SIEBEL eBusiness 7

DEVELOPING AND DEPLOYING SIEBEL *e*BUSINESS APPLICATIONS

VERSION 7.5

12-BD362N

JULY 2002

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Contents

Introduction

Chapter 1. Introducing the Business Case

Chapter 2. Installing Siebel Applications

Chapter 3. Getting Started with Siebel Tools

Chapter 4. Setting Up a Developer’s Local Database

Chapter 5. Getting Started at the User Interface Layer

Chapter 6. Configuring the House and Opportunity Entities

Chapter 7. Pick Lists, Drilldowns, and MVGs

Chapter 8. Creating a Virtual Business Component

Chapter 9. Modifying the Look and Feel of the Web Client

Chapter 10. Migrating Repository Data to the Test Environment

Chapter 11. Using EIM to Load Data into the Test Environment

Chapter 12. Required Application Administration Tasks

Chapter 13. Assignment Manager

Chapter 14. Siebel Business Process Designer

Chapter 15. Personalization

Chapter 16. Implementing Siebel Remote

Chapter 17. Deploying the Application

Introduction

This guide presents the major steps of a typical Siebel implementation and deployment. This guide:

- Helps readers understand the typical steps of designing, configuring, testing, and deploying Siebel applications.
- Presents tasks in the context of business needs.
- Provides hands-on examples of configuration tasks.
- Introduces readers to specific Siebel products and tell readers where to go for more information.

This book will be useful primarily to people whose titles or job descriptions match one of the following:

Call Center Administrators Persons responsible for setting up and maintaining a call center. Duties include designing and managing Computer Telephony Integration (CTI), SmartScripts, and message broadcasts.

Database Administrators Persons who administer the database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.

Marketing Administrators Persons responsible for setting up and maintaining a marketing department. Duties include designing and managing campaigns, product marketing information, and product distribution lists.

Siebel Application Administrators Persons responsible for planning, setting up, and maintaining Siebel applications.

Siebel Application Developers Persons who plan, implement, and configure Siebel applications, possibly adding new functionality.

Siebel System Administrators Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.

This book does assume a basic knowledge of Siebel applications. If you do not have any experience with Siebel products, review *Fundamentals*.

How This Guide Is Organized

The first few chapters in this book cover a sample business case and other basic information to get you started. The chapters in the remainder of the book are presented in an order that loosely corresponds with the three stages of an implementation: development, test, and production.

- Read chapters 1, 2, and 3 first. They will help you understand the business case used as an example throughout the book, the example software and hardware environments, and the basic tasks for using Siebel Tools.
- The tasks in chapters, 4, 5, 6, 7, 8, and 9 are designed to give you hands-on configuration experience. You can follow them like a tutorial, using your sample database as your datasource.
- The tasks in chapters 10, 11, 12, 13, and 14 chapters walk through examples of using Business Process Designer, Assignment Manger, Personalization, and Siebel Remote. While you can follow along with these tasks, entering data as you go, to end up with a working example, you need the requisite Siebel environment to be set up and running, which is outside the scope of this book. For example, to see a working example of Business Process Designer, you need to properly configure the Siebel server environment and email environment.
- These tasks in chapter 15 give you an overview of how the example company rolled out its application to the production environment.

Revision History

Developing and Deploying Siebel eBusiness Applications, Version 7.5.

Introducing the Business Case 1

This chapter introduces you to a fictitious company that is used as an example throughout this book. It summarizes a simple business case giving you a sense of the objectives, requirements, and the details of the company's solution design.

Read this chapter to:

- Understand NREC's business.
- Understand the business problem and scope of the solution.
- Get an idea of the sample company's implementation methodology.
- See a list of project documents used to capture the application design.
- Review the requirements that the tasks in the rest of the book are intended to implement.

The NREC example is a simple example used to highlight particular tasks. It is not intended to present a solution design or suggest a comprehensive implementation methodology. Rather, it highlights particular aspects of NREC's solution design in order to present tasks in a real-world context.

Introducing NREC

The fictitious company used as an example in this book is National Real Estate Clearinghouse (NREC). NREC is a U.S.-based company that participates in the residential real estate industry. It acts as an agent for people who are selling 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 buyers needs. Through pre-negotiated contracts with partner real estate agencies, NREC is able to close sales with fewer incurred costs than traditional real estate agencies.

NREC also solicits interest directly from buyers. Potential buyers call into a toll-free number to inquire about houses. NREC records the buyer's information and, rather than handling the leads itself, NREC passes them onto one of the agencies in its distribution network. See Figure 1 for an illustration of these interactions.

