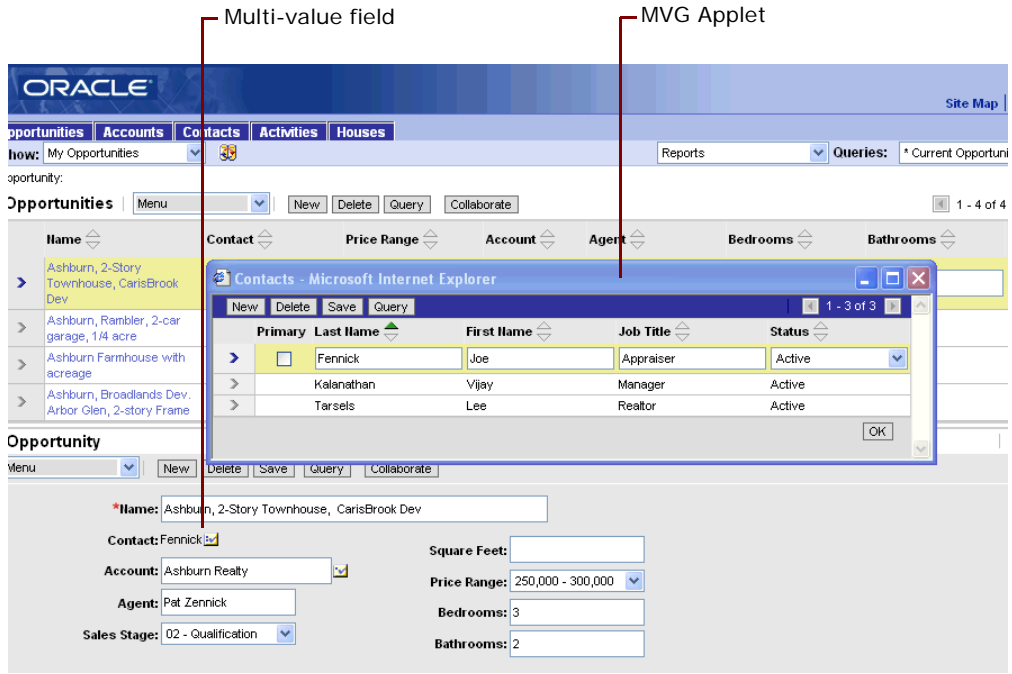


Figure 14. Multi-Value Field

8

Configuring Logos, Banners, and Colors

This chapter describes how to configure logos, banners, and colors. It includes the following topics:

- [Overview of Configuring Logos, Banners, and Colors on page 119](#)
- [Process of Configuring Logos, Banners, and Colors on page 122](#)

Overview of Configuring Logos, Banners, and Colors

This topic describes an overview of how to configure logos, banners, and colors.

About User Interface Elements

Figure 15 illustrates the user interface elements that you modify in this chapter.

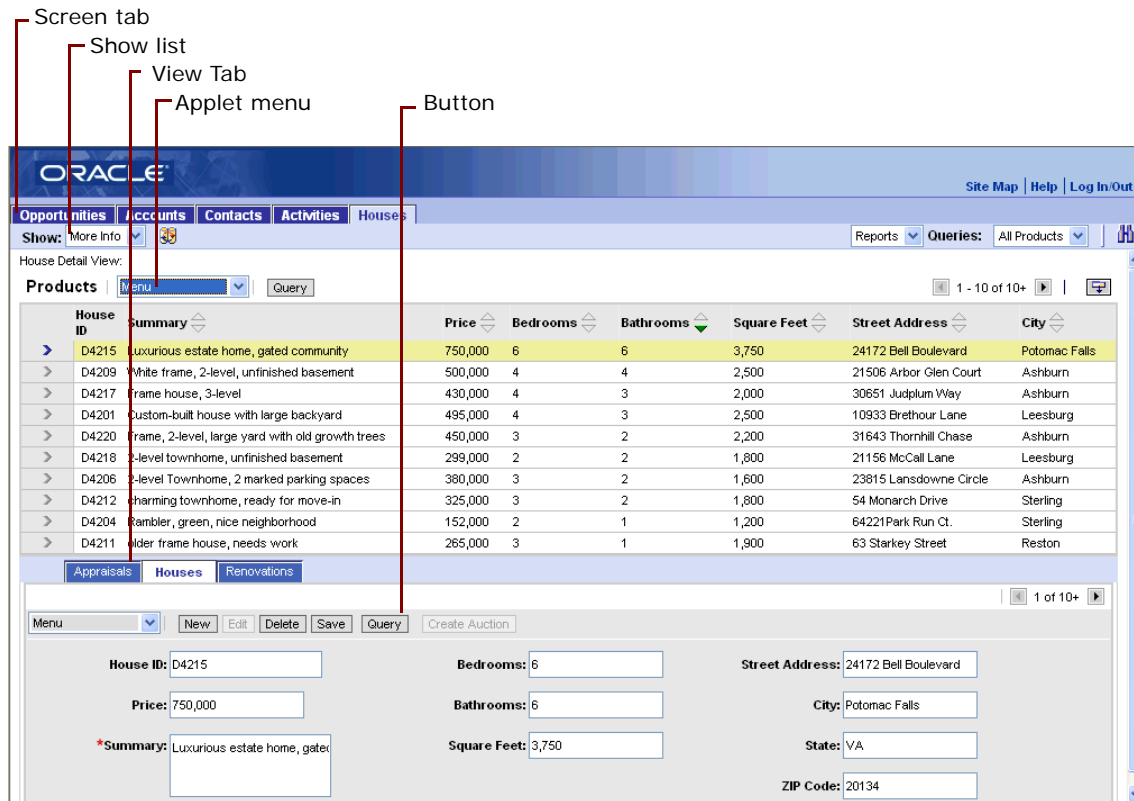


Figure 15. User Interface Elements

Figure 16 illustrates the frames you modify in this chapter.

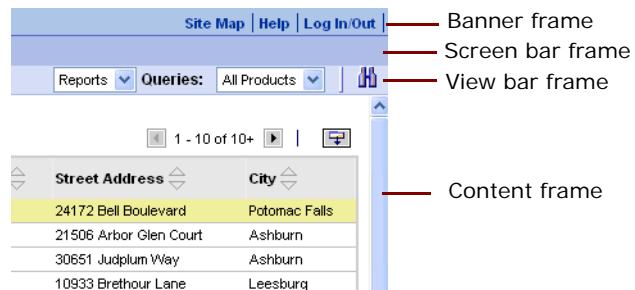


Figure 16. Frames

How Siebel CRM Stores Web Templates and Related Files

Siebel CRM stores Web templates and files that support Web templates under the root directory of your Siebel installation. Siebel CRM does not store Web templates and supporting files in the Siebel repository. Graphics files and cascading style sheets are examples of files that support a Web template. Because these files are not in the Siebel repository, the Siebel Tools check-in and check-out process does not control them. If multiple developers modify these files, then it is recommended that you use a third-party source file control application to make sure a developer does not overwrite customizations that another developer creates.

Where Siebel CRM Stores Web Templates and Related Files

Siebel CRM installs Siebel Web templates and supporting files with the Siebel Server, Siebel Tools, the Siebel Mobile Web Client, and the Siebel Developer Web Client.

In the directory paths listed in this topic, *language_code* is the three letter code of the language pack for the installed software. For example, ENU is the code for English and FRA is the code for French.

Siebel Tools

Siebel Tools uses the files in the Siebel Tools installation directory when it renders objects when you use the Preview option in the Edit Web Layout window. The following table lists the locations of the files in the Siebel Tools installation directory:

File	Directory
Image files	<i>tools_root</i> \PUBLIC\ <i>language_code</i> \IMAGES
Style Sheets	<i>tools_root</i> \PUBLIC\ <i>language_code</i> \FILES
Web templates	<i>tools_root</i> \WEBTEMPL

Siebel Server

The Siebel Web engine uses files in the Siebel Server installation directory when it renders objects in the Siebel client. The following table lists the locations of the files in the Siebel Server installation directory:

File	Directory
Image files	<i>siebsrvr_root</i> \webmaster\images\ <i>language_code</i>
Style Sheets	<i>siebsrvr_root</i> \webmaster\files\ <i>language_code</i>
Web templates	<i>siebsrvr_root</i> \WEBTEMPL

Siebel Mobile Client or Siebel Developer Web Client

Siebel CRM uses files in the installation directories of the Siebel Mobile Web Client or Siebel Developer Web Client when it renders objects in the client. The following table lists the locations of these files:

File	Directory
Image files	<i>client_root</i> \PUBLIC\language_code\IMAGES
Style Sheets	<i>client_root</i> \PUBLIC\language_code\FILES
Web templates	<i>client_root</i> \WEBTEMPL

About Developing Web Templates

If you modify a Siebel Web template, then you typically work with the files stored on your local computer. After you test the changes locally, you copy them to the appropriate directory on the server to make them available to the Siebel Web Engine.

The typical development process for Siebel Web templates varies in the following ways:

- **Simple Web template modifications.** The modifications you make in this chapter are simple. For simple changes, such as moving the position of a control placeholder, it is easiest to make the changes to the Web templates in the Siebel Tools directory. Siebel Tools parses the file when you save it and identifies any format errors that might occur. The Preview functionality displays the Web template and the objects that are mapped to it as they would be rendered in the Siebel client. For minor changes, it is easier to debug the files this way rather than starting the Siebel Mobile Web Client and navigating to the applet where Siebel CRM displays the changes. After you use Siebel Tools to verify the change, you can manually copy the changed template to the Web template directory of your Siebel Mobile Web Client, start the application, and do final testing.
- **Complex Web template development.** For complex template development, such as creating a new Web template or significantly modifying the layout of a predefined template, you must create a prototype in HTML, translate the various parts of the template to SWE, and then debug. For more information, see *Siebel Developer's Reference*.

Process of Configuring Logos, Banners, and Colors

This process is a step in “[Roadmap for Developing a Siebel Application](#)” on page 31.

To configure logos, banners, and colors, perform the following tasks:

- 1 “[Examining Business Requirements for Configuring Logos, Banners, and Colors](#)” on page 123
- 2 “[Adding a Custom Logo to the Banner](#)” on page 124
- 3 “[Modifying the Banner Color](#)” on page 126
- 4 “[Modifying the Screen Bar and View Bar Colors](#)” on page 127

5 “Modifying Applet Colors” on page 128

For more information about content in these topics, see *Configuring Siebel Business Applications*.

Examining Business Requirements for Configuring Logos, Banners, and Colors

Figure 17 includes a screen capture of a typical banner and colors.

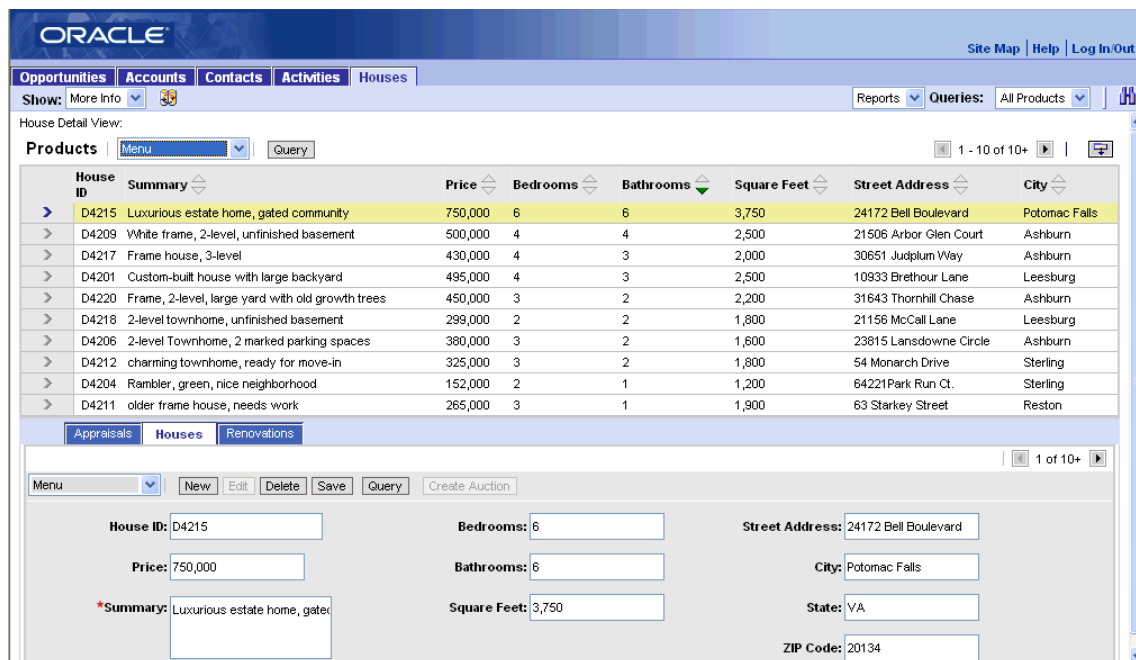


Figure 17. Screen Capture of a Typical Banner and Colors

Figure 18 includes a screen capture of a banner and colors you must modify to meet NREC requirements.

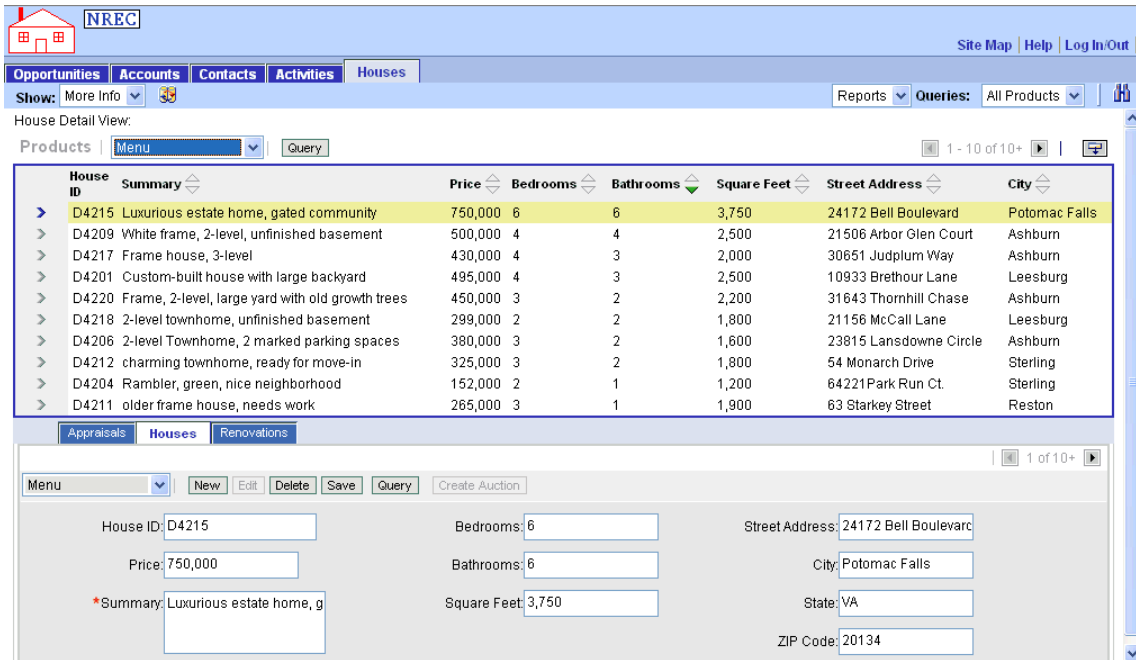


Figure 18. Screen Capture of Banner and Colors Modified for NREC

Adding a Custom Logo to the Banner

This task is a step in “Process of Configuring Logos, Banners, and Colors” on page 122.

The banner is the top frame in the application frame set. It contains the application banner and the Siebel logo. The banner frame includes a Siebel Web template file that you can modify.

Figure 19 includes a screen capture that includes the custom NREC logo.



Figure 19. Banner Frame with a Custom Logo

To add a custom logo to the banner

- 1 In Windows Explorer, add the image file that includes the logo to the images directory for Siebel Tools and the Siebel client.

For example:

- C:\Siebel\8.2\Client_1\PUBLIC\enu\IMAGES
- C:\Siebel\8.2\Tools_1\PUBLIC\ENU\IMAGES

- 2 Rename the custom image file you added in [Step 1](#) to new_banner_logo.gif.
- 3 Make a backup copy of the dCCFrameBanner.swt file.
The dCCFrameBanner.swt file is located in Siebel Tools Web template directory. For example, C:\Siebel\8.2\Tools_1\WEBTEMPL. This file is the banner frame template for a Siebel application.
- 4 In Siebel Tools, make sure you set up the default HTML editor and display the Bitmap Category object type.
For more information, see [“Setting Up Siebel Tools” on page 60](#).
- 5 Edit the Web template:
 - a Choose the View menu, Windows, and then the Web Templates Window menu item.
 - b In the Web Template Explorer, click dCCFrameBanner.
 - c Siebel Tools displays the HTML contents of the dCCFrameBanner Web template in the Web Template File window.
 - d Right-click in the Web Template File window, and then choose the Edit Template menu item.
Siebel Tools opens your HTML editor and displays the code of the dCCFrameBanner.swt file. To use this feature, you must set up the default HTML editor. For more information, see [“Setting Up Siebel Tools” on page 60](#).
 - e In the Web Template File window, locate the following code:


```
<td val i gn="mi ddl e" al i gn="l eft"><span><swe: i mage name="BANNER_LOGO" category="HTML Control Icons" /></span></td>
```
 - f Change BANNER_LOGO to NEW_BANNER_LOGO.
 - g Save your work, and then close the HTML editor.
 - h Close the Web Template File window and the Web Template Explorer window.
- 6 Add a bitmap that references your new file:
 - a In Object Explorer, click Project.
 - b In the Projects list, query the Name property for Bitmap, and then make sure the Locked property contains a check mark.
 - c In Object Explorer, click Bitmap Category.
 - d In the Bitmap Categories list, query the Name property for HTML Control Icons.
 - e In Object Explorer, expand the Bitmap Category tree, and then click Bitmap.
 - f In the Bitmaps list, query the Name property for BANNER_LOGO.
 - g Right-click the BANNER_LOGO bitmap, choose Copy, and then set properties for the new record using values from the following table.

Property	Value
Name	NEW_BANNER_LOGO
Filename	new_banner_logo.gif

- 7 Copy the dCCFrameBanner.swt file from the Web templates directory for Siebel Tools to the Web templates directory for the Siebel client.

For example, copy the dCCFrameBanner.swt file from the C:\Siebel\8.2\Tools_1\WEBTEMPL directory to the C:\Siebel\8.2\Client_1\WEBTEMPL directory.

- 8 Open your browser and then clear the browser cache.

For example, in Internet Explorer 7, do the following:

- a Choose the Tools menu, and then the Internet Options menu item.
- b In the Browsing History section, click Delete.
- c In the Delete Browsing History dialog box, click Delete Files, and then click Close.

- 9 Open the Siebel client.

For the NREC example, open the Partner Portal application.

- 10 Make sure the banner resembles the screen capture in [Figure 19 on page 124](#).

Modifying the Banner Color

This task is a step in [“Process of Configuring Logos, Banners, and Colors” on page 122](#).

To modify color and other styles in a Siebel Web template, you use a cascading style sheet. To modify the color of the banner frame, you change code in the cascading style sheet for your Siebel application.

To modify the banner color

- 1 In Windows Explorer, navigate to the files directory for the Siebel client.