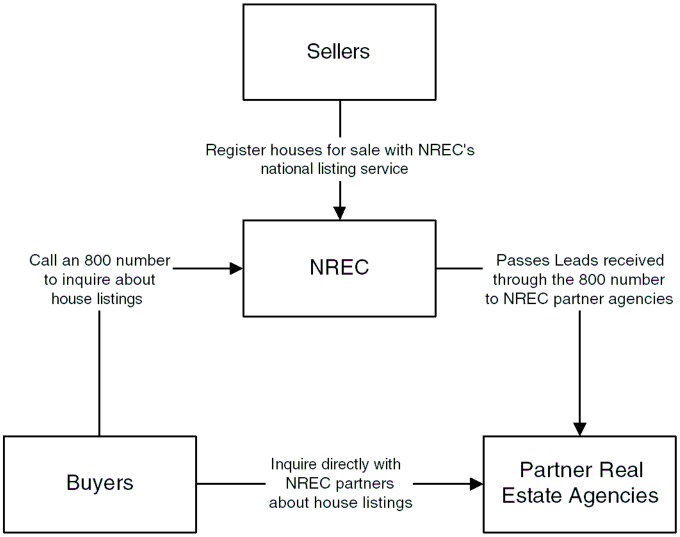

Sellers
Register houses for sale with NREC's national listing service
Call an 800 number to inquire about house listings
NREC
Passes Leads received through the 800 number to NREC partner agencies
Buyers
Inquire directly with NREC partners about house listings
Partner Real Estate Agencies

Figure 1. Key Business Processes

NREC's Business Objectives

Today NREC uses a homegrown solution to manage its pool of houses and buyer leads. It regularly faxes partner real estate agencies with information about available houses and dispatches leads in an ad-hoc manner. However the company realizes that this is an inadequate approach, which is costing it money and making it difficult to attract new independent real estate agencies to join its distribution network.

NREC has decided to address these problems by implementing a Siebel eBusiness solution. This solution will be both employee and partner facing, allowing all participants access to the key information they need to do their job. In addition, business rules within the Siebel application will streamline inefficiencies in the current process. For example, buyer leads can be automatically dispatched to partners based on agreed-upon assignment rules. In addition, agents will automatically be notified by email when an opportunity has been assigned to them.

The employee applications in NREC's implementation will be based on Siebel Sales and Siebel Partner Manager, while the partner application will be based on Siebel Partner Portal. These modules will all share a common back end, making sure that everyone always sees the most up-to-date information.

This book assumes that the configuration for the employee applications is proceeding successfully and focuses on the configuration of the partner application, Siebel Partner Portal.

NREC's User Profiles

- **NREC Application Administrators.** NREC operates the Partner Portal Web site for its partner real estate agencies and their agents. It is responsible for adding houses as they become available and adding leads that are then automatically sent to a partner agency. They also perform administration tasks such as managing partner information.
- **Real Estate Agent.** Real estate agents of the partner agencies log into the NREC Partner Portal site to manage opportunities and contact information, browse houses for sale, and track activities related to each opportunity.
- **Real Estate Agent Manager.** Managers at partner real estate agencies log into the NREC Partner Portal site to track opportunities and activities for all agents in the agency. They also log in to see houses for sale, and to track their own opportunities, contact information, and activities.

Siebel eBusiness Functionality

NREC is planning to implement the following functionality to support its business needs:

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

Access Requirements

In analyzing its requirements, NREC determined that it had the following data access needs:

- **Access through low-bandwidth Internet connections.** NREC's partner agents must be able to access data over low-bandwidth connections (such as dial-up connections) without having to install Siebel software on their personal computers.
- **Access through corporate intranet.** NREC Administrators and employees need to be able to administer the system (creating new users, assigning responsibilities, and so on) and maintain user data (houses, opportunities, contacts, and so on) through a graphical user interface connected to the corporate intranet.
- **Mobile access.** NREC employees need to be able to access and manipulate data on their laptops even without being connected to a corporate database. They must be able to synchronize with the central corporate database, when needed.

NREC will implement Siebel's Web client and Mobile Web client to meet all three requirements:

- The Web client—through which users connect to the Siebel server with their browser—meets the first and second requirements. The Siebel Web client is an HTML, Web interface that allows users to log on to Siebel applications through either the Internet or an intranet.
- The mobile Web client meets the third requirement because agents can log into Siebel eBusiness Applications and work offline in a local database that resides on the agent's computer. Users synchronize the local database with the server database when necessary. The mobile Web client provides the same HTML interface as the Web client, but the Web pages are served from the local Siebel application rather than a server.

For more information about deploying the Mobile Web client, see *Siebel Web Client Administration Guide*.

NREC's Implementation Strategy

NREC has chosen to implement Siebel business applications using a phased approach. A phased approach will help NREC minimize risk and realize a return on its investment incrementally, thereby increasing the probability of project success. NREC's end users and partners will experience the benefits of the project as early as possible. Additionally, the project team will be able to take feedback gathered during the early phases and incorporate it into the end solution.

Siebel eRoadmap Methodology

NREC will follow Siebel's eRoadmap Methodology through each one of its implementation phases. The Siebel eRoadmap Methodology provides a six-stage framework for implementing Siebel eBusiness Applications. The stages include the following:

- **Project Definition.** The project team assembles, defines the project approach and scope, and implements project management controls.
- **Discovery.** The project team refines and documents functional and technical requirements that support the business goals.
- **Design.** The project team designs a hard-copy mock-up of the solution and uses the discovery stage requirements to develop application screen flows and design layouts.
- **Configuration.** The project team configures the application, extensions, and external interfaces required to support the new system.
- **Validation.** The project team conducts a full-function test, including a user acceptance test of the application using production data.
- **Deployment.** The project team first conducts a Production Pilot that field tests and revises all aspects of the new system, user training, technical infrastructure, the network, and the help desk. The team then focuses on a successful transition from the production pilot to a full rollout.

The tasks covered in this book are primarily in the configuration and deployment stages of the eRoadmap Methodology. The scenario for the book assumes that a design team has planned out the details of the system and has provided NREC developers with design specifications they will use to configure the system.

Project Team

It is a common practice to have a dynamic project team wherein the number and specialty of people varies according to the stage of the implementation. For example, Table 1 shows the breakdown of resources needed at each stage in NREC's implementation project.

Table 1. NREC's Project Resources

Project Stages	Project Manager	Lead Business Analyst	Lead Configuration Specialist	Lead Architect	Configuration Specialists	Architect	Training Specialist	Documentation Specialist
Definition	X	X						
Discovery	X	X	X					
Design	X	X	X	X				
Configuration	X	X	X	X	X			
Validation	X	X	X	X	X	X	X	X
Deployment	X	X	X	X	X	X	X	X

Sample Project Design Documentation

The types of documents listed in this section capture the business requirements and detailed design of NREC's Siebel solution. These documents are outputs of the design stage of the Siebel eRoadmap implementation methodology. They define the scope of the solution and document the detailed specifications needed to configure the system.

Design documentation is a critical part of an implementation. Customers can submit these documents to Siebel Expert Services for a design review. If all documents are not submitted or the templates are not used to guide them through the documentation process, the customer may omit important information and reduce the value of the review.

The types of documents used to capture the system design are:

- **Entity Relationship Diagram.** This type of diagram (See Figure 2 on page 31) shows the relationships between the major entities that will be part of NREC's solution. Entities include object definitions such as business components and business objects.
- **Business Component Design.** The business component design template is used to define the properties of business components, including the user properties, joins, single-value fields, multi-value fields, and multi-value links.
- **Business Object Design.** The business object design template is used after business components are designed. The template allows you to group business components and identify links between business components. This allows you to show the parent and associated child records in the user interface.
- **Applet Design.** Applet designs are created after the implementation team decides what entities to track in their Siebel applications. Applet designs are reference documents created for each applet in the system. They comprehensively document an applet's properties, the fields that will be available on the applet, and the properties associated with each field. They should also include a mock-up or screen shot of the applet.

- **View Design.** View design documentation should show what applets are part of the view definition, identify any special visibility associated with a view, for example, a *My Team's View* or an *All* View. It should also document how the user navigates through the views using drilldowns, view tabs (third-level navigation), or the Site Map.
- **Screen Flow.** The screen design template lists all views available from each screen and indicates the default view for each screen.
- **Report Design.** Report templates track key information relating to reports. The report template includes the business component, Actuate file used, whether the report is dynamic or static, and any subreports available.
- **Workflow Policies.** Workflow Policy templates gather the key information needed to create workflow rules.
- **Assignment Rules.** Assignment Manager templates gather the key information needed to create assignment rules.

NREC's Business Requirements

By progressing through the Discovery and Design stages of Siebel's eRoadmap methodology, the project team identified and documented the detailed requirements of the system. These requirements provide the basis for the tasks covered in later chapters.