For example, C:\Siebel\8.2\Client_1\PUBLIC\enu\FILES.

- 2 Open the dCCmain.css file.

The dCCmain.css file is the cascading style sheet for Siebel Customer and Partner Applications. The main.css file is the cascading style sheet for Employee applications.

- 3 In the Banner Definitions section, locate the following code:

```
. banner,  
TD. banner TD A,  
TD. banner TD A: visited,  
TD. banner TD A: hover {background-color: #afc2ea; color: #003399; text-  
decoration: none; font-weight: bold; }
```

This code defines the banner selector and properties.

- 4 Change the value of the background-color attribute and the color property. Use the following code:

```
. banner,  
TD. banner TD A,  
TD. banner TD A: vi si ted,  
TD. banner TD A: hover {background-col or: #BDD3FF; col or: #333399; text-  
decorati on: none; font-wei ght: bol d; }
```

- 5 Locate the following code:

```
. appl i cati onMenu {background: #123783; }
```

- 6 Change the value of the background attribute. Use the following code:

```
. appl i cati onMenu {background: #BDD3FF; }
```

- 7 Save the file.

- 8 Clear your the browser cache and then open the Siebel client.

For the NREC example, open the Partner Portal application.

- 9 Verify that Siebel CRM displays the banner frame with the new background color, as illustrated in the following screen capture:



Modifying the Screen Bar and View Bar Colors

This task is a step in [“Process of Configuring Logos, Banners, and Colors” on page 122](#).

NREC requires that the background color for the screen bar and view bar match the background color of the other user interface elements of the banner frame.

The screenbar and viewbar frames contain the screen tabs and show list. For example, see [Figure 16 on page 120](#). These frames contain color elements that the cascading style sheet controls.

The dccsrnbar_back.gif image file controls the background color of the screenbar frame in a customer application or partner application. This image is a tiling background that Siebel CRM uses to make sure that the background displays appropriately in Internet Explorer and other browsers.

The Tier2On and Tier2Rule attributes control the viewbar background.

Note that you can also modify the background color of a view frame. You modify the bgcolor tag in the following files:

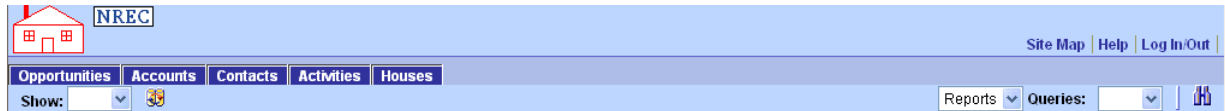
- CCHTMLHeader.swt for an employee application
- dCCHTMLHeader.swt for a customer application

To modify the screen bar and view bar colors

- 1 In the dCCmain.css file, use the following code to modify the value for the background color:

```
. tier2Back {background-color: #BDD3FF; }  
TD. tier2Rule {background-color: #003366; }  
TR. tier2On, TD. tier2On,  
TD. tier2OnNormal  
font-size: 9pt;  
color: #000000;  
text-decoration: none;  
background-color: #BDD3FF;  
font-weight: bold; }
```

- 2 Save the dCCmain.css file, clear the browser cache, and then open the Siebel client.
- 3 Verify that Siebel CRM displays the background color on the screen bar and view bar, as illustrated in the following screen capture:



Modifying Applet Colors

This task is a step in “Process of Configuring Logos, Banners, and Colors” on page 122.

A Siebel application includes the following major applet styles:

- Parent
- Child
- Grandchild

The applet styles define the color of the tab that Siebel CRM displays at the top of the applet and the color of the control bar where Siebel CRM displays applet menus and buttons. The cascading style sheet includes the following items for each applet style:

- **AppletStyle1.** The selector for the parent applet style.
- **AppletStyle3.** The selector for the child applet style.
- **AppletStyle7.** The selector for the grandchild applet style.

Attributes define each applet style. To modify each applet style, you modify the definitions for the following items:

- **AppletButtons.** The bar where Siebel CRM displays applet buttons.
- **AppletBorder.** The border around the content of the applet.
- **AppletBack.** The background color of the applet content.
- **AppletTitle.** The area where Siebel CRM displays the applet title.

The modification you make for a standard interactivity applet is different from the modification you make for a high interactivity applet.

Modifying Applet Colors for Standard Interactivity Applets

In this topic, you modify applet colors for standard interactivity applets.

To modify applet colors for standard interactivity applets

- 1 In the dCCmain.css file, locate the following selectors and then modify the attributes using values from the following table.

Selector	Attribute	Modified Value
AppletStyle1.AppletButtons	Background-color	#3366CC
AppletStyle1.AppletBorder	Background-color	#3366CC
Modify the following selectors:	Background-color	#3366CC
■ AppletStyle1 .AppletTitle	Color	#FFFFFF
■ AppletStyle1 TD.AppletTitle A		
■ AppletStyle1 TD.AppletTitle A:visited		
■ AppletStyle1 TD.AppletTitle A:hover		

- 2 Locate the following selectors and then modify the attributes using values from the following table.

Selector	Attribute	Modified Value
AppletStyle3 .AppletButtons	Background-color	#BDD3FF
AppletStyle3 .AppletBorder	Background-color	#BDD3FF
Modify the following selectors:	Background-color	#BDD3FF
■ AppletStyle3 .AppletTitle		
■ AppletStyle3 TD.AppletTitle A		
■ AppletStyle3 TD.AppletTitle A:visited		
■ AppletStyle3 TD.AppletTitle A:hover		

- 3 Locate the following selectors and modify the attributes using values from the following table.

Selector	Attribute	Modified Value
AppletStyle7 AppletButtons	Background-color	#BDD3FF
AppletStyle7 AppletBorder	Background-color	#BDD3FF

Selector	Attribute	Modified Value
Modify the following selectors:	Background-color	#BDD3FF
■ AppletStyle7 .AppletTitle	Color	#003399
■ AppletStyle7 TD.AppletTitle A		
■ AppletStyle7 TD.AppletTitle A:visited		
■ AppletStyle7 TD.AppletTitle A:hover		

- 4 Save your changes.
- 5 Clear the browser cache and then open the Siebel client.
- 6 Make sure Siebel CRM displays the custom background color on the parent and child applets.
Figure 18 on page 124 illustrates the custom colors.

Modifying Applet Colors for High Interactivity Applets

In this topic, you modify the background color for high interactivity applets.

To modify applet colors for high interactivity applets

- 1 In the main.css file, locate the following code:

```
. Selected TR. AppletButtons {background-color: #658AC3; color: #ffffff; }  
. Selected TR. AppletButtons TD. AppletTitle,  
. Selected TR. AppletButtons TD. AppletTitle A,  
. Selected TR. AppletButtons TD. AppletTitle A: link,  
. Selected TR. AppletButtons TD. AppletTitle A: visited,
```
- 2 Change the value in the background-color attribute to the custom color.
- 3 Save the file, and then test your changes.

9

Configuring Siebel CRM to Use Data From an External Data Source

This chapter describes how to configure Siebel CRM to use data from an external data source. It includes the following topic:

- [Process of Configuring Siebel CRM to Use Data From an External Data Source on page 131](#)

Process of Configuring Siebel CRM to Use Data From an External Data Source

This process is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

To configure Siebel CRM to use data from an external data source, perform the following tasks:

- 1 [“Examining Business Requirements for Using Data from an External Data Source” on page 131](#)
- 2 [“Creating a Business Service” on page 132](#)
- 3 [“Creating a Virtual Business Component” on page 135](#)
- 4 [“Creating a Link Between Houses and House Renovations” on page 136](#)
- 5 [“Modifying the Business Object” on page 137](#)
- 6 [“Displaying Custom Fields in the Siebel Client” on page 137](#)
- 7 [“Compiling and Testing Your Changes” on page 139](#)
- 8 [“Documenting Your Configuration in an Entity Relationship Diagram” on page 140](#). For more information about the content described in this chapter, see *Overview: Siebel Enterprise Application Integration* and *Integration Platform Technologies: Siebel Enterprise Application Integration*.

Examining Business Requirements for Using Data from an External Data Source

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131](#).

Assume NREC stores house renovation data in a comma-delimited text file that a third-party vendor provides to NREC. NREC must retrieve the data from this file and display it in the Partner Portal application. Partner agents and NREC employees can use it to display a history of renovation information for each house stored in the NREC Siebel database. The text file includes the following information:

```
RENOVATION HOUSE_ID="1F-X234" DATE="07/13/2008" DESCRIPTION="Remodel ed ki tchen"
```

```
RENOVATION HOUSE_ID="1F-A431" DATE="07/07/2009" DESCRIPTION="Added second story"  
RENOVATION HOUSE_ID="1F-L134" DATE="08/01/2010" DESCRIPTION="Added two rooms"
```

Figure 20 includes a screen capture that illustrates how Siebel CRM must display the data from the external file in the Siebel client. It display this data in a typical master-detail view that includes a house record in the top applet and a list of renovations in the bottom applet.



Figure 20. Applet Displaying Data Brought in from an External File

About Virtual Business Components

A *virtual business component* (VBC) is a type of business component that brings data from an external data source to a Siebel application. For example, you can use a VBC to represent data from the following data sources:

- Back office application
- Legacy application
- Transaction service
- Web site

The example in this chapter uses a simple file as an external data source instead of an external database or external application.

Creating a Business Service

This task is a step in “Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131.

A *business service* is an object that defines a set of functionality. For example, a business service can perform certain tasks, such as moving data or converting data formats. Although a business service is stored in the Siebel repository, it does not interact with tables and data. Instead, it interacts with other objects.

A virtual business component uses a business service to help it manipulate external data. The business service defines the methods, properties, and states that determine the behavior of the virtual business components.

You can use a predefined business service, such as the XML Gateway, or you can create your own custom business service. You create virtual business components and business services in Siebel Tools. In the NREC example, you use Siebel Tools to create a business service that reads data from and writes data to a comma-delimited text file.

For more information, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

To create a business service

- 1 In Siebel Tools, create a symbolic string for the following text:

NREC Text File Handler

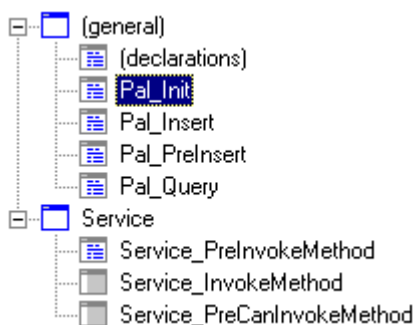
For more information, see [“Configuring Text for Labels and Other User Interface Elements” on page 80](#).

- 2 In the Object Explorer, click Business Service.
- 3 In the Business Services list, create a new record using values from the following table.

Property	Value
Name	NREC Text File Handler
Project	NREC Configuration
Class	CSSService
Display Name	NREC Text File Handler

- 4 In the Business Services list, right-click the NREC Text File Handler business service, and then choose the Edit Server Scripts menu item.
- 5 In the Scripting Language dialog box, choose Visual Basic and then click OK.
- 6 In the script editor, enter general custom functions:
 - a Expand the general tree and then click declarations.
 - b In the text editor, enter the code for the declaration custom function.
For more information, see [“Code for the Declaration Function” on page 199](#).
 - c Choose the File menu, and then the Save menu item.
 - d Repeat [Step a](#) through [Step c](#) for each of the following custom functions:

- ❑ **Pal_Init.** For more information, see [“Code for the Pal_Init Function” on page 199.](#)
- ❑ **Pal_Insert.** For more information, see [“Code for the Pal_Insert Function” on page 200.](#)
- ❑ **Pal_PreInsert.** For more information, see [“Code for the Pal_PreInsert Function” on page 201.](#)
- ❑ **Pal_Query.** For more information, see [“Code for the Pal_Query Function” on page 202.](#)
- ❑ After you enter these functions, the script editor must resemble the following screen capture:



- 7 In the script editor, enter a service custom function:
 - a Expand the service tree and then click Service_PreInvokeMethod.
 - b In the text editor, delete all existing code, and then enter the code for the Service_PreInvokeMethod custom function.

For more information, see [“Code for the Service Function” on page 205.](#)
 - c Choose the File menu, and then the Save menu item.

Table 10 describes the custom business service methods you add.

Table 10. Business Service Methods

Method	Description
Init	Pal_Init creates the initial link between the columns in the VBC and the columns in the external data source.
Insert	Pal_Insert inserts a record in the external data source.
PreInsert	Pal_PreInsert provides default values for new rows. In the NREC example, there are no default values.
Query	Pal_Query queries the external data source and returns rows that match a given value. In the NREC example, the method queries the text file and returns rows that include a matching value for the current House_ID field.
Service_PreInvokeMethod	Service_PreInvokeMethod handles requests that come to the business service and calls other functions as appropriate.

Table 10. Business Service Methods

Method	Description
Update	Updates a row in the external data source.
Delete	Deletes a row in the external data source.

Creating a Virtual Business Component

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131](#).

If you create a virtual business component you do not use the Business Component Wizard for the following reasons:

- The wizard forces you to choose a table that the business component references.
- The wizard does not set the class to CSSBCVExtern.

To create a virtual business component

- 1 In Siebel Tools, make sure the Business Component User Prop object type is displayed.

For more information, see [“Setting Up Siebel Tools” on page 60](#).

- 2 Define the business component:

- a In the Object Explorer, click Business Component.
- b In the Business Components list, right-click and then choose the New Record menu item.
- c Create a new record using values from the following table.

Property	Value
Name	NREC Renovations VBC
Project	NREC Configuration
Class	CSSBCVExtern

- 3 Add fields to the business component:

- a In the Object Explorer, expand the Business Components tree, and then click Single Value Field.
- b In the Single Value Fields list, create a new record for each row in the following table.

Name	Type	Comments
Date	DTYPE_TEXT	Date the renovation was recorded.
Description	DTYPE_TEXT	Description of the renovation.
House_ID	DTYPE_TEXT	House Identification Number.

- 4 Define user properties for the business component:
 - a In the Object Explorer, in the Business Component tree, click Business Component User Prop.
 - b Add a new business component user property for each row in the following table.

Name	Value
Service Name	NREC Text File Handler
Service Parameters	D:\NREC\nrec.txt; 3; House_ID,Date,Description For more information, see “Service Parameters” on page 136.

Service Parameters

The service parameters define the parameters that Siebel CRM sends to the business service during initialization. The script for the NREC example uses the following format:

path; num_columns; column_names

where:

- *path* is the full path to the file name including the file name
- *num_columns* is number of columns listed in the *column_names* parameter
- *column_names* is a list of column names. You must use a comma to separate each column name.

Note the following format requirements:

- You must use a semicolon to separate each parameter.
- You must not include a space in a comma-delimited string.

Service parameters vary depending on the business service you use. In this example, you define the parameters in the script for the business service.

Creating a Link Between Houses and House Renovations

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131.](#)

The relationship between houses and house renovations is one-to-many. A user can add many renovations for one house. NREC uses the Internal Product business component to store houses. To create the parent-child relationship, you must add a link between the Internal Product business component and the NREC Renovations VBC. For more information, see [“Design Strategy for the Data Layer” on page 29.](#)

To create a link between houses and house renovations

- 1 In the Object Explorer, click Link.

- 2 In the Links list, add a new record using values from the following table.

Property	Value
Name	Internal Product/NREC Renovations VBC
Project	NREC Configuration
Parent Business Component	Internal Product
Child Business Component	NREC Renovations VBC
Source Field	Part #
Destination Field	House_ID

The source field is the unique ID on the parent record. This property is typically Row Id. In this example, the user enters the unique field.

The destination field is the column in the child record that stores the parent ID.

Modifying the Business Object

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131](#).

After you define a link between the Internal Product business component and the NREC Renovations VBC, you can add the Renovation VBC to the Internal Product business object.

To modify the business object

- 1 In the Object Explorer, click Business Object.
- 2 In the Business Objects list, locate the Internal Product business object.
- 3 In the Object Explorer, expand the Business Object tree, and then click Business Object Component.
- 4 In the Business Object Components list, add a new record using values from the following table.

Property	Value
Bus Comp	NREC Renovations VBC
Link	Internal Product/NREC Renovations VBC

Displaying Custom Fields in the Siebel Client

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131](#).

NREC requires a typical master-detail view that displays the House Form applet on top and the Renovations list applet on the bottom. To display the Renovations list applet you add it to a view.

To display custom fields in the Siebel client

- 1 Create a new list applet:
 - a In Siebel Tools, choose the File menu, and then the New Object menu item.
 - b In the New Object Wizards dialog box, click the Applets tab, click List Applet, and then click OK.
 - c Complete the wizard using values from the following table.

Property	Value
Project	NREC Configuration
Applet Name	Renovations List Applet
Display Title	Renovations
Business Component	NREC Renovations VBC
Upgrade Behavior	Admin
Web Templates	Choose the following values: <ul style="list-style-type: none"> ■ Base. Applet List (Base/List Edit) ■ Edit List. Applet List Edit (Edit/New/Query) ■ Edit. Applet List Edit (Edit/New/Query)
Field	Add the following fields: <ul style="list-style-type: none"> ■ House_ID ■ Date ■ Description
Controls	Accept all default values.

- 2 Create a new view:
 - a Choose the File menu, and then the New Object menu item.
 - b In the New Objects Wizard dialog box, click View, and then click OK.
 - c Complete the wizard using values from the following table.

Property	Value
Project	NREC Configuration
Name	House Details - Renovations View
Title	Renovations

Property	Value
Business Object	Internal Product
Upgrade Behavior	Admin
Web Template	View Detail (Parent with Pointer)
Applets	Add the following applets: <ul style="list-style-type: none">■ Product Form Applet■ Renovations List Applet

3 Add the view to a screen:

- a** Create a symbolic string for the following text:

Renovations

For more information, see [“Configuring Text for Labels and Other User Interface Elements” on page 80](#).

- b** In the Object Explorer, click Screen.

- c** In the Screens list, locate the NREC House Screen.

For information, see [“Creating the Houses Screen” on page 87](#).

- d** In the Object Explorer, expand the Screen tree, and then click Screen View.

- e** In the Screen Views list, add a new record using values from the following table.

Property	Value
View	House Details - Renovations View
Type	Detail View
Parent Category	NREC
Viewbar Text	Renovations
Sequence	30
Menu Text	Renovations

Compiling and Testing Your Changes

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131](#).

To compile and test your changes

- 1 Compile your changes.
For more information, see [“Compiling Projects” on page 184](#).
- 2 Register the House Details - Renovations View.
For more information, see [“Registering a New View” on page 186](#).
- 3 Test your work.
The results must resemble [Figure 20 on page 132](#).

Documenting Your Configuration in an Entity Relationship Diagram

This task is a step in [“Process of Configuring Siebel CRM to Use Data From an External Data Source” on page 131](#).

It is recommended that you document in an entity relationship diagram (ERD) the following configuration work:

- Changes to data objects. For more information, see [Chapter 6, “Configuring the House and Opportunity Entities.”](#)
- New virtual business components.

[Figure 7 on page 29](#) illustrates a prototype of the relationships between NREC data objects. You can create an Entity Relationship Diagram in Siebel Tools.

For more information, see *Configuring Siebel Business Applications*.

To document your configuration in an entity relationship diagram

- 1 In Siebel Tools, in the Object Explorer, click Entity Relationship Diagram.
- 2 In the Entity Relationship Diagrams list, right-click and then choose the New Record menu item.
- 3 Add a new record using values from the following table.

Property	Value
Name	NREC House ERD
Project	NREC Configuration
Status	Scope

- 4 In the Entity Relationship Diagrams list, right-click the NREC House ERD, and then choose the Edit Entity Relationship Diagram menu item.
- 5 In the Entity Relationship Designer, drag shapes from the palette and drop them on the canvas until your diagram resembles [Figure 7 on page 29](#).

- 6 Click an entity, and then use the Properties window to set properties for the entity using values from the following table. Each row in this table represents values for a single entity. Repeat this step until you define these properties for all entities.

Name	Business Component
House Renovations (VBC)	NREC Renovations VBC
House (Product)	Internal Product
Appraisals	Appraisals
Buying Interest	Opportunity
Activity Plan	Activity Plan
Contact	Contact
Independent Agency (Partner)	Account Home Search

- 7 Add relationship lines between all entities until the diagram resembles [Figure 7 on page 29](#). Make sure you attach all endpoints. If you attach an end point, then the Entity Relationship Designer changes the end point from white to red.

- 8 Bind each relationship:

- a Right-click a relationship line, and then choose the Bind Entity Relation menu item.
- b In the Bind Relationships dialog box, choose a relationship and then click OK.

If the Bind Relationships dialog box:

- ❑ Lists multiple relationships, then examine the business components and tables listed, and then choose the most appropriate pairing.
- ❑ Lists no relationships, then there is no link or join between the business components.

If you bind a relationship, then Siebel Tools increases the thickness of the relationship line.

- 9 (Optional) To improve legibility, reshape the relationship lines:

- a Choose a relationship line.
- b To add a reshaping point, press Ctrl+A.
- c To adjust the line, move the reshaping point as required.
- d Repeat [Step b](#) and [Step c](#), as required.

- 10 Edit end names for each relationship line:

- a Choose a relationship line.
- b Set the End Name 1 and End Name 2 properties.
For more information, see [Figure 21](#).
- c Repeat [Step a](#) and [Step b](#), as required.

[Figure 21](#) includes a screen capture of a completed entity relationship diagram.

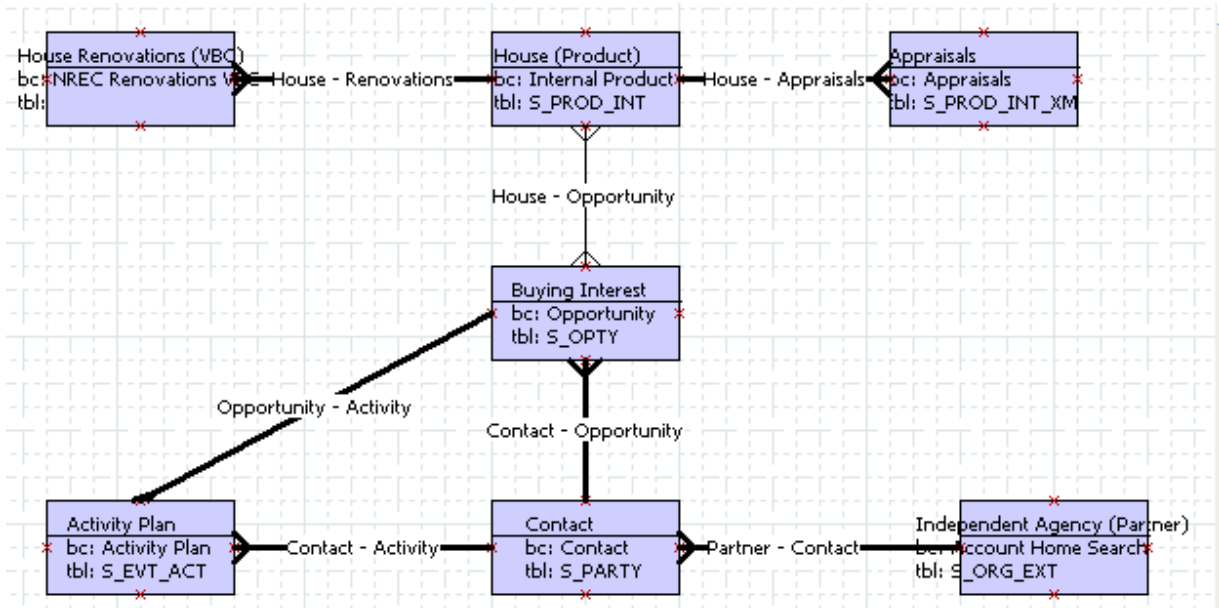


Figure 21. NREC Entity Relationship Diagram in the Entity Relationship Designer

10 Assigning Objects and Automating Business Processes

This chapter describes how to configure Siebel CRM to assign objects and automate business processes. It includes the following topics:

- [Customizing How Siebel CRM Assigns Objects on page 143](#)
- [Process of Automating a Business Process on page 146](#)
- [Filtering Records According to a User Profile on page 155](#)

Customizing How Siebel CRM Assigns Objects

This task is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

In this topic, you create an assignment rule that assigns opportunities to agents depending on the ZIP code. When a user creates a new opportunity, NREC requires Siebel CRM to automatically assign the opportunity to an NREC agent or a partner agent according to the ZIP code. Siebel CRM must assign the opportunity to the real estate agent whose sales region includes the ZIP code that is specified in the Zip Code field.

Siebel Assignment Manager is a Siebel CRM feature that allows you to assign the most qualified person to a specific task. To accomplish this, Assignment Manager matches candidates to assignment objects. To assign the most qualified candidate to each object, Assignment Manager applies assignment rules that you define. You can specify how Siebel CRM uses Assignment Manager to evaluate a record. You can configure Assignment Manager to make assignments in the following ways:

- Interactively in real time
- Dynamically when the user makes a database change
- Periodically in batches

For more information, see *Siebel Assignment Manager Administration Guide*.

To customize how Siebel CRM assigns objects

- 1 Create an assignment rule:
 - a Log in to the Siebel client as an administrator, navigate to the Administration - Assignment screen, and then the Assignment Rules List view.

For more information, see [“Logging In as the Siebel Administrator” on page 185](#).
 - b In the Assignment Rules List, click Menu, and then choose the Columns Displayed menu item.

- c In the Columns Displayed dialog box, make sure the Assignee Filter column is included in the Selected Columns window, and then click Save.
- d In the Assignment Rules List, click New.
- e Enter a new record using values from the following table.

Field	Value	Description
Name	San Francisco	Enter a unique name for the rule.
Object	Opportunity	To choose the object Siebel CRM must assign, do the following: <ul style="list-style-type: none"> ■ In the Objects to be Assigned field, click the select button. ■ In the Assignment Object dialog box, click New. ■ In the Name window, choose Opportunity and then click Save. ■ Click OK.
Rule Group	Default Rule Group	Not applicable.
Minimum Score	0	Although NREC does not require scoring, you must enter a value.
Assignee Filter	All Above Minimum	Choose the filter from the list. This filter causes Siebel CRM to assign all assignees who possess an assignment score that is greater than or equal to the minimum score of the assignment rule. For more information, see <i>Siebel Assignment Manager Administration Guide</i> .

- 2 Specify assignment criteria and values:
 - a In the Assignment Rules List, in the Name field, click San Francisco.
This is the rule you created in [Step d on page 144](#).

- b In the Criteria list, click New, and then define fields values from the following table.

Field	Value	Description
Rule Criterion	Prospect ZIP code	Choose the type of criteria.
Comparison Method	Compare to Object	Choose the method that Siebel CRM uses to match objects to candidates. For example, Compare to Object compares the criteria value to an object attribute.
Inclusion	Include	Choose the method that Siebel CRM uses to match criteria values and candidates.
Required	Always	Determines whether the criteria is required.
Score	0	If the candidate satisfies this criteria, then Siebel CRM adds this score to the total score for the candidate.

Siebel CRM uses the criteria to determine which candidates qualify to receive the assignment. In the NREC example, the criteria compares the ZIP code of an opportunity to the ZIP codes that constitute the region that the agent covers.

- c In the Values list, add a new record for each ZIP code included in this assignment rule using values from the following table.

Score	Zip Code
Leave empty	94101
Leave empty	94102
Leave empty	94103

Siebel CRM displays the Values list below the Criteria list. In this list you specify values or a range of values for a given criterion. Siebel CRM uses these criteria values for comparison. In the NREC example, the ZIP codes that are in a particular territory are the values for the assignment criteria.

- 3 Add positions to the assignment rule:
 - a Click the Position Candidates view tab.
 - b In the Position Candidates list, click New.
 - c In the Add Positions dialog box, in the Find window, choose Position.
 - d In the Starting With window, enter NREC Sales Agent, click Go, and then click OK.
 - e Repeat [Step c](#) for the NREC Partner Agent position.

Each rule must include a list of positions, employees, or organizations that specifies the candidates for the assignment. In the NREC example, the position determines the assignment. Each assignment rule is associated with a position at NREC or at the partner real estate agency. For example, the positions at the San Francisco Real Estate Agency are associated to the assignment rule for the Peninsula and East Bay territories.

- To add rules for the Peninsula and East Bay territories, repeat [Step 1](#) through [Step 3](#) for each of the following assignment rules.

Assignment Rule Name	Zip Code
Peninsula	Add the following ZIP codes:
	■ 94301
	■ 94401
	■ 94010
East Bay	Add the following ZIP codes:
	■ 94701
	■ 94602
	■ 94580

- Navigate to the Assignment Rules List, and then click Release.

This step instructs Assignment Manager to use these rules.

NOTE: This step releases all assignment rules. Do not release assignment rules while associated server tasks are running.

- Activate Assignment Manager rules.

Process of Automating a Business Process

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To automate a business process, perform the following tasks:

- [“Examining Business Requirements for Automating a Business Process”](#) on page 147
- [“Configuring Siebel Communications Server”](#) on page 147
- [“Creating an Email Template”](#) on page 147
- [“Creating and Deploying a Workflow Process”](#) on page 149
- [“Creating a Workflow Policy”](#) on page 151
- [“Running the Generate Triggers Server Component”](#) on page 153
- [“Defining and Starting a New Workflow Monitor Agent”](#) on page 154

Examining Business Requirements for Automating a Business Process

This task is a step in [“Process of Automating a Business Process” on page 146](#).

In the NREC example, if an opportunity is assigned to an agent, then Siebel CRM must send an email notification to a partner real estate agent. To implement this functionality, NREC uses *Siebel Workflow*, which is a customizable business application that allows you to manage and enforce business processes, such as meeting response time objectives, creating review policies, or monitoring service requests or opportunities over time. Siebel Workflow can model the same business processes that an organization uses in their sales, marketing, and service departments. You can use Siebel Workflow to automatically enforce business policies and procedures. In the NREC example, Siebel Workflow uses the following objects:

- **Workflow process.** An object type that defines the steps that are required to send the email.
- **Workflow policy.** An object type that starts the workflow process if the trigger detects that an opportunity is assigned.

This topic describes an overview of how to create a workflow process. For more information, see *Siebel Business Process Framework: Workflow Guide*.

Configuring Siebel Communications Server

This task is a step in [“Process of Automating a Business Process” on page 146](#).

To use Siebel Workflow to send an email message, you must configure the email communications application and the Siebel Communications Server.

To configure Siebel Communications Server

- Set up the computer you use for the Siebel Server to interface with the different messaging applications that Siebel Communications Server supports.
- Configure and administer the Siebel Communications Server components you require, including the Communications Outbound Manager server component.
- Configure communications drivers and profiles that the Siebel Communications Server uses.
- Configure recipient groups for your communications, according to Siebel business objects and business components.

For more information, see *Siebel Communications Server Administration Guide*.

Creating an Email Template

This task is a step in [“Process of Automating a Business Process” on page 146](#).

To send an email message from a Siebel application, Siebel CRM uses the following types of email templates:

- **Simple.** Performs simple commands and sends replies in a Siebel eMail Response. Examples of simple commands include Send Email, Send Fax, Send Wireless Message, Send Page, and so forth.
- **Advanced.** Performs an outbound communication request, such as the request that a workflow process starts.

To create an email template

- 1 Log in to the Siebel client as an administrator, navigate to the Administration - Communications screen, and then the All Templates view.

For more information, see [“Logging In as the Siebel Administrator” on page 185](#).

- 2 In the Templates list, enter a new record using values from the following table.

Field	Value
Name	NREC Email template
Channel Type	Email

- 3 Click the Advanced tab.

- 4 In the Advanced form, click New, and then enter a new record using values from the following table.

Field	Value	Description
Delivery Profile	Default SMTP Profile	Created for the communication drivers for the SMTP/POP3 servers for an email template. The profile must exist before you create the template. To define the communications sender, use the From Address override parameter in this profile.
Recipient Group	Opportunity Sales Team Members	The recipient group you specify here only determines which substitution fields are available for the template text. Siebel CRM determines the actual recipients in the context in which it uses the template.
Subject	New Opportunity in Queue	Default subject line for the email. The subject line can also include substitution fields.
Public	Contains a check mark	Determines if the template is available to all users.
Template Text	Dear [Full Name], A new opportunity is assigned to you. Thank you, NREC	Text that Siebel CRM displays in the body of the template. The term in brackets is a substitution.

For more information, see *Siebel Communications Server Administration Guide*.

Creating and Deploying a Workflow Process

This task is a step in [“Process of Automating a Business Process”](#) on page 146.

A workflow process defines the series of actions that must occur in the workflow. It includes one or more process steps, such as a start step, decision point, business service step, subprocess step, and so forth. The NREC example uses the following steps:

- **Start.** Each workflow process must begin with a start step.
- **Send Email business service.** A predefined business service that sends an email.
- **End.** Each workflow process must end with an End step.

To create and deploy a workflow process

- 1 In Siebel Tools, make sure you complete the following setup work:
 - Display the Workflow Process object type and all child object types of the Workflow Process.

- Display the WF/Task Editor Toolbar.

For more information, see [“Setting Up Siebel Tools” on page 60.](#)

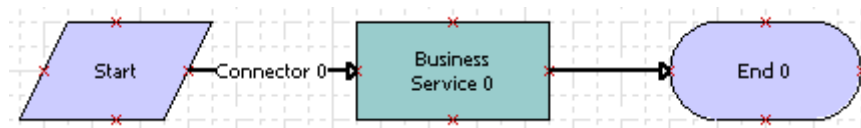
- 2 In the Object Explorer, click Workflow Process.
- 3 In the Workflow Processes list, add a new record using values from the following table.

Property	Value
Process Name	NREC New Opportunity Notification
Business Object	Opportunity
Project	NREC Configuration

- 4 In the Workflow Processes list, right-click the NREC New Opportunity Notification workflow process, and then choose the Edit Workflow Process menu item.

Siebel Tools displays the Workflow Process palette and workspace.

- 5 Drag steps and connectors from the palette to the canvas until your workflow resembles the workflow in the following diagram.



NOTE: A connector end point that is colored white is not connected correctly to a step. A red end point indicates that the connector is correctly connected. Make sure every end of every connector in the workflow process is colored red.

- 6 Right-click the business service step, and then choose the View Properties Window menu item.
- 7 In the Properties window, set the properties using values from the following table.

Property	Value
Name	Send Email
Business Service	Outbound Communications Manager
Business Service Method	CreateRequest

Note that you must choose the business service before you can choose the business service method.

- 8 With the Send Email business service step still chosen, right-click anywhere in the Multi Value Property Window, and then choose the New Record menu item.

- 9 Define the record using values from the following table.

Input Argument	Type	Value	Property Name
PackageNameList	Literal	NREC Email Message	Not applicable

For more information, see *Siebel Business Process Framework: Workflow Guide*.

- 10 Repeat [Step 8](#) and [Step 9](#) for each input argument listed in the following table.

Input Argument	Type	Value	Property Name
RecipientGroup	Literal	Opportunity Sales Team	Not applicable
RequestName	Literal	NREC Request	Not applicable
SourceIdList	Process Property	Not applicable	Object Id

- 11 Choose the File menu, and then the Save menu item.

- 12 In the WF/Task Editor Toolbar, click the Publish/Activate icon.

In the Workflow Processes list, Siebel Tools changes the Status property for the workflow to Completed.

- 13 Deploy the workflow process:

- a Log in to the Siebel client as an administrator, navigate to the Administration-Business Process screen, and then the Workflow Deployment view.

For more information, see [“Logging In as the Siebel Administrator” on page 185](#).

- b In the Repository Workflow Processes list, click Query.

- c In the Name field, enter the following text, and then click Go:

NREC New Opportuni ty Noti fi cati on

- d Click Activate.

- e Click the Active Workflow Process tab.

- f In the Active Workflow Processes list, click Query.

- g In the Name field, enter the following text, and then click Go:

NREC New Opportuni ty Noti fi cati on

Siebel CRM must display the record with a Deployment Status of Active.

Creating a Workflow Policy

This task is a step in [“Process of Automating a Business Process” on page 146](#).

In this topic you create a workflow policy that calls the workflow process. A workflow policy includes a condition and an action. In this example, the condition is met when Assignment Manager assigns an opportunity. The action calls the workflow process you defined in [“Creating and Deploying a Workflow Process” on page 149](#).

To create a workflow policy

1 Create a workflow policy action:

- a** Navigate to the Administration - Business Process screen, and then the Actions view.
- b** In the Actions list, add a new record using values from the following table.

Field	Value	Description
Name	NREC Opportunity Notification	Enter any name that is meaningful to you.
Program	Run Workflow Process	Choose this value to specify that the policy calls a workflow process.
Workflow Object	Opportunity	Choose this value to specify the object that is associated with the workflow policy.

- c** In the Arguments list, click New, and then add a new record using values from the following table.

Field	Value	Description
Argument	ProcessName	Because you specified Run Workflow Process in Step b , ProcessName is the only item you can choose.
Value	NREC New Opportunity Notification	This value is the name of the workflow process you created in “Creating and Deploying a Workflow Process” on page 149 .

2 Create a workflow policy condition:

- a** Click the Policies link.
- b** In the Policies list, click New, and then enter a new record using values from the following table.

Field	Value	Description
Name	NREC Opportunity Notification	You can use any name that is meaningful to you.

Field	Value	Description
Workflow Object	Opportunity	Choose this value because the action is in accordance with changes to an Account field.
Policy Group	NREC	<p>Because this group does not currently exist, do the following work to create it:</p> <ul style="list-style-type: none"> ■ Click the select button in the Group field. ■ In the Workflow Groups dialog box, click New. ■ Specify NREC in the Name field. ■ Click Save and then click OK.

- c In the Conditions list, click New, and then enter the condition using values from the following table.

Condition Field	Operation
Position ID	IS ADDED

- d In the Actions list, enter the record using values from the following table.

Field	Value	Description
Action	NREC Opportunity Notification	This is the name of the Workflow Policy Action you created in Step 1 .
Sequence	10	Sequence of the actions.

Running the Generate Triggers Server Component

This task is a step in [“Process of Automating a Business Process”](#) on page 146.

To create a database trigger, you use the Generate Trigger (GenTrig) server component on the Siebel Server. Siebel Workflow uses a database trigger to identify the records that might match a policy condition. You must run the Generate Triggers server component any time you create, modify, or delete a workflow policy.

You can run the Generate Triggers server component from the Server Manager or from the command line interface. This topic describes how to use the Server Manager. For information about using the command line interface, see *Siebel System Administration Guide*.

To run the Generate Triggers server component

- 1 Navigate to the Administration - Server Management screen, and then the Jobs view.
- 2 In the Jobs list, click New.
- 3 In the Component/Job field, click the select button.