Some of these requirements are satisfied using standard Siebel functionality. Other requirements cannot be satisfied with standard functionality and are identified as gaps. Gaps require configuration work to modify the standard application to meet the specific requirements.

NOTE: As a general rule, a gap analysis should result in less than 25% of the system requiring customization. If the gap is more than 25%, it may mean that standard functionality is not being employed. It is always a good practice to engage Siebel Expert Services for a Configuration Design Review to verify gaps and required configuration changes.

Manage the List of Houses for Sale

NREC requires the ability to manage the list of houses that are currently for sale. This includes storing key characteristics, such as price, number of bedrooms, and square feet; keeping a history of appraisals; and pulling in any data about renovation history from an external file that is supplied by a third-party vendor.

Table 2 lists the detailed requirements for the House entity and lists where you can find the tasks in this book that describe how to implement the requirement.

Table 2. Requirements Summary for the House Entity

Requirement	Comment	Cross Reference to Task
Track houses for sale.	Create a new screen for Houses and add to the eChannel application.	See "Creating the Houses Screen" on page 128.
Track the features of each house, such as square feet, number of bedrooms, and so on.	Add columns to S_PROD_IN T to store additional attributes and expose them in the user interface.	See "Configuring the House Detail View" on page 114.
Allow users to enter and Edit records for houses.	Modify Internal Product business component to allow updates and add necessary controls to the Product Form Applet.	See "Configuring the Internal Product Business Component" on page 121 and "Modify Existing Product Applets to Display NREC Attributes" on page 123.
Provide users with pre-defined values to select from when entering house features.	Configure static pick lists for fields.	See "Configuring Pick Lists" on page 154.
Track and display previous appraisal information (date, amount, assessor) for each house.	Use standard 1:M table to store appraisal data and expose fields in the user interface.	See "Configuring the House Detail - Appraisals View" on page 132.
Display renovation information for each house.	Create virtual business component to store and display data from flat file. Create new view to display the data.	See "Creating a Virtual Business Component" on page 171.
Display houses to partner users based on ZIP Code.	Use Siebel Personalization to filter the list of houses based on the user's ZIP Code.	See Chapter 15, "Personalization."

Manage Opportunities

NREC needs to give its internal employees as well as its partner agents ability to manage opportunities. Opportunities are recorded for potential buyers and include contact information and information about the type of house the buyer is interested in. Opportunities that come into NREC directly are passed on to a partner agency based on location. Table 3 summarizes the requirements for the opportunity entity.

Table 3. Requirements Summary for the Opportunities Entity

Requirement	Comment	Cross Reference to Task
For each opportunity, track the house features that the buyer is looking for and buyer characteristics.	Add columns to the base opportunity table S_OPTY to store the additional attributes.	See "Configuring the House Detail View" on page 114.
Allow users to select values for house attributes—square feet, price range, and so on—from a list rather than enter them.	Add pick lists for each of the fields that need pre-defined values.	See "Static Pick Lists" on page 154.
Assign opportunities to partner agents based on ZIP Code.	Use Assignment Manager to assign opportunities to partner agencies.	See Chapter 13, "Assignment Manager."
Send an email notification to partner agents when a new opportunity is assigned to them.	Use Business Process Designer to automatically send email notification.	See Chapter 14, "Siebel Business Process Designer."

Manage Activities

NREC requires the ability to manage activities for each opportunity and contact. The standard Activity management functionality meets NREC requirements well, except for the gaps listed in Table 4.

Table 4. Requirements Summary for the Activities Entity

Requirement	Comment	Cross Reference to Task
Allow users to see the opportunity associated with each activity.	Add the opportunity to the Activity List and Form Applets.	"Exposing Fields in the User Interface" on page 107
When entering an activity, allow users to associate an opportunity to an activity.	Create a dynamic pick list that allows users to update the Opportunity field on the Account Form Applet. Constrain the values in the pick applet so it only shows the opportunities for the account associated with the activity.	"Dynamic Pick Lists" on page 157 "Constraining a Pick List" on page 161
Allow users to navigate from the Opportunity Detail view to one of the following views, depending on the sales stage: Account Detail - Activities (SCW) Account Detail - Contacts (SCW)	Configure dynamic drilldown that implements the conditions described in the business requirement.	"Creating a Dynamic Drilldown" on page 164

Manage Contacts

The standard Contact management functionality meets NREC requirements. There are no gaps identified.