- 4 In the Components/Jobs dialog box, in the Find window, choose Name.
- 5 In the Starting With window, enter the following text, and then click Go:
Generate Triggers
- 6 In the Job Parameters list, create a new parameter for each row in the following table.

Name	Value	Description
Trigger File Name	trigger.sql	This file name identifies the SQL script that the Siebel Server must run. The server creates this file in the <i>siebsrvr_root</i> directory.
EXEC	TRUE	If you set this parameter to TRUE, then Siebel CRM automatically runs the SQL script.
Table Owner Password	Your password	If you use an Oracle database or an IBM DB2 database, then enter the password for the table owner.

- 7 In the Jobs list, click Submit Job.
- 8 To view updates that the Siebel Server displays in the Status field, click Query, and then click Go. Upon completion, the Siebel Server displays Success or Error in the Status field.
- 9 To view log details, do the following:
 - a Click the Tasks link.
 - b In the Tasks list, locate the job you started in [Step 7](#).
 - c Examine the records in the Log list.

Defining and Starting a New Workflow Monitor Agent

This task is a step in [“Process of Automating a Business Process”](#) on page 146.

In this topic, you define and start a separate Workflow Monitor Agent task for Assignment Manager, and another task for Workflow policies.

To define and start a new Workflow Monitor Agent

- 1 Navigate to Administration - Server Configuration screen, and then the Enterprises view.
- 2 Click the Component Definitions tab.

- 3 In the Component Definitions list, click New, and then create a new record using values from the following table.

Field	Value
Component	NREC WorkMon
Component Type	Workflow Monitor Agent
Component Group	NREC Workflow Component Group
Alias	NRECWorkMon

- 4 Activate the Workflow Monitor Agent server component:
 - a In the Component Parameters list, query the Parameter field for Group Name, and then set it using values from the following table.

Parameter	Value
Group Name	Enter the name of the workflow policy group for which this component handles requests.

- b In the Component Parameters list, click Advanced, and then set the value for the Default Tasks parameter using values from the following table.

Parameter	Value
Default Tasks	1

This step sets the component to start when the Siebel server starts, and to stop when the Siebel server shuts down.

- c In the Component Definitions list, click Menu, and then choose the Enable Component Definition menu item.
The Siebel Server changes the state to Active.
 - d Restart the Siebel Server.

Filtering Records According to a User Profile

This task is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

Siebel Personalization is a solution that allows you to filter content in an applet that Siebel CRM displays for a specific user according to the requirements of a preference or profile of the user. This topic describes an overview of how to configure Siebel Personalization. For more information, see *Siebel Personalization Administration Guide*.

To create and test a personalization rule

- 1 Examine the business requirements to filter records according to a user profile.

NREC must filter the list of houses that are for sale according to the ZIP code in the profile of the partner agent. If the agent views the list of houses for sale, then Siebel CRM must display only the houses for sale that are located in the same ZIP codes that constitute the sales region to which the agent is assigned. In this example, filtering must occur in the Product Form Applet and the Product List Applet.

- 2 Create a personalization rule set:
 - a Log in to the Siebel client as an administrator, navigate to the Administration - Personalization screen, and then the Rule Sets view.
For more information, see [“Logging In as the Siebel Administrator” on page 185](#).
 - b In the Rule Sets list, click New, and then add a new record using values from the following table.

Field	Value
Name	NREC House Filter
Active	Contains a check mark.

A *rule set* is a collection of the rules that define how Siebel CRM displays content. For the NREC example, you create one rule set to contain these rules.

- 3 Associate the personalization rule set with an applet:
 - a Click the Applets link.
 - b In the Applets list, click New, and then create a new record using values from the following table.

Field	Value
Name	Product List Applet

- c In the Rule Sets list, add a new record using values from the following table.

Field	Value	Description
Name	NREC House Filter	This is the rule set you created in Step 2 .
Sequence	1	Choose the order in which Siebel CRM must evaluate the rule sets.

Field	Value	Description
Start Date	Leave empty	The date when Siebel CRM must start to evaluate the conditional expression. If Start Date is empty, then Siebel CRM evaluates the conditional expression continuously or until the End Date occurs.
End Date	Leave empty	The date when Siebel CRM must stop evaluating the conditional expression. If End Date is empty, then Siebel CRM evaluates the conditional expression continuously after the Start Date occurs. If Start and End Date are empty, then Siebel CRM always evaluates the conditional expression.

- d To associate the Product Form Applet with the NREC House Filter, repeat [Step a](#) through [Step c](#) with the following differences:
 - In the Applets list, create the Product Form Applet instead of the Product List Applet.

4 Create a personalization rule:

- a Click the Rule Sets link.
- b In the Rule Sets list, choose the NREC House Filter rule set that you created in [Step 2](#).
- c In the Rules list, click New, and then enter a new record using values from the following table.

Field	Value	Description
Name	NREC House Filter	Enter a name that describes the purpose of the rule.
Rule Type	Expressions	Set this field to instruct Siebel CRM to evaluate the expression.
Sequence	1	Enter a number to set the order in which Siebel CRM evaluates the rules.
Active	Contains a check mark.	Make sure the Active field contains a check mark.

- d In the More Info form, add information using values from the following table.

Field	Value	Description
Include Expression	[ZIP Code] = GetProfileAttr('Me.Org.Postal Code')	Enter the expression that Siebel CRM must evaluate. For more information, see "About Personalization Rules" on page 158 .

5 Test the personalization rule:

- a Click the Test link.

- b** In the Test Mode form, enter a Primary User Login and Primary User Password.
This is the user name and password of the user you must use to test the personalization rules. For example, to test the NREC personalization rule, use the SADMIN user and SADMIN password.
- c** In the Test Application field, enter the command string for the Siebel Mobile Web Client.
For more information, see [“Entering the Command String to Test a Personalization Rule” on page 159](#).
- d** Click Load.
Siebel CRM loads the persistent user profile attributes of the primary user in the Primary User Attributes list with an *Me* prefix. It uses the following values in the Source field:
 - Person for a person attribute
 - Organization for an organization attribute
- e** In the Primary User Attribute list, query the Name field for Me.Org.Postal Code, and then enter a value in the Value field.
For example, enter 94121.
- f** In the Test Mode form, click Test.
Siebel CRM opens a new instance of the Siebel Partner Portal application. If you use the Siebel Web Client to perform the test, then Siebel CRM displays the Test Mode dialog box that includes instructions about how to start a new instance of the application.
- g** Navigate to the House Detail view, and then enter several house records.
Make sure you use several different ZIP codes for the different records you enter. For more information, see [“Process of Configuring the House Detail View” on page 75](#).
- h** Make sure Siebel CRM only displays records that include a ZIP code value of 94121.

About Personalization Rules

Siebel CRM uses one of the following personalization rules to control how it filters content that it displays to the user:

- Business Service
- Expression
- Invoke Method

For the NREC example, you write an Expression personalization rule.

To define the parameters that control this content, an expression rule uses Siebel Query Language. For this example, the following expression specifies that when Siebel CRM displays records from the Internal Product business component, it must display only those records whose value in the ZIP Code field matches the value in the Postal Code for the user:

```
[Zip Code] = GetProfileAttr('Me.Org.Postal Code')
```

Entering the Command String to Test a Personalization Rule

This topic describes how to enter the command string to test a personalization rule.

to enter the command string to test a personalization rule

- 1 Navigate to the Administration - Personalization screen, and then the Test view.
- 2 In the Test Application field, enter the following command:

```
C: \ORACLE_HOME\BI\si ebel . exe /c "C: \ORACLE_HOME\bi n\ENU\confi g_fi l e. cfg" /d  
data_source
```

where:

- *ORACLE_HOME* is the root directly of where you installed the Siebel client
- *config_file* is the application configuration file. for example, uagent.cfg for Siebel Call Center or siebel.cfg for Siebel Sales.
- *data_source* is the database to which to connect. You must use one of the following values:
 - Local
 - Sample
 - ServerDataSrc

For example:

```
C: \Si ebel \8. 2\cl i ent\BI\si ebel . exe /c "C: \Si ebel \8. 2\cl i ent\BI\ENU\scw. cfg" /d  
Sampl e
```

You can also enter the URL for a Siebel Web Client. For example, you can enter the following URL:

```
http: //computer_name/eChannel
```


11 Configuring Siebel Remote

This chapter describes how to configure Siebel Remote. It includes the following topics:

- [About Siebel Remote on page 161](#)
- [Process of Configuring Siebel Remote on page 162](#)

About Siebel Remote

Oracle's Siebel Remote allows a Siebel Mobile Web Client that typically operates remotely in a disconnected mode to perform *synchronization*, which is the process of a laptop, or other remote computer that is periodically disconnected from the network, connecting to a Siebel Server and exchanging updated data and files. Siebel Remote allows field personnel to share current information with members of virtual teams of other mobile and connected users across the organization.

A remote client can download a portion of the Siebel database and the Siebel File System to a laptop. The User can access their data locally without a connection to the Database Server, Siebel Server, or file system.

Figure 22 illustrates the main parts of Siebel Remote.

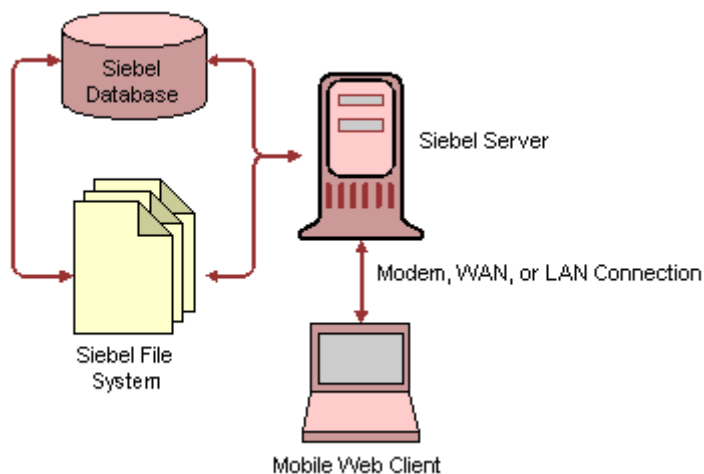


Figure 22. Main Parts of Siebel Remote

Between synchronization sessions, the Siebel Remote server prepares transactions that mobile and connected users apply to CRM data. Siebel Server components write the transactions to a separate directory for each mobile user. During the next synchronization session, Siebel CRM downloads transactions and items from the file system. Items from the file system include updated, published, or requested marketing literature, correspondence templates, and other types of file attachments.

Process of Configuring Siebel Remote

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To configure Siebel Remote, perform the following tasks:

- 1 [“Setting Up the Siebel Remote Server”](#) on page 162
- 2 [“Setting Up a New Siebel Remote User”](#) on page 162
- 3 [“Registering a Siebel Mobile Web Client”](#) on page 163
- 4 [“Extracting, Initializing, and Synchronizing the Remote Database”](#) on page 164

For details about content in this topic, see *Siebel Remote and Replication Manager Administration Guide*.

Setting Up the Siebel Remote Server

This task is a step in [“Process of Configuring Siebel Remote”](#) on page 162.

The Siebel Remote Server does the following:

- Runs Siebel Remote server components, such as the transaction processor, transaction router, and other server components.
- Manages synchronization sessions with Siebel Mobile Web Clients.
- Provides an interim storage area for data that Siebel Remote requires to synchronize mobile databases with the Siebel database server.

To set up the Siebel Remote Server

- 1 Set system preferences for Siebel Remote.
- 2 Disable local access to all views.
- 3 Start server components for Siebel Remote.
- 4 Generate a new database template.

Setting Up a New Siebel Remote User

This task is a step in [“Process of Configuring Siebel Remote”](#) on page 162.

Setting up a Siebel Mobile Web Client involves certain tasks for the Siebel Remote server and the Siebel Mobile Web Client. You must repeat each of these steps for each mobile client.

To set up a new Siebel Remote user

- 1 Create a Siebel Mobile Web Client user account and privileges.

Because NREC does not use authentication for the Siebel Mobile Web Client, you do not create this account or password for the NREC example. Not using accounts or passwords is the easiest and most popular configuration. It is the default configuration.

- 2 Install and configure the hardware and software for the Siebel Mobile Web Client.

Depending on your implementation, this work might include the following items:

- Install disk drives, memory cards, and operating system software.
- Configuring user information.
- Install the necessary hardware and software on the Siebel Mobile Web Client so the Siebel Mobile Web Client can exchange files with the Siebel Remote server. This work might include choosing communication settings and installing networking cards, modems, and software.

For more information, see *Siebel Installation Guide for Microsoft Windows*.

- 3 Set Siebel Remote preferences.

To identify the location of the Siebel Server, the Siebel Remote client reads parameters in the siebel.cfg configuration file. Before you use Siebel Remote, you must set the values for the configuration parameters. The Siebel installation utility creates a siebel.cfg file in the client BIN directory with default values for each configuration parameter.

NREC uses the default settings for the Siebel Remote parameters in the client configuration file that are established during installation. For the NREC example, you do not set any remote preferences.

Registering a Siebel Mobile Web Client

This task is a step in [“Process of Configuring Siebel Remote” on page 162](#).

In this topic, you register a Siebel Mobile Web Client.

To register a Siebel Mobile Web Client and extract and initialize the database

- 1 Make sure you set up Dana Castro as an employee.

For more information, see [“Creating Positions and Position Skills” on page 44](#).

- 2 Navigate to the Administration - Application screen, and then the Responsibilities view.
- 3 In the Views list of the Real Estate Agent responsibility, add the Client Status view.
- 4 In the Responsibilities list, click Query, enter the following text in the Responsibility field, and then click Go.

Real Estate Agent

- 5 In the Views list, click Add, add then add a new record using values from the following table.

Field	Value
View Name	Client Status

This view displays conflict information after synchronization. It allows the Siebel Mobile Web Client to resolve data merge conflicts on the local database. For more information about responsibilities, “[Creating Responsibilities](#)” on page 38 and *Siebel Applications Administration Guide*.

- 6 To register the remote user, do [Step 3 on page 52](#) but with the following differences:

Field	Description
Mobile Client	DCASTRO
User ID	DCASTRO
Routing Model	Sales Representative

- 7 Add the mobile user to the responsibility:
 - a Navigate to the Administration - Application screen, and then the Responsibilities view.
 - b In the Responsibilities list, click Query, enter the following text in the Responsibility field, and then click Go:

Sales Representative Routing Mode

 Siebel CRM uses this responsibility for the data routing model you chose in [Step 6](#).
 - c In the Users list, click Add.
 - d In the Add Users dialog box, choose the record for Dana Castro, and then click OK.

Extracting, Initializing, and Synchronizing the Remote Database

This task is a step in “[Process of Configuring Siebel Remote](#)” on page 162.

In this topic, you extract, initialize, and synchronize the remote database.

To extract, initialize, and synchronize the remote database

- 1 To extract the database, do [Step 7 on page 55](#) but use DCASTRO for the Client Name.
- 2 Initialize the database on the Siebel Mobile Web Client:
 - a On the remote computer, in the Siebel program group, click Siebel Remote.

- b In the Siebel Remote Parameters dialog box, enter information using values from the following table.

Field	Entry
Client Name	DCASTRO
User Name	DCASTRO
Password	DCASTRO

- c Click Continue.
 - d Monitor the process for errors.
- 3 Synchronize the Siebel Mobile Web Client:
- a Start a Siebel application on the remote computer.
 - b Choose the File menu, and then the Synchronize Database menu item.
 - c Choose the synchronization options.
 - d Click Synchronize.

Siebel Remote does the following work:

- Connects
- Validates Siebel Mobile Web Client
- Checks for the correct version
- Checks for database extract
- Retrieves transactions and file attachments
- Sends transactions and file attachments
- Applies changes to the local database
- Disconnects
- Applies changes to the server database
- Performs clean up work

12 Setting Up the Test Environment

This chapter describes how to set up the test environment. It includes the following topics:

- [Process of Moving Data to the Test Environment on page 167](#)
- [Process of Loading CRM Data in the Test Environment on page 170](#)
- [Administering the Test Environment on page 177](#)

Process of Moving Data to the Test Environment

This process is a step in [“Roadmap for Developing a Siebel Application” on page 31](#).

To move repository data to the test environment, perform the following tasks:

- 1 [“Moving Repository Data” on page 167](#)
- 2 [“Moving Web Templates, Image Files, and Cascading Style Sheets” on page 169](#)
- 3 [“Moving Nonrepository Data” on page 170](#)

Moving Repository Data

This task is a step in [“Process of Moving Data to the Test Environment” on page 167](#).

This topic describes how to move repository data, such as applets, views, and business components. The tool you use to move data depends on the type of configuration changes you must move:

- If you must move a database schema change, then use the Repository Migration Configuration Utility. For example, if you add columns to a base table, which you did do in [Chapter 6, “Configuring the House and Opportunity Entities.”](#)
- If you must move only user interface and business object definitions, then you can use the Export and Import Repository Utilities.

For more information about managing repositories, see *Going Live with Siebel Business Applications*.

To move repository data

- 1 Prepare to move repository data:
 - a Make sure each developer checks in their projects from their local database to the server database.
 - b Instruct your developers not to make any more changes after you begin to move repository data. These changes might be lost.

- c Stop all server tasks on the target environment.
 - d Disconnect database access until the move is complete.
- 2 To start the Database Server Configuration Utility, in Microsoft Windows, choose the Start menu, Programs, Siebel Enterprise Server *version_number*, and then the Configure DB Server menu item.
 - 3 In the Gateway Server Address dialog box, specify your Siebel Gateway Name Server Address and Enterprise Server Name, and then click Next.
 - 4 In the Siebel Server Directory dialog box, accept the default value or click Browse, and then click Next.
 - 5 In the Siebel Database Server Directory dialog box, accept the default value or click Browse, and then click Next.
 - 6 In the RDBMS Platform dialog box, choose the platform for your environment, and then click Next.
 - 7 In the Siebel Database Operations dialog box, choose Migrate Repository from the list of operations, and then click Next.

For more information, see ["About the Repository Migration Configuration Utility"](#) on page 169.

- 8 Proceed through each dialog box using values from the following table, and then clicking Next.

Dialog Box	Description
ODBC Data Source Name	Enter the name of the ODBC data source.
Database User Name	Enter the source database user name and source database password.
Database Table Owner	Enter the source database table owner and source table owner password.
Source Database Repository Name	Enter the database repository name and the target database repository name.
Target RDBMS Platform	Choose the target RDBMS platform.
Target Database Encoding	Choose whether the target database is Unicode.
Target Database ODBC Datasource	Enter the target database ODBC datasource.
Target Database User Name	Enter the target database user name and password.
Target Database Table Owner	Enter the target database table owner and table owner password.
Index Table Space Name	Enter a value only if you indicated that the target database is IBM DB2. Enter the index table space name and 4 KB Table Space Name.
16-KB Table Space Name	Enter a value only if you indicated that the target database is IBM DB2. Enter the 16 KB table space name and 32 KB table space name.

Dialog Box	Description
Index Table Space Name	Enter a value only if you indicated that the target database is Oracle. There are no 4 KB, 16 KB, or 32 KB table spaces in Oracle. For Microsoft SQL server, there are no dialog boxes for table spaces.
Configuration Parameter Review	Summary screen that displays a list of your choices. To accept this configuration, click Finish.