Reviewing NREC's Design

The following sections describe NREC's solution design. An entity relationship diagram, such as the one shown in Figure 2, is particularly useful for describing the design. You will refer back to Figure 2 often when completing the tasks in subsequent chapters.

Data Layer

NREC will implement the entities shown in Figure 2. The diagram shows the relationships (1:1, 1:M, or M:M) between the major entities of NREC's solution. The diagram also shows the tables that will be used to store the data for each entity. For example, house information will be stored in the table called S_PROD_INT.

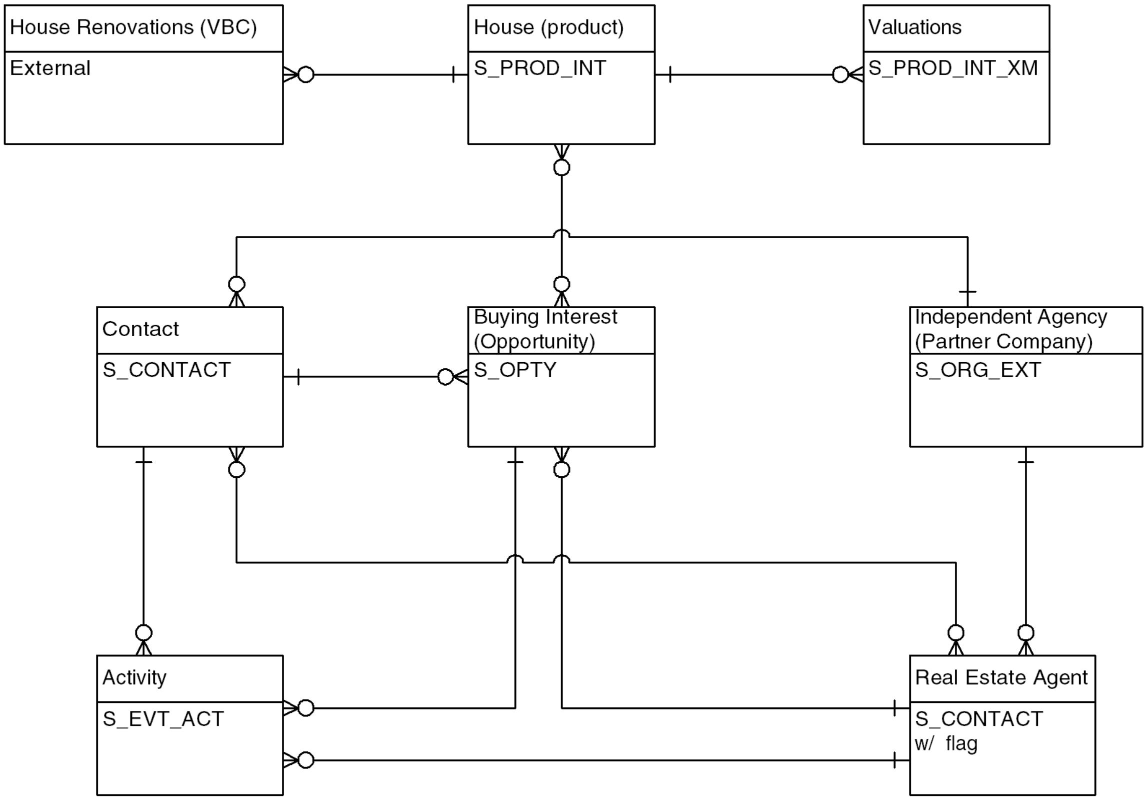


Figure 2. NREC Entity Relationship Diagram

Business Object Layer

This section defines the business objects and business components that will represent the entities that make up NREC’s solution.

Business Objects

The NREC implementation will use the following business objects:

- Opportunities
- Accounts
- Internal Products
- Contacts
- Activities

Business Components

The NREC implementation will use the following business components:

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

User Interface Layer

At the user interface layer, NREC is planning to implement five screens for the Partner Portal. Each view contains one or more applets. The Opportunities, Contacts, and Activities screens are standard in the Partner Portal application. A new screen for Houses will be created for NREC. The views that make up each screen are either master-detail views or list-form views. The applets are either list applets or form applets.

NOTE: Figure 3 on page 34 is a simple representation of part of NREC's user interface layer. There are other screens, such as Partner Management and Application Administration, that would be implemented for NREC.