About the Repository Migration Configuration Utility

The Repository Migration Configuration Utility is a wizard that prompts you for the required information as you proceed through a succession of dialog boxes. You start the Repository Migration Configuration Utility from the Siebel Database Configuration Utility. When you complete the configuration session, you can use the command line to perform the move immediately or at a later time. The Repository Migration Configuration Utility does the following work:

- 1 Exports repository data from the source environment
- 2 Imports the object definitions
- 3 Updates the physical database schema of the target database.

As a result of this work, the database schema and object definitions in the target environment matches the database schema and object definitions in the source environment.

Moving Web Templates, Image Files, and Cascading Style Sheets

This task is a step in [“Process of Moving Data to the Test Environment” on page 167](#).

The Siebel Server installation includes a set of Web template files and other related files, such as image files and cascading style sheets. If you make changes to any of these files in your source environment, then you must copy the modified files to the target environment.

To move Web templates, image files, and cascading style sheets

- Copy any new or modified files of the following types from the development server to the test server. *ORACLE_HOME* in the following table indicates the root directory of where you installed the Siebel Server.

Files	File Location
Cascading style sheets	<i>ORACLE_HOME</i> \webmaster\files\language_code
Images	<i>ORACLE_HOME</i> \webmaster\images\language_code
Web templates	<i>ORACLE_HOME</i> \WEBTEMPL

Moving Nonrepository Data

This task is a step in [“Process of Moving Data to the Test Environment”](#) on page 167.

Moving application run-time customization data from one Siebel application environment to another Siebel application environment is a common requirement when you go live to a new development, test, or production environment. The Application Deployment Manager (ADM) automates moving this data. For example, ADM can move customization data from a development environment to a testing environment. ADM eliminates the need to transfer this data manually. Modified views, responsibilities, and assignment rules are examples of enterprise customization data. ADM does not change the data that it moves.

You can use ADM to move the following data types:

- Access Groups
- Assignment Rule
- Expense Types
- List of Value (LOV)
- Product Feature
- Product Line
- Public Predefined Query (PDQs)
- Responsibility
- State Model
- User List
- View

These data types are included in the predefined Siebel application. You can use Siebel Tools to add more data types for use with ADM.

If you use ADM, then you perform most of the administrative tasks to move data in the Application Deployment Manager screen in the Siebel client. These tasks require the Siebel administrator responsibility. For more information, see [“Creating Responsibilities”](#) on page 38.

The ADM setup process in the ADM user interface creates a template in which you can move one data type on a regular basis, if required. The deployment project is the fundamental structure of this template. The deployment project includes one or more data types that you can move.

For more information, see *Siebel Application Deployment Manager Guide*.

Process of Loading CRM Data in the Test Environment

This process is a step in [“Roadmap for Developing a Siebel Application”](#) on page 31.

To load CRM data in the test environment, perform the following tasks:

- 1 "Loading the Interface Tables" on page 171
- 2 "Adding the EIM Configuration File" on page 172
- 3 "Disabling Transaction Logging" on page 175
- 4 "Running the Enterprise Integration Manager" on page 175

Overview of Using Siebel Enterprise Integration Manager

This topic describes how to use Siebel Enterprise Integration Manager (EIM) to load data from your legacy application to a Siebel database. Generally, you use EIM to exchange data between Siebel database tables and other data sources. This exchange of data can include bulk imports, exports, merges, and deletes. Because the NREC example only requires you to load data to a Siebel database, this topic only describes how to perform a bulk import. For more information, see *Siebel Enterprise Integration Manager Administration Guide*.

This topic describes only one import example, which is how to load account data. You must load the EIM_ACCOUNT interface table, prepare the NRECimp.ifb configuration file, and run the EIM import process.

About Interface Tables

An *interface table* is an intermediate database table that resides between the Siebel database and another database.

Figure 23 illustrates the flow of information when you use EIM.

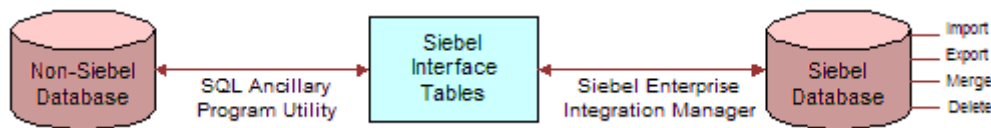


Figure 23. Flow Between a Non Siebel Database and a Siebel Database

Loading the Interface Tables

This task is a step in "Process of Loading CRM Data in the Test Environment" on page 170.

In this topic, you load the interface tables.

To load the interface tables

- 1 Open Siebel Tools.
- 2 Determine which columns are required:
 - a In the Object Explorer, click EIM Interface Table.

If the EIM Interface Table object type is not visible, then display it. For more information, see "Setting Up Siebel Tools" on page 60.

- b** In the EIM Tables list, locate the interface table you must load.
For the NREC example, locate the EIM_ACCOUNT table.
- c** In the Object Explorer, expand the EIM Interface Table tree, and then click EIM Interface Table Column.
- d** In the Interface Table Columns list, enter a query that locates all records that contain a check mark in the Required property.

For the EIM_ACCOUNT table, the following interface table columns are required:

- ❑ IF_ROW_BATCH_NUM
- ❑ IF_ROW_STAT
- ❑ PARTY_TYPE_CD
- ❑ PARTY_UID
- ❑ ROW_ID

For more information, see *Siebel Enterprise Integration Manager Administration Guide*.

3 Load the interface tables.

To load legacy data to Siebel interface tables, you use a database software utility that is specific to your database. For example, if you use DB2, then use one of the load utilities available for DB2. For more information, see the documentation for your database software.

Make sure you load the required columns you identified in [Step 2](#).

Some columns in the interface tables are part of a user key sequence. The *user key sequence* is the set of user key columns whose values uniquely identify a row. To prevent importing duplicate rows, as EIM loads each row, it checks whether the interface table already contains the user key sequence for that row.

For information on the following topics, see *Siebel Enterprise Integration Manager Administration Guide*:

- Specific data and file attachments that EIM can process
- The names of the interface tables
- The target base tables mapped to the interface tables
- Any secondary tables associated with the target tables

Adding the EIM Configuration File

This task is a step in [“Process of Loading CRM Data in the Test Environment”](#) on page 170.

EIM reads a configuration file that specifies which process it must perform and defines the parameters EIM uses to run the process. This file is an ASCII text file that contains an IFB extension in the file name. Before you start an EIM process, you must edit the contents of this EIM configuration file. If you do not specify a configuration file, then EIM uses the default.ifb file.

In this example, EIM sets the process locale as specified in the configuration file at start up. For information on the following topics, see *Siebel Enterprise Integration Manager Administration Guide*:

- Other process and header parameters
- Alternative sources from which EIM accepts parameter values
- How to define multiple processes in the configuration file

To add the EIM configuration file

- 1 Using a text editor, such as Notepad, create a file named NRECimp.ifb, and then save it to the following directory on the Siebel Server:

```
siebsrvr_root\admin
```

In a Unicode environment, you must save the EIM configuration file as a Unicode text file.

- 2 Copy the text from the example to the NRECimp.ifb file you created in [Step 1](#), and then save your work.

For more information, see ["Example Text for the Configuration File" on page 173](#).

Example Text for the Configuration File

You can use the following text for the example NRECimp.ifb file. It includes only the entries that EIM requires to support import operations for accounts:

```
/* These are header parameters. */
[Siebel Interface Manager]
  LOGIN USER = "SADMIN"
  LOGIN PASSWORD = "SADMIN"
  RUN PROCESS = Import Accounts

/* These are generic process parameters. */
[Import Accounts]
  TYPE = IMPORT
  TABLE = EIM_ACCOUNT
  ONLY BASE TABLES = S_ORG_EXT, S_ADDR_ORG, S_PARTY

  ONLY BASE COLUMNS = S_ORG_EXT.NAME, S_ORG_EXT.LOC, S_ADDR_ORG.ADDR,
S_ADDR_ORG.OU_ID, S_ADDR_ORG.CITY, S_ADDR_ORG.STATE, S_ADDR_ORG.ZIPCODE,
S_PARTY.PARTY_UID, S_PARTY.PARTY_TYPE_CD

/* There are no process-specific parameters required in this example. */
```

Header Section Parameters

Table 11 describes the required header section parameters in the NRECimp.ifb file to configure EIM for an import process. The header section specifies global parameters that apply to the process sections defined later in the file.

Table 11. Header Section Parameters of the Configuration File

Parameter	Description
Siebel Interface Manager	The header section must use this reserved name
USERNAME	Database employee login. Use SADMIN for NREC.
PASSWORD	Database password. Use SADMIN for NREC.
PROCESS	Initial or main process section to run. Use Import Accounts for NREC.

Process Section Parameters

Table 12 describes the required process section parameters in the NRECimp.ifb file to configure EIM for an import process. The configuration file must include at least one process section. Note that process section parameters can be one of the following:

- Generic to all EIM processes.
- Specific to a particular EIM process. Each parameter applies only to the particular process you specify. It overrides any corresponding value in the header section.

Table 12. Process Section Parameters of the Configuration File

Parameter	Description
ONLY BASE COLUMNS	For the NREC example, only process the following columns: <ul style="list-style-type: none"> ■ S_ORG_EXT.NAME ■ S_ORG_EXT.LOC ■ S_ADDR_ORG.ADDR ■ S_ADDR_ORG.OU_ID ■ S_ADDR_ORG.CITY ■ S_ADDR_ORG.STATE ■ S_ADDR_ORG.ZIPCODE
ONLY BASE TABLES	For the NREC example, only process the following tables: <ul style="list-style-type: none"> ■ S_ORG_EXT ■ S_ADDR_ORG ■ S_PARTY

Table 12. Process Section Parameters of the Configuration File

Parameter	Description
TABLE	For the NREC example, only process the EIM_ACCOUNT interface table.
TYPE	For the NREC example, only process the following types: <ul style="list-style-type: none"> ■ IMPORT ■ EXPORT ■ DELETE ■ MERGE ■ SHELL

Disabling Transaction Logging

This task is a step in [“Process of Loading CRM Data in the Test Environment”](#) on page 170.

To improve EIM performance, it is recommended that you disable transaction logging before you perform an initial load through EIM.

CAUTION: If there are active mobile clients, then do not disable Docking Transaction Logging. If there are active mobile clients, and if you disable Docking Transaction Logging, then the server database and mobile client databases will be out of synchronization after the import. To resynchronize them, you must extract the database for each mobile client, and then initialize the database on each client. For more information, see [Chapter 11, “Configuring Siebel Remote,”](#) and *Siebel Remote and Replication Manager Administration Guide*.

To disable transaction logging

- 1 Log in to the Siebel client as an administrator.

For more information, see [“Logging In as the Siebel Administrator”](#) on page 185.

- 2 Navigate to the Administration - Siebel Remote screen, and then the Remote System Preferences view.

- 3 Make sure the Enable Transaction Logging option does not contain a check mark.

To disable transaction logging, you can also change the LOG TRANSACTIONS parameter in the EIM configuration file. For more information, see [“Adding the EIM Configuration File”](#) on page 172.

Running the Enterprise Integration Manager

This task is a step in [“Process of Loading CRM Data in the Test Environment”](#) on page 170.

In this topic, you start an import process for Enterprise Integration Manager from the Siebel client. For information about starting the import process from the command line, see *Siebel Enterprise Integration Manager Administration Guide*.

To run the Enterprise Integration Manager

- 1 Navigate to the Administration-Server Management screen, and then the Jobs view.
- 2 Add a new job:
 - a In the Jobs list, click New.
 - b In the Component/Job field, click the select button.
 - c In the Components/Jobs dialog box, in the Find window, choose Name.
 - d In the Starting With window, enter Enterprise Integration Mgr, click Go, and then click OK.
- 3 Add a job parameter:
 - a In the Job Parameters list, click New.
 - b In the Name field, click the select button.
 - c In the Job Parameters dialog box, in the Find window, choose Description.
 - d In the Starting With window, enter Configuration file, click Go, and then click OK.
 - e In the Value field, enter NRECImp. i fb.
- 4 Repeat [Step 3](#) for each of the parameters listed in the following table.

Parameter	Value
Batch Number	Enter 0, which instructs EIM to use the batch number you specify in the EIM configuration file.
Process	Enter Import Accounts. This is the process name you specified in the RUN PROCESS parameter in the EIM configuration file. For more information, see “Adding the EIM Configuration File” on page 172 .

- 5 In the Jobs list, click Submit Job.

CAUTION: If the EIM process is running, then do not stop or pause the task. If you stop an EIM process that is running, and if you must roll back changes, then some steps might not roll back correctly.
- 6 Examine the results of the data import:
 - a In Windows Explorer, navigate to the `si ebsrvr_root\log\server` directory.
 - b Examine the EIM log file for the corresponding task number.

The EIM log file contains status and diagnostic information about the import process. The log file name uses the following format:

`Eim_task#.trc`

where:

□ *task#* is the task number for the EIM process.

To determine the status of each row that EIM attempted to import, you can also examine the IF_ROW_STAT column. For more information, see *Siebel Enterprise Integration Manager Administration Guide*.

Administering the Test Environment

You must administer the company structure and views in the test environment.

To administer the test environment

- 1 Administer the company structure and set up test users.

You must administer the same company structure and set up users in the test environment that you created in [Chapter 3, "Installing Software, Adding Users, and Creating the Company Structure."](#)

- 2 Register the following views in the test environment:

- House Detail View
- House Detail - Appraisals
- House Detail - Renovations

You created these views in the development environment. For more information, see [Chapter 6, "Configuring the House and Opportunity Entities"](#) and ["Registering a New View"](#) on page 186.

- 3 Turn off forecast views that you do not intend to use.

13 Common Software Development Tasks

This chapter describes tasks you can perform in a typical software development effort. It includes the following topics:

- [Using Siebel Tools on page 179](#)
- [Process of Modifying Siebel Objects on page 183](#)

Using Siebel Tools

Siebel Tools is an integrated development environment that you use to create and modify object definitions in the Siebel repository. You can use Siebel Tools to configure a predefined Siebel application without modifying source code or SQL code. This topic only provides a summary. For more usage information, see *Using Siebel Tools*.

Displaying the Object Explorer

The *Object Explorer* is a window in Siebel Tools that displays parent object types and their child object types in a tree hierarchy. A *parent-child relationship* is a hierarchical relationship that defines a relationship between object definitions. For example, if you expand the applet tree, then Siebel Tools displays child object types, such as the following:

- Applet method menu item
- Applet web template
- Control
- Drilldown object
- List
- Tree

A parent-child relationship between object definitions implies that the child object definition is in, or belongs to the parent object definition. It does not imply inheritance between object definitions. The set of properties of a parent object is not related to the set of properties of a child object.

To view object types listed alphabetically in a nonhierarchical structure, you can click the Flat tab in the Object Explorer.

To display the Object Explorer

- 1 In Siebel Tools, choose the View menu, and then the Object Explorer menu item.

Displaying the Object List Editor

The *Object List Editor* is a window that lists the object definitions for the object type chosen in the Object Explorer. Siebel Tools displays this window to the right of the Object Explorer.

To Display the Object List Editor

- Click any object in the Object Explorer.

Displaying the Properties Window

The *Properties window* displays the property values for the object definition chosen in the Object List Editor. It displays the same values that the Object List Editor displays, but it displays them in a smaller window and only for the record currently chosen in the Object List Editor. If you must view objects that contain many properties, then the property window is sometimes more convenient than the Object List Editor.

To display the Properties window

- Choose the View menu, Windows, and then the Properties Window menu item.

Opening a Wizard

Siebel Tools includes wizards that guide you through the task of creating a new object definition. The wizard prompts you for information that is required to define a particular object, such as an applet or a view. When you click Finish, the wizard creates the object definition and any related object definitions according to the choices you make. You can use the Object List Editor to create an object manually, but if a wizard is available, then it is strongly recommended that you use the wizard.

To open a wizard

- 1 Choose the File menu, and then the New Object menu item.
- 2 In the New Object Wizards dialog box, choose an object, and then click OK.

Siebel Tools starts the appropriate wizard.

Opening the Web Layout Editor

The *Web Layout Editor* is an HTML editor that allows you to drag a user interface object, such as an applet or control, and drop it in a placeholder in a Siebel Web template. You use the Web Layout Editor to design the layout of an applet or view.

To open the Web Layout Editor

- In the Object List Editor, right-click one of the following objects, and then choose the Edit Web Layout menu item:
 - Applet
 - Applet Web template
 - View
 - View Web template
 - Web Template

Figure 24 illustrates usage of the Web Layout Editor.

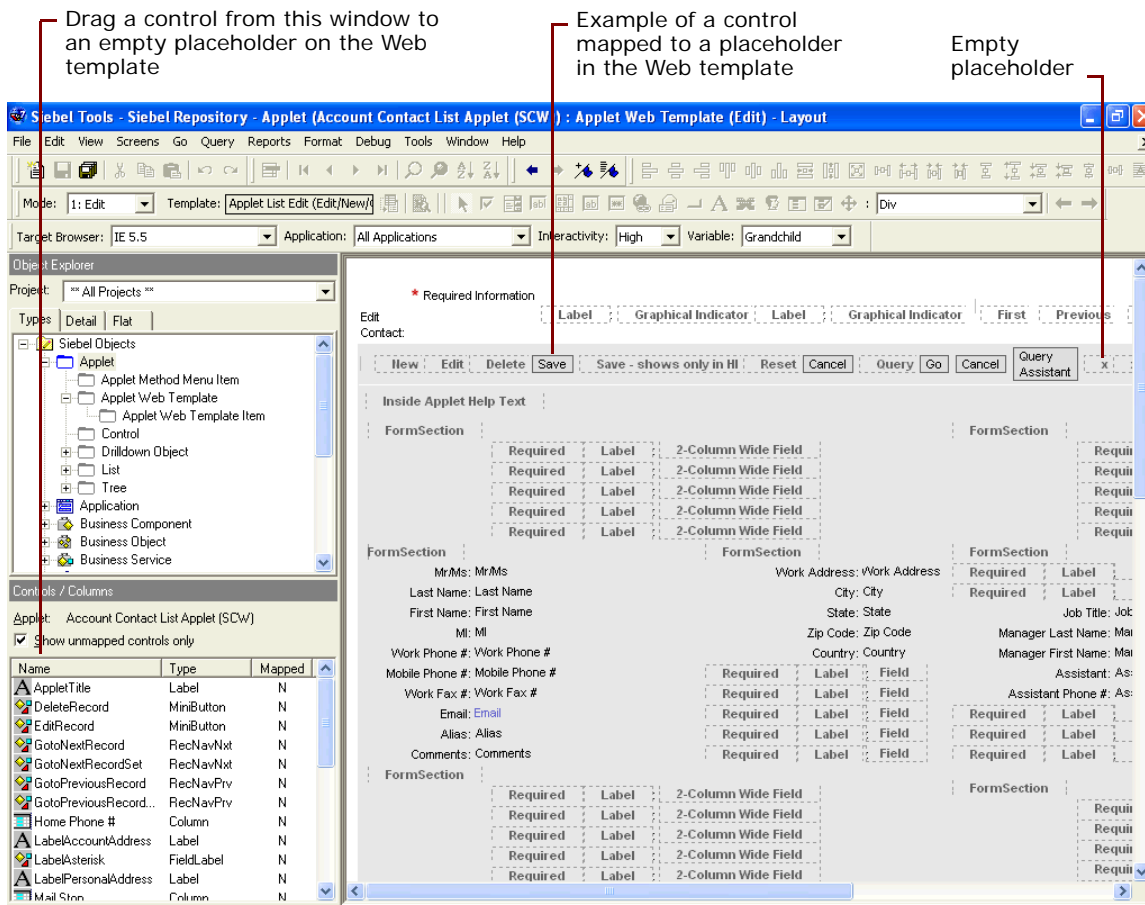


Figure 24. Web Layout Editor for an Applet

About Applets and Applet Web Templates

To render an applet in the Siebel client, Siebel CRM uses configuration information from the following items:

- Object definitions that are stored in the Siebel repository

- Layout and format information contained in an applet Web template

A user interface object, such as a control or list column, is mapped to a placeholder in a Web template. At run time, the Siebel Web Engine does the following:

- Creates each object
- To display the applet in the Siebel client, uses the HTML contained in the Web template
- Places each object in the appropriate location in the Web template
- To enter CRM data that the object displays, retrieves the relevant data from the Siebel database

To modify or create an applet, in Siebel Tools you can use the Object List Editor to define object properties. You can use the Web Layout editor to map controls to placeholders in Web templates and to define object properties. To modify other qualities of a user interface element you use an external cascading style sheet. For example, to change the style of an applet, such as the color scheme. You typically do not modify Web templates themselves.

For more information, see *Configuring Siebel Business Applications*.

Setting Options in Siebel Tools

To customize how Siebel Tools behaves, you can set options in Siebel Tools.

To set options in Siebel Tools

- 1 In Siebel Tools, choose the View menu, and then the Options menu item.
- 2 Choose one of the following tabs, make your adjustments, and then click OK:
 - **Language Settings.** For more information, see [“Setting Language Options” on page 182](#).
 - **Check In/Out.** Specifies the development server database and the local client database that Siebel Tools uses to check in and check out objects.
 - **Web Template Editor.** Specifies the default Web template editor. To use a default editor to open a Web template, you can click Edit Template in the Controls/Column window when the Web Layout Editor is open.

Setting Language Options

An object can include properties that are specific to a locale. For example, you can define a text string in several languages, such as an applet title. The Language Mode determines how Siebel Tools handles this locale data in the following situations:

- Data displayed in Siebel Tools
- Data available to edit
- Data compiled to the SRF file
- Data transferred to and from the Siebel Server during the check-in and check-out process

Siebel Tools displays the current Language Mode in the status bar located in the lower right corner of the Siebel Tools interface.

For more information, see *Using Siebel Tools*.

To set language options

- 1 In Siebel Tools, choose the View menu, and then the Options menu item.
- 2 Click the Language Settings tab.
- 3 In the language list, choose a language.
- 4 (Optional) To edit locale attributes, make sure the Enable and Use Language Override option contains a check mark.

If the Enable and Use Language Override option contains a check mark, then you can switch between the following modes:

- **Base.** Siebel Tools stores changes you make to locale-specific attributes in the base table. Siebel Tools applies these changes to every language.
- **Language Override.** Siebel Tools stores changes you make to locale-specific attributes in a child locale table. Siebel Tools overwrites the attribute for the current language.

If the Enable and Use Language Override option does not contain a check mark, then you can work only in Base model.

Process of Modifying Siebel Objects

To develop a Siebel application, you perform the following tasks:

- 1 [“Checking Out Projects from the Siebel Server” on page 183](#)
- 2 [“Making Configuration Changes” on page 184](#)
- 3 [“Compiling Projects” on page 184](#)
- 4 [“Logging In as the Siebel Administrator” on page 185](#)
- 5 [“Registering a New View” on page 186](#)
- 6 [“Verifying Your Changes” on page 187](#)
- 7 [“Checking Projects In to the Siebel Server” on page 187](#)

Checking Out Projects from the Siebel Server

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

To do configuration work in a local database, you first check out projects from the Siebel Server. For more information, see [“About the Local Development Environment” on page 18](#).

To check out projects from the Siebel Server

- 1 In Siebel Tools, choose the Tools menu, and then the Check Out menu item.
- 2 In the Check Out dialog box, choose the correct repository.
The default value is Siebel Repository.
- 3 Choose the project or projects that contain the object definitions you must check out.
- 4 Click Options.
- 5 In the Development Tools Options dialog box, make sure the Server and Client data sources are specified correctly, and then close the Development Tools Options dialog box.
- 6 In the Check Out dialog box, click Check Out.

Making Configuration Changes

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

When you check out a project, the project remains locked in the Siebel database on the Siebel Server and in your local database. This prevents other developers from checking out the project. During this time, you can use Siebel Tools to make configuration changes, such as modifying user interface objects, extending business logic, or extending the database.

You can also lock projects locally without checking them out from a server database. For more information, see [“Managing Projects When You Prototype” on page 189](#).

Compiling Projects

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

After you make your configuration changes, you must compile them to a Siebel repository. Until you compile, the Siebel Mobile Web Client that reads the repository file does not include configuration changes you made since the prior compile. You can compile entire projects or you can compile individual objects. Compiling an individual object is faster, but you must remember to compile each object you modify.

To compile projects

- 1 Close any Siebel clients that reference the SRF file you must compile.

While running, a Siebel client maintains a lock on the SRF file.

- 2 In Windows Explorer, make a backup copy of the SRF file.

So that you can revert to the prior configuration, it is recommended that you create a backup copy. The SRF file for the Siebel application resides in the objects directory of the Siebel application client directory. For example, the following directory is a typical location for the SRF file for a Siebel Mobile Web Client that you install for testing in a development environment:

```
C: \Siebel\8.2\Client_1\OBJECTS\ENU\si_ebel.srf
```


- 3 In Siebel Tools, choose the Tools menu, and then the Compile Projects menu item.
- 4 In the Object Compiler dialog box, in the Projects window, choose the project you must compile.
Choose the NREC Configuration project for work you perform on the NREC example. You can also compile all locked projects or all projects. Compiling only the locked projects is faster than compiling all projects. If you work in multiple projects, then compiling locked projects might be easier than choosing individual projects from the list.
- 5 Click Browse, and then choose the SRF file you must compile.
NOTE: Do not compile to the objects directory for Siebel Tools. The SRF file in this directory is locked because Siebel Tools frequently reads from it while it runs. If you attempt to compile to this SRF file, then Siebel Tools displays an error message and prevents the compile. The following directory is an example of the objects directory for Siebel Tools:
C:\Siebel\8.2\Tools_1\OBJECTS\ENU\siebel.srf.
- 6 Click Compile.
After Siebel Tools successfully compiles, the SRF file you specified contains the configuration changes you made.

To compile an individual object

- 1 In the Object Explorer, click the object type you must compile.
- 2 In the Object List Editor, locate the object definition for the object you must compile.
- 3 Right-click the object, and then choose the Compile Selected Objects menu item.
- 4 In the Object Compiler dialog box, choose the SRF file you must compile.
- 5 Click Compile.

Logging In as the Siebel Administrator

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

To perform administrative tasks in the Siebel client, you must use the *administrator* user name and password that your database administrator assigned you.

To log in as the Siebel Administrator

- 1 In Microsoft Windows, click Start, choose Programs, *Siebel_home*, Siebel Web Client *release number*, and then choose the Siebel Partner Manager menu item.

For example, click Start, choose Programs, Siebel8 home, Siebel Web Client 8.2, and then choose the Siebel Partner Manager menu item. In the NREC example, you use the Siebel Partner Manager application.

- 2 In the Siebel Partner Manager dialog box, Enter SADMIN in the User ID and Password fields.

Field	Value
User ID	SADMIN
Password	SADMIN
Connect to	Do one of the following: <ul style="list-style-type: none"> ■ To developer locally, choose Local or Sample. For more information, see “Prototyping with the Sample Database” on page 63. ■ To administer values on the Siebel server, choose Server.

- 3 Click OK.

Logging in as SADMIN provides access to the screens and views you use to administer a Siebel application.

Registering a New View

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

If you add a new view to a Siebel application, then you must make it visible in the Siebel application before you can test it. If you work with a predefined view, then it is not necessary to register the view. A predefined view is already registered.

You typically do not modify or delete a record for a view that is already registered. You only modify a view record if you change the view name. To restrict access to a view, it is recommended that you not assign any responsibilities to the view rather than deleting the view.

To register a new view

- 1 Register the new view:
 - a Open the Siebel client, navigate to the Administration - Application screen, and then the Views view.

You must open the client for a Siebel employee application, such as Siebel Call Center or Siebel Sales. You cannot register a view in Siebel Partner Portal because Siebel CRM does not display the administrative views you must use to register a view.
 - b In the View list, click New.
 - c In the View Name field, click the select button.
 - d In The View dialog box, choose the new view, and then click OK.
 - e Enter a description.
 - f Make sure the Default Local Access option contains a check mark.

This option makes the view available in the Siebel Mobile Web Client.

- g Save the record.
- 2 Associate the new view with a responsibility:
 - a Click the Responsibilities link.
 - b In the Responsibilities list, locate the responsibility to which you must associate the view.
 You cannot add a new view to a responsibility that is part of the seed data that ships with the product. To associate a new view with a responsibility, you must do one of the following:
 - Create a new responsibility.
 - Use an administrative responsibility, such as Administrator.
 - Copy a predefined responsibility, such as Siebel Administrator, and then add the new view to the copy.
 - c In the Views list, click New.
 - d In the Add Views dialog box, choose the new view you must register, and then click OK.

Siebel CRM associates the view with the responsibility. If a user who is assigned the responsibility logs in to the Siebel application, then Siebel CRM displays the view in the Siebel client. For more information, see [“Creating Responsibilities” on page 38](#).

Verifying Your Changes

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

After you make new views available, you can log in to the Siebel client to verify the results of your configuration changes.

To verify your changes

- 1 Log in to the Siebel client.
- 2 Navigate to the objects you modified.
 For example, if you added a new applet to a view, go to the view to make sure Siebel CRM displays the applet at the appropriate location.
- 3 Note any differences between what you find and the expected display or behavior.
- 4 If you find problems, do the following work:
 - a Use Siebel Tools to make the required corrections.
 - b Compile required projects.
 - c Repeat [Step 1](#) through [Step 3](#).

Checking Projects In to the Siebel Server

This task is a step in [“Process of Modifying Siebel Objects” on page 183](#).

After you verify changes, you must check in to the development server any projects that reference the objects you modify. Use the following guidelines when you check in projects:

- Test objects before you check them in.
- Because other developers might depend on your configuration, it might be necessary for you to check in projects before your work is complete.

For more information, see [“About the Local Development Environment” on page 18](#).

To check projects in to the Siebel Server

- 1 Choose the Tools menu, and then the Check In menu item.
- 2 In the Check In dialog box, click Options.
- 3 To validate projects, click Validate in the Check In dialog box.

For more information, see [“Viewing Differences Between Your Local Repository and the Server Repository” on page 188](#).

- 4 In the Development Tools Options dialog box, do the following:
 - Make sure the Siebel Server references the correct database.
 - Make sure the data source references the correct database.
- 5 Close the Development Tools Options dialog box.
- 6 Click Locked/New Projects.
- 7 Click Check In.

Options for Managing Objects

This topic describes other tasks that you might find useful when you manage objects.

Viewing Differences Between Your Local Repository and the Server Repository

Before checking in changes, you can view the differences between projects on your local repository and the projects on the server repository. This is useful for finding errors or omissions before you commit changes to the Siebel Server.

To view differences between your local repository and the server repository

- When you check in projects in Siebel Tools, in the Check In dialog box, click Diff.

Siebel Tools displays the checked out projects that you modified since the last check-out.

Managing Projects When You Prototype

If you prototype ideas or make changes to objects that you plan to discard, then it is recommended that you lock the project locally rather than checking it out from the Siebel Server. This way the project remains unlocked on the server, which allows other developers to check out the project while still allowing you to prototype temporary changes in your local repository.

To manage projects when you prototype

- 1 Open Siebel Tools.
- 2 In the Object List Editor, choose the project or other object type you must modify, choose the Tools menu, and then the Lock Project menu item.

Siebel Tools locks the project in your local repository so you can make changes. It does not lock the project on the server database. Note that because you cannot check your changes in to the Siebel Server, you cannot save any changes you make to the server.

NOTE: Do not lock a project locally on the server repository.

Exporting Object Definitions to Another Repository

You can export object definitions to another repository, which is useful if you work in a development environment that includes multiple repositories.

To export object definitions to another repository

- 1 In Siebel Tools, in the Object Explorer, click the object type you must export.
- 2 In the Object List Editor, locate the object you must export.
- 3 Choose the Tools menu, and then the Add to Archive menu item.
- 4 In the Export to Archive File dialog box, in the Archive File window, enter the path to the directory where you must save the archive file.
- 5 Click Save.

Siebel Tools saves the file with an SIF extension. Siebel Tools overwrites any SIF files that contain the same name that already exist in the directory. You cannot add objects to an export file.

Importing Object Definitions from Another Repository

You can import object definitions from an archive file to your local repository.

To import object definitions from another repository

- 1 Make sure you must possess access to the archive files that include the object definitions you must import.

- 2 Make sure the target repository is the active repository on your local computer.

A local database or a server database can contain more than one repository. However, Siebel Tools displays information from only one repository at a time. This is the *active repository*.

If the target repository is not active, then do the following:

- a Choose the File menu, and then the Open Repository menu item.
 - b In the Open Repository dialog box, choose the repository you must open, and then click Open.
- 3 Lock the projects that will contain the objects you must import.
 - 4 Choose the Tools menu, and then the Import From Archive menu item.
 - 5 In the Select Archive File to Import dialog box, choose the archive file you must import.
For example, objects.sif.
 - 6 Click Open.
 - 7 In the Import Wizard dialog box, choose a conflict resolution option.
 - 8 Click Next.

The Import Wizard checks for differences between the object definitions in the current repository and the object definitions in the archive file.

If there are no differences, then Siebel Tools imports the object definitions.

If there are differences, then the Import Wizard displays the following:

- **Conflicting Objects.** Displays the object types in which differences were found.
- **Object Differences.** Displays whether the object exists in the following items:
 - Archive file only
 - Current repository
 - Archive file and in the current repository

It also displays the conflict resolution action that is defined. You can change the action.

- **Attribute Differences.** Displays the object attributes that include different values. If the action for the object difference is *merge*, then to update the value in the Resolution column, you can right-click attribute, and then choose Repository or File.

14 Guidelines for Testing a Deployment

This chapter describes guidelines for testing a deployment. It includes the following topics:

- [Guidelines for Integration Testing on page 191](#)
- [Guidelines for User Acceptance Testing on page 192](#)
- [Guidelines for Performance Testing on page 193](#)
- [Guidelines for Continuous Improvement on page 194](#)
- [Benefits of Functional Test Automation on page 195](#)
- [Benefits of Load Test Automation on page 196](#)

For more information, see [“Testing You Perform When You Develop a Siebel Application” on page 14](#).

Guidelines for Integration Testing

Completion of the Functional Testing process verifies that the Siebel application functions correctly as a unit. It is recommended that you use the following guidelines during integration testing:

- Use a repeatable, measurable, and documented process. For more information, see [Figure 25 on page 192](#).
- Verify that each unit functions correctly when you include it in the larger environment.
- To test the integration points between the Siebel application and other applications or components, define test cases. For example:
 - Back-office applications
 - Integration middleware
 - Network infrastructure components
 - Security infrastructure
- Make sure your test exercises integration logic.
- Make sure your test validates the business process.

Figure 25 illustrates the integration testing process.

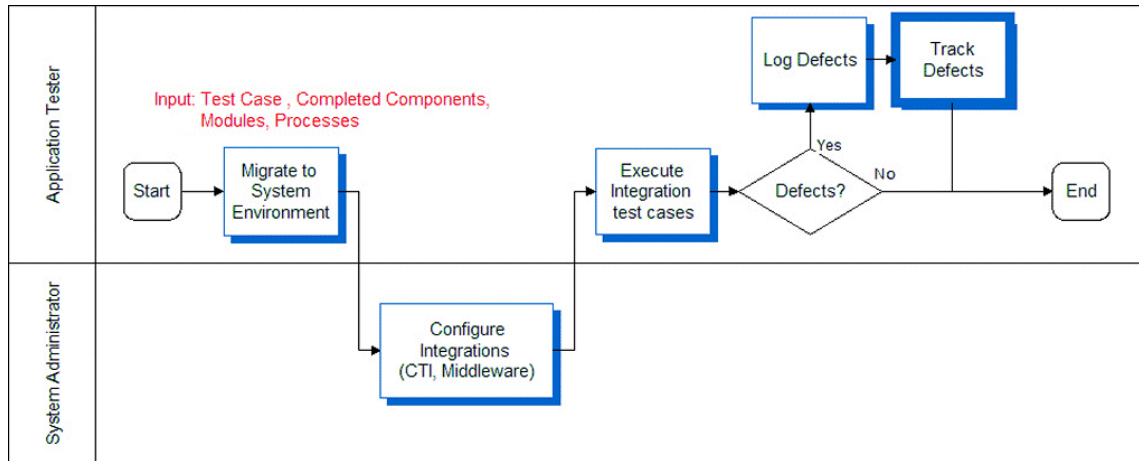


Figure 25. Integration Testing Process

Guidelines for User Acceptance Testing

It is recommended that you use the following guidelines during acceptance testing:

- Use a repeatable, measurable, and documented process. For more information, see [Figure 26 on page 193](#).
- Make sure that the functionality is acceptable to business users.
- Include the business user as an approver at each stage of the project.
- Make the deployment available to a larger community of trained users.
- Design user acceptance testing to simulate live business as closely as possible.
- Get approval from the user community to move to the next stage of development.

Figure 26 illustrates the acceptance testing process.

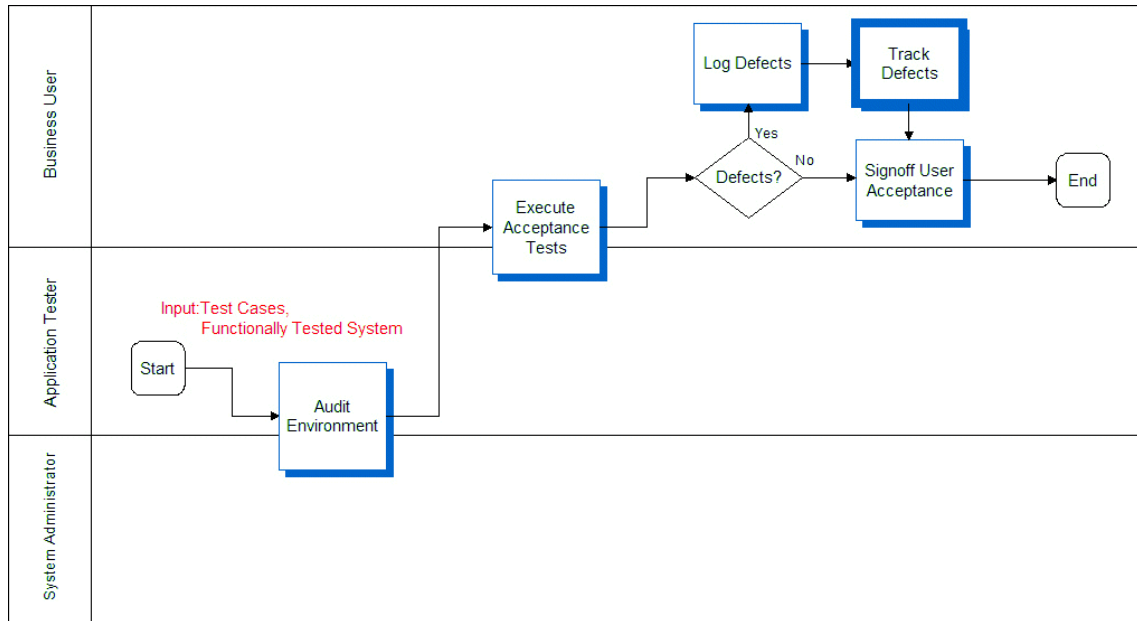


Figure 26. Acceptance Testing Process

Guidelines for Performance Testing

It is recommended that you use the following guidelines during performance testing:

- Use a repeatable, measurable, and documented process. For more information, see [Figure 27 on page 194](#).
- Perform the following tests:
 - Response time test
 - Stress test
 - Reliability test

Each test measures different types of performance. Individuals and groups who possess ownership of the architecture and infrastructure typically perform and administer these tests.

- To validate the health of the environment, test each custom script you use.
- Before you attempt the full user load, obtain a low load baseline. To use this baseline to measure scalability, compare results between the baseline and the target loads.

Guidelines for Testing a Deployment ■

- To prevent excessive resource utilization due to large numbers of users who simultaneously login, start users at a controlled rate. This rate depends on the total configured capacity of the deployment. For every one thousand users of configured capacity, add one user every three seconds. For example, if the product is configured for five thousand users, then add five users every three seconds.

Excessive login rate can cause the application server tier to consume 100% CPU, and logins begin to fail.

- To prevent inaccuracies during load testing due to simulated users who simultaneously perform transactions, randomize the *think time*, which is the amount of time the user requires before the user provides input to the server.
- Set randomization ranges according to the relative think times of expert and new users when compared to the average think times in the script.

Figure 27 illustrates the performance testing process.

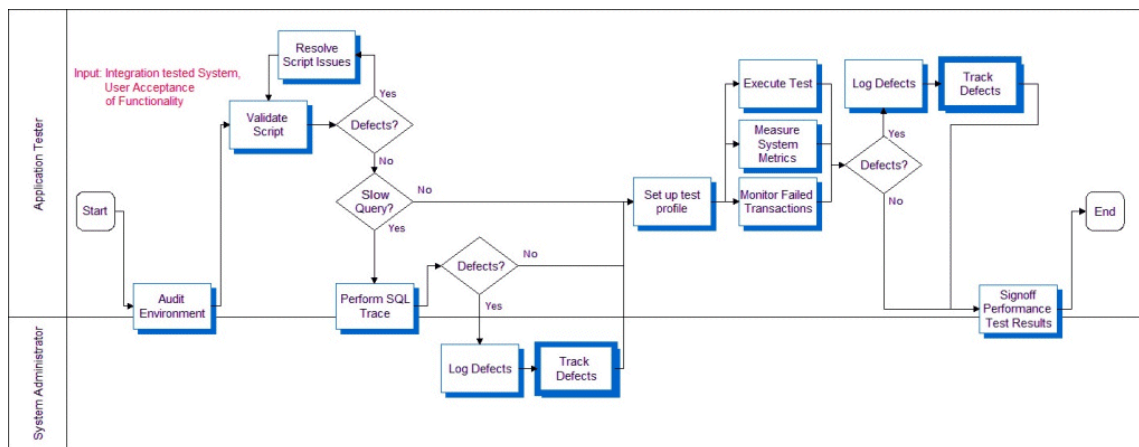


Figure 27. Performance Testing Process

Guidelines for Continuous Improvement

It is recommended that you use the following guidelines during continuous improvement:

- Use a repeatable, measurable, and documented process. For more information, see [Figure 28 on page 195](#).
- After the initial deployment, Oracle delivers periodic configuration changes in new releases. You must test these configuration changes.
- Identify a suite for your regression test. To identify regression problems that might be introduced, this suite provides an abbreviated set of test cases that you can run with each delivery. Streamlining the regression test process helps you to incorporate changes in the Siebel application at a much lower cost.
- Review the testing strategy and objectives. A full review of the open and closed problems can help calibrate the risk assessment you perform earlier in the product development lifecycle.

- Measure the observed risk that specific components cause. For example, identify the component that introduces the largest number of problems.
- Perform a final review meeting. This post-mortem meeting allows you to identify future improvements you can make.
- Review test plans and test cases. Update test cases to include testing scenarios during testing that were not previously identified.

Figure 28 illustrates the continuous improvement process.

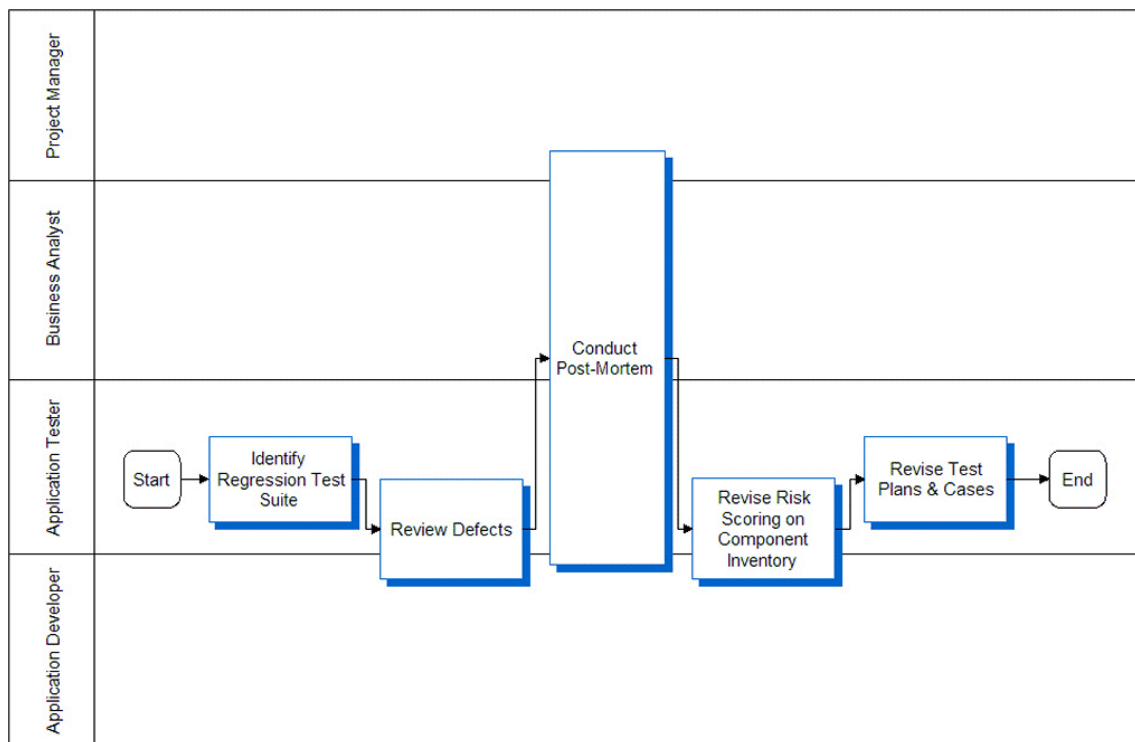


Figure 28. Continuous Improvement Process

Benefits of Functional Test Automation

Functional testing validates the functionality of your Siebel application. If you do not use test automation, then users must manually log in and manually perform numerous tasks. This technique can be effective but costly.

It is recommended that you use a test automation tool. You can prepare a test script that exercises the functionality of a particular module or that performs the tasks in a comprehensive business process. Then you can share and use this script repeatedly. This technique improves the configuration of your Siebel application, and it increased user acceptance because it allows you to perform software testing with little additional cost. Test automation reduces the requirement for multiple human test passes and reduces the amount of human error in the testing process.

Features of Functional Test Tools

Functional test automation tools provide the following features:

- **Test script recording.** Instead of composing a script in a text editor, you can start a recorder and perform a series of steps. The test tool records the session in a script, which you can replay later to perform the test.
- **Automated test execution.** Instead of manually performing tasks in the test script each time, you can use a test tool to start the test and then let the script run.
- **Data value parameterization.** You can use variables in your test script that pull in data values from an external table during test execution. This technique avoids the requirement to hard code data values in a test script.
- **Reusable tests.** You can share a test among multiple testers and across multiple cycles of application testing.
- **Result tracking.** The test tool records important statistics for identifying functional errors.

Not all functional test automation tools include all these features, and some tools include more features. For more information, see the documentation for your testing tool.

About Siebel Test Automation for Functional Testing

Siebel Test Automation objects support the automation of functional tests on the following types of Siebel clients:

- Siebel Web Client
- Siebel Mobile Web Client
- Siebel Developer Web Client

Siebel Test Automation does not provide general automation services. It only supports test automation.

You can use a third-party test automation tool to write, record, and run test scripts on these clients. If you use a third-party tool, then see the documentation that accompanies the tool for information about the features and functionality that it provides.

Benefits of Load Test Automation

Load testing measures *key performance indicators* (KPIs) of your Siebel application while it is under a load of multiple users. Response time and reliability are examples of KPIs. To perform load testing without load test automation, you can direct users to log in and run through a business process during a specified period of time. This technique is effective but costly.

You can use a test automation tool to simulate the load and achieve comprehensive and precise results without the cost associated with multiple human testers. Load testing automation also reduces the risk associated with user errors.

Features of Load Test Tools

Tools for load test automation provide the following features:

- **Recording test scripts.** For more information, see [“Features of Functional Test Tools” on page 196.](#)
- **Using virtual users.** Many virtual users from a single computer can simultaneously run a single test script.
- **Controlling the execution of the test.** Use the test tool to specify parameters that indicate how to run the load test. For example, how quickly to add users, how many users to add, and which scripts to run.
- **Tracking results.** The test tool records important statistics for tracking KPIs and isolating bottlenecks. Example statistics include memory usage and response time.

Not all load testing tools include all of these features, and some tools include more features. For more information, see the documentation for your load testing tool.

How You Can Use the Siebel Correlation Library to Perform Load Testing

The Siebel Correlation Library allows you to use a third-party tool to automate load testing on a Siebel application. This is a dynamically linked library (DLL) that provides services to help the testing tool generate and run test scripts against your Siebel application.

After you install the Correlation Library and set up your test environment, the Correlation Library helps the test tool translate the recorded test session in a script that you can run to test your Siebel application. The Correlation Library acts as an interpreter for the test tool. The test tool does the following work:

- Records HTTP traffic from the Siebel application. This HTTP traffic equates to a Web page.
- Sends the Web page to the Correlation Library.

The Correlation Library parses the Web page and returns to the test tool the appropriate correlation information for the Web page. The correlation information allows you to parameterize data values.

For more information, see the chapter about automating load tests in *Testing Siebel Business Applications*.

The Correlation Library can parameterized the following entities:

- **Application data entities.** Application data entities are associated with the function of the Siebel application. Examples of application data entities include Row ID, SWE Timestamp, and SWE Count.
- **Transaction data entities.** Transaction data entities contain record data that the user enters in the Siebel application. Examples of transaction data entities include Contact Name and Account Name. The Correlation Library does not handle transaction data entities. You must parameterize them manually.

Setting Up Your Environment to Use the Correlation Library

You can set up your environment to support automated load testing. Note that the Correlation Library is licensed for use only by licensed users of Siebel Test Automation. If you have not purchased a license for Siebel Test Automation, then you cannot write to or access the Correlation Library. For more information, see *Testing Siebel Business Applications*.

To set up your environment to support automated load testing

1 Install your load testing tool.

2 Locate the ssdtcorr.dll file in your Siebel Server installation.

The ssdtcorr.dll file includes the Correlation Library. It is located in one of the following directories:

- si ebsrvr\bin on Windows
- si ebsrvr/lib on UNIX

3 Copy the ssdtcorr.dll file to a location that is accessible to the load testing tool.

For example, the bin directory of the load testing tool.

4 Set up your test tool so that it communicates properly with the Correlation Library.

For more information, see the documentation from your test tool vendor.

A

Code for NREC Business Services

This appendix includes code for the NREC business services. It includes the following topics:

- ["Code for the General Function" on page 199](#)
- ["Code for the Service Function" on page 205](#)

For more information, see ["Creating a Business Service" on page 132](#).

Code for the General Function

This topic includes code you copy to the general function. It includes the following topics:

- ["Code for the Declaration Function" on page 199](#)
- ["Code for the Pal_Init Function" on page 199](#)
- ["Code for the Pal_Insert Function" on page 200](#)
- ["Code for the Pal_PreInsert Function" on page 201](#)
- ["Code for the Pal_Query Function" on page 202](#)

Code for the Declaration Function

Copy the following code to the declaration function:

```
Dim FieldCount As Integer
Dim NoOfRecords As Integer
Dim DataSource As String
Dim ColumnsReqd As String
Dim VBCName as String
```

Code for the Pal_Init Function

Copy the following code to the Pal_Init function:

```
Sub Pal_Init (Inputs as PropertySet, Outputs as PropertySet)

    Dim theFieldName(100) As String
    Dim ParamText as String

    ' Identify setup details of calling VBC and add as Parameter to Service

    ParamText = Inputs.GetProperty("Parameters")
    VBCName = Inputs.GetProperty("Business Component Name")

    Me.SetProperty VBCName, ParamText
    me.SetProperty VBCName & "Qry", "None"
```

```

If ParamText <> "" Then
    DataSource = Left(ParamText, Instr(ParamText, ";")-1)
    ParamText = Mid(ParamText, Instr(ParamText, ";")+1)
    FieldCount = CVar(Left(ParamText, Instr(ParamText, ";")-1))
    ColumnsReqd = Mid(ParamText, Instr(ParamText, ";")+1)
End If

' Initialize the full list of Fields Available

Open DataSource For Input As #1
    For Count = 1 To FieldCount
        Input #1, theKey
        theFieldName(Count) = RTrim(theKey)
        Outputs.SetProperty theFieldName(Count), ""
    Next Count
Close #1

End Sub

```

Code for the Pal_Insert Function

Copy the following code to the Pal_Insert function:

```

Sub Pal_Insert (Inputs As PropertySet, Outputs As PropertySet)

    Dim InputFields(100) As String
    Dim InputValues(100) As String
    Dim NoOfValues As Integer
    Dim InputPS As PropertySet
    Dim OutputStr As String
    Dim PadReqd As Integer

    NoOfValues = 0

    ' *****
    ' * Establish the Value Pairs to Be Used for Insert from the Inputs Property set *
    ' *****

    If Inputs.GetFirstProperty <> "" Then
        Set InputPS=TheApplication.NewPropertySet()
        Set InputPS = Inputs.GetChild(0)
        InputFields(1) = InputPS.GetFirstProperty
        InputValues(1) = InputPS.GetProperty(InputPS.GetFirstProperty)
        NoOfValues = NoOfValues + 1
        For i = 2 to FieldCount
            InputFields(i) = InputPS.GetNextProperty
            If InputFields(i) <> "" Then
                InputValues(i) = InputPS.GetProperty(InputFields(i))
                NoOfValues = NoOfValues + 1
            End If
        Next i
    End If

End Sub

```



```

' *****
' * Create the Output String for Writing to the File in the right order *
' *****

Open DataSource For Input As #1

For Count = 1 To FieldCount
    PadReqd = 1 ' Identify by default we need a Pad string in the file
    Input #1, theKey
    For i = 1 to NoOfValues
        If InputFields(i) = theKey Then
            PadReqd = 0
            If OutputStr <> "" Then
                OutputStr = OutputStr & ","
            End If
            OutputStr = OutputStr & InputValues(i)
        End If
    Next i
    If PadReqd = 1 Then
        If OutputStr <> "" Then
            OutputStr = OutputStr & ","
        End If
        OutputStr = OutputStr & " "
    End If
Next Count

Close #1

' *****
' * Create the Output String In the DataSource File *
' *****

Open DataSource For Append As #1

Print #1, OutputStr

Close #1

End Sub

```

Code for the Pal_PreInsert Function

Copy the following code to the Pal_PreInsert function:

```

Sub Pal_PreInsert (Outputs As PropertySet)

' *****
' * A Pre Insert Method is Required before any insert is applied, It *
' * provides Default Values, This service does not provide any defaults, But *
' * must return a property set that includes values, so this references the first field
by default. *
' *****

```

```

dim newRow As PropertySet
set newRow = TheApplication.NewPropertySet()
newRow.SetProperty Left(Col umnsReqd, Instr(Col umnsReqd, ",")-1), ""
Outputs.AddChild newRow

```

```
End Sub
```

Code for the Pal_Query Function

Copy the following code to the Pal_Query function:

```

Sub Pal_Query (Inputs as PropertySet, Outputs as PropertySet)
' *****
' * The adding of properties in the me.SetProperty statements in this      *
' * code is used To store key information and the relevant data against  *
' * parameters identified by the VBC name. This technique is used to create *
' * a persistantstore and negate yhe need for repeat calls to the external *
' * source for the same data.                                           *
' *****

Dim Row(999) As PropertySet

Dim theFieldName(100) As String
Dim QueryValues(100) As String
Dim QueryFields(100) As String
Dim QueryList As PropertySet
Dim ColList, ColData, QryData As String
Dim OutColCount, OutRowCount, NoOfQs, RowMatches, DoQuery As Integer

DoQuery = 1
NoOfQs = 0
OutColCount = 0
OutRowCount = 0

' *****
' * Establish the list of Columns Which define the Query Values from the VBC *
' *****

Set QueryList = TheApplication.NewPropertySet()
Set QueryList = Inputs.GetChild(0)

' Inputs.GetChild(0) retrieves user inputs ,like ("First Name","Angela") , etc...

If QueryList.GetFirstProperty <> "" Then

    QueryFields(1) = QueryList.GetFirstProperty
    QueryValues(1) = QueryList.GetProperty(QueryList.GetFirstProperty)
    NoOfQs = NoOfQs + 1
    QryData = QueryFields(1) & "=" & QueryValues(1)
    For i = 2 to FieldCount
        QueryFields(i) = QueryList.GetNextProperty
        If QueryFields(i) <> "" Then
            QueryValues(i) = QueryList.GetProperty(QueryFields(i))
            NoOfQs = NoOfQs + 1

```

```

        QryData = QryData & ";" & QueryFields(i) & "=" & QueryValues(i)
    End If
Next i
End If

' *****
' * Establish If Query is same as Last Time for this VBC and If So Set Query Flag *
' * to use Persistent Cache
' *****

    If me.GetProperty(VBCName & "Qry") <> QryData Then
        me.SetProperty VBCName & "Qry", QryData
    Else
        DoQuery = 0
    End If

' *****
' * Establish Full List of Columns Available and Which ones are needed by the VBC *
' * Where not required, Identify this with Column Value Not Required CVNR *
' *****

    If DoQuery = 1 Then

        Open DataSource For Input As #1
        For Count = 1 To FieldCount
            Input #1, theKey
            If Instr(ColMnsReqd, theKey) <> "0" Then
                theFieldName(Count) = RTrim(theKey)
                OutColCount = OutColCount + 1
            Else
                theFieldName(Count) = "CVNR"
            End If
        Next Count

        me.SetProperty VBCName & "ColCnt", CStr(OutColCount)

' *****
' * Retrieval of Data from Text File, Including ignoring Columns Not Required and *
' * Ignoring rows which do not match the selection criterion *
' *****

        For r = 1 to 999

            On Error Goto EndLoop
            rowMatches = 1 'set to true
            Input #1, theKey
            theValue = RTrim(theKey)
            If theValue <> "" Then
                Set Row(r) = TheApplication.NewPropertySet()
                If theFieldName(1) <> "CVNR" Then
                    Row(r).SetProperty theFieldName(1), theValue
                    ColList = theFieldName(1)
                    ColData = theValue
                    For x = 1 to NoOfQs

```

```

        If QueryFields(x) = theFieldName(1) and QueryValues(x) <> theValue Then
            rowMatches = 0
        End If
    Next x
End If
For Count = 2 To FieldCount
    Input #1, theKey
    theValue = RTrim(theKey)
    If theFieldName(Count) <> "CVNR" Then
        Row(r).SetProperty theFieldName(Count), theValue
        ColList = ColList & ";" & theFieldName(Count)
        ColData = ColData & ";" & theValue
    For x = 1 to NoOfQs
        If QueryFields(x) = theFieldName(Count) and QueryValues(x) <> theValue Then
            rowMatches = 0
        End If
    Next x
    End If
Next Count
If rowMatches = 1 Then
    Outputs.AddChild Row(r)
    OutRowCount = OutRowCount + 1
    me.SetProperty VBCName & CStr(OutRowCount), ColData
End If
Else
EndLoop:

    Exit For
End If
Next r
    Close #1
    me.SetProperty VBCName & "RowCnt", CStr(OutRowCount)
    me.SetProperty VBCName & "Col", ColList
Else

' *****
' * Re Create Output property set from the Persistent Cache held in the Business *
' * Service Property Set
' *****

    OutColCount = CVar(GetProperty(VBCName & "ColCnt"))
    OutRowCount = CVar(GetProperty(VBCName & "RowCnt"))
    For i = 1 to OutRowCount
        ColList = GetProperty(VBCName & "Col")
        ColData = GetProperty(VBCName & CStr(i))
        Set Row(i) = TheApplication.NewPropertySet()
        For x = 1 to OutColCount-1
            theField = Left$(ColList, Instr(ColList, ";")-1)
            theValue = Left$(ColData, Instr(ColData, ";")-1)
            ColList = Mid$(ColList, Instr(ColList, ";")+1)
            ColData = Mid$(ColData, Instr(ColData, ";")+1)
            Row(i).SetProperty theField, theValue
        Next x
    Next i

```

```

        Row(i).SetProperty ColList, ColData
        Outputs.AddChild Row(i)
    Next i
End If

End Sub

```

Code for the Service Function

Copy the following code to the Service_PreInvokeMethod function:

```

Function Service_PreInvokeMethod (MethodName As String, Inputs As PropertySet, Outputs
As PropertySet) As Integer

```

```

    Dim ParamText as String

' *****
' * SYSTEM DEFINITION SECTION
' *
' * The Calling V.BusComp must include the following parameters
' *
' * Service Name = Pal Mk2 (Unless Renamed)
' * Service Parameters = <Full Path & DataFile Name>;
' *                   <No Of Columns in File>;
' *                   <Columns in Calling VBC comma separated>
' *
' * e.g Service Parameters = c:\Siebel\Data.VBC;5;KeyId,FName,LName,Age,DOB
' *
' *****

' Identify the Context the Service is Called from and setup parameters

VBCName = Inputs.GetProperty("Business Component Name")
ParamText = me.GetProperty(VBCName)
If ParamText <> "" Then
    DataSource = Left(ParamText, Instr(ParamText, ";")-1)
    ParamText = Mid(ParamText, Instr(ParamText, ";")+1)
    FieldCount = CVar(Left(ParamText, Instr(ParamText, ";")-1))
    ColumnsReqd = Mid(ParamText, Instr(ParamText, ";")+1)
End If

' Handle the Method

If MethodName = "Init" Then
    Pal_Init Inputs, Outputs
    Service_PreInvokeMethod = CancelOperation
ElseIf MethodName = "Query" Then
    Pal_Query Inputs, Outputs
    Service_PreInvokeMethod = CancelOperation
ElseIf MethodName = "Update" Then
    ' Pal_Update Inputs, Outputs - Not Yet Completed
    Service_PreInvokeMethod = CancelOperation
ElseIf MethodName = "PreInsert" Then

```

```
    Pal_PreInsert Outputs
    Service_PreInvokeMethod = Cancel Operation
Else If MethodName = "Insert" Then
    Pal_Insert Inputs, Outputs
    Service_PreInvokeMethod = Cancel Operation
Else If MethodName = "Delete" Then
    ' Pal_Delete Inputs      - Not Yet Completed
    Service_PreInvokeMethod = Cancel Operation
Else
    Service_PreInvokeMethod = Continue Operation
End If
End Function
```

Index

A

- access requirements** 24
 - applet design, described** 13
 - applet Web templates, about** 181
 - applets**
 - about 181
 - attributes, modifying to display more 102
 - business requirements, modifying for 70, 71
 - colors, modifying 128
 - rule set, associating with 155
 - user interface data, creating to display 96
 - application administration Siebel administrator, logging on as** 185
 - application administration tasks** 42
 - partner, registering and promoting to an organization 48
 - users, setting up 36
 - views, registering 186
 - applications, installing** 33
 - architecture**
 - about Siebel Object architecture 20
 - archive files, importing** 189
 - Assignment criteria, defined** 145
 - Assignment Manager**
 - about 143
 - assignment criteria, tasks to create 143
 - Assignment Manager rules, activating 146
 - assignment rules, adding positions for each rule 145
 - assignment rules, creating according to territories 143
 - assignment rules, releasing 146
 - assignment rules, described** 13
 - automation**
 - functional testing 195
 - load testing 196
- ## B
- banner**
 - background color, modifying 126
 - color scheme, about modifying 126
 - frame, adding new logo to 124
 - new logo, about adding 124
 - business component**
 - adding field to 101
 - business object, adding to 95

- child business component, creating link 94
- creating 92
- design, described 12
- business object design, described** 13
- business object layer** 29
- business object, adding to business component** 95
- business rules**
 - See Assignment Manager; Siebel Personalization
- business services**
 - business service, defining 132
 - creating, about 133
 - scripts, defining and writing 133
 - scripts, table of methods 132

C

- Check In/Out tab, about** 182
- child business components**
 - business component, creating link 94
 - parent business components, creating link between 136
- common test definitions** 14
- company structure, defining** 42
 - access to data, understanding 43
 - division, reviewing 44
 - entities to setup 43
 - note, about changing company structure and routers 43
 - organizations and organizational skills, about setting up 43
 - organizations, setting up 39
 - position skills, setting up 47
 - positions, setting up 44
 - responsibilities, about defining 38
 - responsibilities, copying a predefined 39
- configuration file**
 - editing, about 172
 - example file 172
 - preferences, setting 163
- configuring views** 75
- controls**
 - removing 69
 - Web templates, mapping to 72
- creating entity relationship diagram** 140
- creating strings** 80, 82

D**data layer** 29**data source, logging on as an administrator to** 185**database**

base table, extending by adding new columns 77

developers local database, initializing 61

populating (full get) 63

Sample, about installing 34

schema changes, adding to 77

Siebel Mobile Web Client, running database extract for 56

database extensions, adding field to business component 101**database server component** 55**database software**

Development environment, installed in 34

Test environment, installed in 35

database template, generating new 55**database users**

database account name, setting up 36

setting up, about 37

definitions, testing terminology 14**design, review** 28**develop tests** 14**developer's local database, setting up**

database users, about setting up 37

developer's local database, initializing 61

developers, setting up as Siebel Mobile Web Clients 54

full get, performing 63

local database, extracting 56

local development environment, about working in 18

positions, about creating 44

responsibilities, about associating 38

development 14**development environment**

described 17

software installed, table of 33

development strategy 27**division, reviewed for organization** 44**Docking Transaction Logging, about disabling for active mobile clients** 175**documentation**

development 140

types of example project design documentation 12

drilldowns

dynamic drilldowns, about and creating 108

dynamic drilldowns, creating drilldown

destination objects 115

static and dynamic, about and example 108

dynamic drilldowns

about and creating 108

about and example 108

creating drilldown destination objects 115

dynamic picklists

creating 112

defined 109, 111

defined and related objects 107

E**EIM**

configuration file, editing 172

import process, reviewing 177

interface tables, determining which columns are required 171

interface tables, loading 172

interface tables, process flows 171

process overview 170

transaction logging, disabling 175

email

Siebel Workflow, configuration steps to use 147

template, creating 147

employee

Siebel application, setting up 41

users, described 37

Enable Language Override check box, about 183**entity relationship diagram**

creating 140

described 12

purpose 140

See entity relationship diagram

environment

production 36

eRoadmap

development methodology 14

stages of methodology 12

exporting object definitions 189**extension tables**

data, displaying data from 92

one-to-many, about 91

F**fields, displaying in the Siebel client** 71**full get, performing** 63**functional test automation** 195**functional tests** 15**G****gap analysis, described** 25

generating new database template 55
goals 11

H

House entities

base table, extending by adding new columns to 77
 compiling and testing 90
 database schema changes, adding to 77
 House Detail view, creating 86
 Houses screen, creating 88
 Internal Products business components, adding fields to 79
 Page Tab and Screen Menu Items, defining 90
 Product Form Applet, configuring 85
 Product List Applet, configuring 83

Houses screen

compiling and testing 90
 creating 88
 Page Tab and Screen Menu Items, defining 90

I

IBM HTTP Server

development environment, installed in 35
 test environment, installed in 36

importing object definitions 189

improve testing 194

installation

application rollout, stages in 17
 Development environment, table of software installed 33, 35
 environments, described 17, 18
 test environment, table of software installed 35, 36

installing Siebel applications 33

integration testing

defined 15
 executing 15, 191

interface tables

columns, determining which are required 171
 loading 172

Internal Products business components, adding fields to 79

introduction, application software testing 16

L

Language Setting tab, about layer 182

business object 29

data 29
 Siebel client 30

links

parent and child business components, creating links between 136

list columns

removing 69
 Web templates, mapping to 72

load test automation 196

localization 80

M

methodology, eRoadmap development 14

Microsoft Internet Explorer

development environment, installed in 35
 test environment, installed in 36

Microsoft Internet Information Server (IIS)

development environment, installed in 35
 test environment, installed in 36

moving

See development environment; production environment; test environment

Multi-value groups

about and example 115
 creating 117
 Siebel client, displaying MVG in 116

MVG

See Multi-value groups

N

National Real Estate Clearinghouse (NREC)

about 11
 access requirements 24
 business requirements 25
 example project design documentation, types of 12
 project resources, example breakdown 21
 Siebel eRoadmap methodology, stages of 12
 solution design described 28
 user profiles 24

NREC

See National Real Estate Clearinghouse (NREC)

NREC business objectives 23

NREC project team 21

NREC requirements

activities 26
 business object layer 29
 contacts 27
 data layer 29
 design review 28
 manage houses for sale 25
 opportunities 26

user interface layer 30

O

object definitions

architecture 20
differences, viewing before check in 188
exporting 189
importing 189
local repository, locking projects in 189
Object List Editor, about using to list 180

object layers

architectural layers diagram and layers 20

Object List Editor, about 180

Object Properties window, displaying and hiding 180

Objects Explorer, about using 179

objects, compiling individual objects 185

one-to-many extension tables

about 91
data, displaying data from 92

Opportunity Details view

applets, modifying to display more attributes 102
compiling and testing 105
configuring, about 99
Opportunity business component, adding fields to 101

organizations

and organization skills, about setting up 43
division, reviewing 44
partner, registering them and promoting to 48
setting up 39

P

Page Tab, defining 90

parent business components, creating link between child business components 136

partner

registering and promoting to an organization 48
users, about 37

performance testing

defined 16
executing 16, 193

Person users, about 37

Pick applet, about 107

Pick List object, about 107

pick maps

about 107
filtering records 108

picklists

dynamic picklists, creating 112
dynamic picklists, defined and related objects 107
filtering records 108
static picklists, creating 109
static picklists, defined and example 107
types of 109, 111

plan testing strategy 14

positions

creating 44
creating, about 44
logging on, changing while 43
skills, setting up 47

Product Form Applet

configuring 85
view, associating it with a screen 98
view, creating to display applet 97

Product List Applet, configuring 83

production environment 36

described 17
software installed, about 36

products table, adding new columns 77

project team, NREC 21

projects

checking in 187
checking in and out, defined 18
checking out 183
compiling 184
creating new project 37, 60
local repository, locking projects 189

Properties window, displaying and hiding 180

R

regression testing

defined 16

report templates, described 13

repository

moving from development to test environment 169
projects, locking in 189

responsibilities

associating, about 38
defining, about 38
predefined responsibility, copying 39

review design 28

rule set, associating with applet 155

rules, activating

generate triggers, running 153
Workflow Monitor Agent, starting 154

S

sample data source, logging on as

- administrator to 185
- Sample database, about installing** 34
- scope** 11
- screen design template, described** 13
- Screen Menu Items, defining** 90
- screenbar**
 - about and controls 127
 - modifying 127
- screens**
 - inactivating, about 65
 - inactivating, screen page tabs and screen menu items 66
- server data source, logging on as an administrator to** 185
- Siebel** 180
- Siebel administrator. logging on as** 185
- Siebel client**
 - Multi-value group, displaying in the Siebel client 116
 - new view, creating 138
 - user interface elements 120
- Siebel database schema, installed in test environment** 35
- Siebel Developer Web Client**
 - installing, about 34
- Siebel Enterprise Integration Manager (EIM)**
 - See EIM
- Siebel eRoadmap, stages of methodology** 12
- Siebel File System**
 - development environment, installed in 34
 - test environment, installed in 35
- Siebel Gateway Server**
 - development environment, installed in 34
 - test environment, installed in 35
- Siebel Mobile Web Client**
 - caution, about disabling Docking Transaction Logging 175
 - development environment, installed in 34
 - local database, initializing 164
 - registering 164
 - starting synchronization session 165
 - synchronizing, process of 165
 - test environment, installed in 35
- Siebel Mobile Web Clients**
 - database extract, running 56, 59
 - developers, setting up as 54
 - hardware and software, about setting up 163
 - network connectivity, enabling 163
 - registering 163
 - Siebel Remote architecture diagram 161
- Siebel object definitions**
 - architecture 20
- Siebel objects**
 - individual objects, compiling 185
 - model hierarchy, understanding 21
- Siebel Personalization**
 - rule set, associating with applet 155
 - rule set, creating 155
 - rule, creating 155
 - rules, using to create 156
 - testing personalization rules 157
- Siebel Remote**
 - architecture diagram 161
 - authenticating Siebel Mobile Web Clients, about 163
 - network connectivity, enabling 163
 - preferences, setting 163
 - server, about setting up 162
 - Siebel Mobile Web Client hardware and software, about setting up 163
 - Siebel Mobile Web Client local database, initializing 164
 - Siebel Mobile Web Client, registering 163, 164
 - Siebel Mobile Web Client, running database extract 59
 - Siebel Mobile Web Client, setting developer up as 54
 - user, tasks to setup 162
- Siebel Server**
 - development environment, installed in 34
 - files, location of 121
 - test environment, installed in 35
- Siebel Tools**
 - configuration options, about 182
 - files, locations of 121
 - installing, about 35
 - Object Explorer, about 179
 - Object List Editor, about 180
 - Properties Window, about and displaying and hiding 180
 - Web layout editor, about and example 180
 - wizards, about and opening new wizard object 180
- Siebel Tools, using to configure**
 - objects, compiling 185
 - projects, checking in 187
 - projects, checking out 183
 - projects, compiling 184
- Siebel Web Client**
 - applet colors, modifying 128
 - banner background color, modifying 126
 - banner color scheme, about modifying 126
 - banner frame, adding new logo to 124
 - banner, about adding new logo 124
 - screenbar and viewbar, about and

- controls 127
 - screenbar and viewbar, modifying 127
 - Web templates and related files, source control 121
 - Siebel Workflow**
 - email template, creating 147
 - policy condition, creating 152
 - rules, running generate triggers 153
 - rules, starting Workflow Monitor Agent 154
 - sending email, configuration steps 147
 - workflow policies, creating 152
 - workflow policies, defined 151
 - workflow process, steps 149
 - software testing, introduction** 16
 - static drilldowns, about and example** 108
 - static pick lists**
 - creating 109
 - defined and example 107
 - static picklists**
 - defined 109, 111
 - strategy** 27
 - stress testing, defined** 16
 - strings**
 - symbolic 80
 - strings, symbolic** 80, 82
 - symbolic strings** 80, 82
 - synchronization**
 - defined 161
 - options, setting 163
 - starting session, mobile user 165
- T**
- team, NREC project** 21
 - test cases document, described** 13
 - test environment**
 - described 17
 - move, list of data to move 167
 - moving, preparing for 167
 - repository, moving 169
 - software installed, table of 35
 - Web templates and related files, moving 169
 - test plan** 191
 - testing**
 - automation of functional testing 195
 - automation of load testing 196
 - changes, verifying 187
 - developing tests 14
 - functional 15
 - functional, automation 195
 - improvements 16, 194
 - integration 15, 191
 - integration, defined 15
 - introduction 16
 - load testing, automation 196
 - performance 16, 193
 - performance, defined 16
 - plan strategy 14
 - prior to production 191
 - regression, defined 16
 - strategy document, described 13
 - stress, defined 16
 - test cases document 13
 - unit 105
 - unit, defined 15
 - usability, defined 15
 - user acceptance 15, 192
 - user acceptance, defined 15
 - views, associating with a responsibility 187
- Tools Language Mode, about** 182
- transaction logging**
 - disabling 175
- U**
- unit testing**
 - defined 15
 - executing 105
 - usability testing, defined** 15
 - user acceptance testing**
 - defined 15
 - executing 15, 192
 - user interface**
 - applet, creating to display data 96
 - user interface layer** 30
 - applets and applet Web templates, about 181
 - applets, modifying for business requirements 70
 - fields, displaying 71
 - inactivating page tabs and screen menu items, about 66
 - inactivating screens, about 65
 - inactivating views, about 67
 - inactivating views, associating with a screen 67
 - list columns and controls, removing 69
 - project, creating new project 37, 60
 - user key sequence, described** 172
 - user properties, defining for virtual business components** 135
 - users, setting up**
 - about 36
 - database users, setting up 36
 - positions, changing positions while logged on 43
 - Siebel application, setting an employee in 41
 - types of 36

V**view design, described** 13**viewbar**

- about and controls 127
- modifying 127

views

- inactivating 67
- inactivating associating with a screen 67, 68
- registering 186
- responsibility, associating with 187

views, configuring 75**virtual business components**

- about 132
- business object, updating 137
- business service scripts, defining and writing 133
- business service scripts, table of methods 132
- business services, about creating 133
- business services, defining a business service 132
- code example 199
- compiling and testing 139
- creating 135
- creating scenario 131
- fields, adding 135

- parent and child business components, creating link 136
- user properties, defining 135

W**Web layout editor, about** 180**Web server**

- development environment, installed in 35
- test environment, installed in 36

Web Template Editor tab, about 182**Web templates**

- development process 122
- list columns or controls, mapping to 72
- locations of 121
- modifying, about 180
- source control, about 121
- test environment, moving to 169

wizards, about and opening new wizard object 180**Workflow Monitor Agent, starting** 154**workflow policy**

- creating 152
- defined 151
- policy condition, creating 152
- templates, described 13