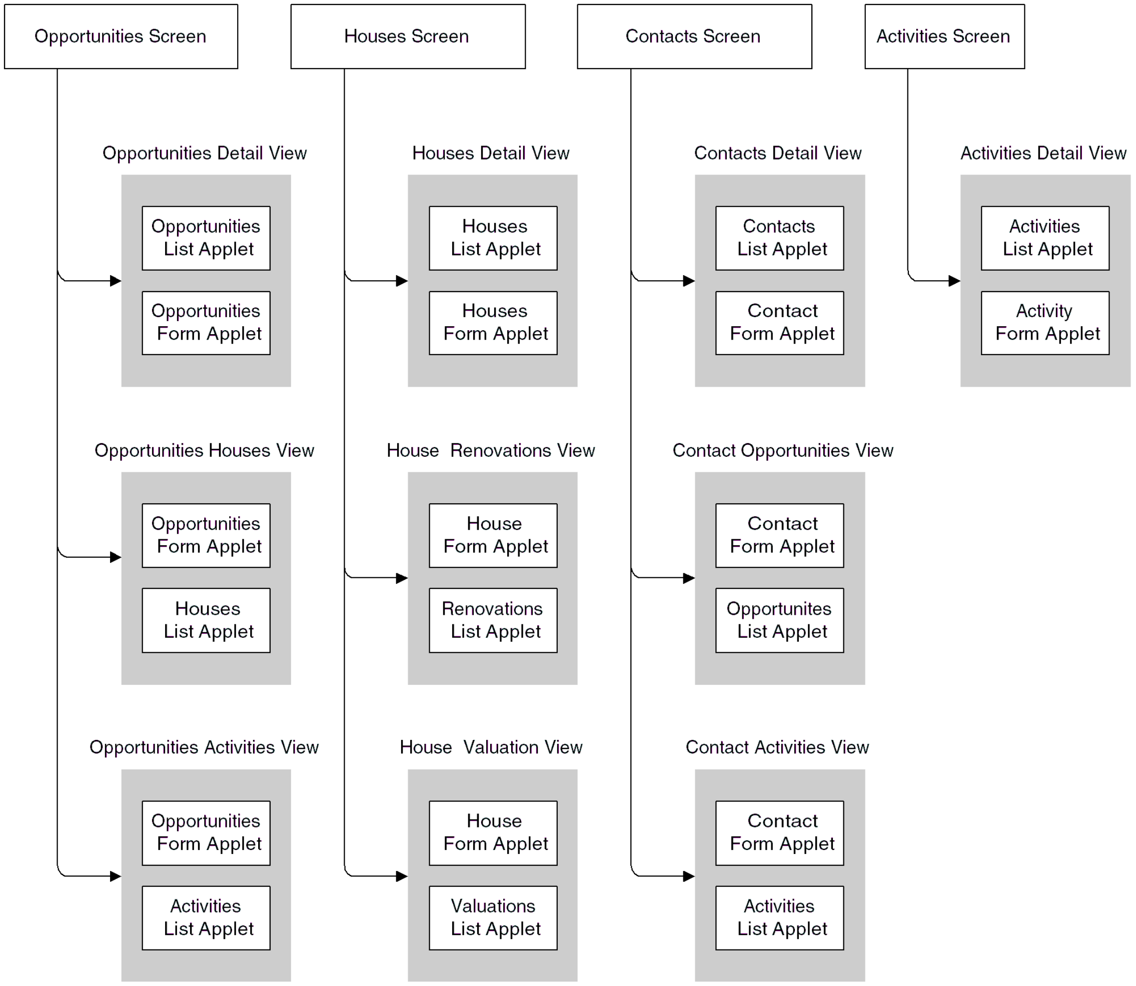


Figure 3. Screens, Views, Applets

Installing Siebel Applications 2

This chapter provides you with an overview of a typical installation, using NREC as an example. It tells you the different environments to set up to support the implementation process and then describes the hardware and software NREC installed in each environment. The example gives details, such as the machine names on which products were installed.

This chapter does not provide you with installation instructions. For each product mentioned in the following sections, there is a cross reference to where you can find detailed instructions.

Siebel Application Environments

Following the recommendation of Siebel Systems, NREC chose to install Siebel products in three separate environments—one environment for each stage in the implementation process. The three environments are development, test, and production. Each environment contains hardware, software, and data. Using this structure prevents the activities in one environment from interfering with activities in another environment. For example, configuration work in the development environment won't interfere with testing activities in the test environment.

- **Development.** The environment is used for developing customized applications and configurations. It typically consists of a server machine, several client machines, development tools, and a small set of data (repository data, seed data, and sample transactional data). The server machine in the Development environment is often called the *development server.*
- **Test.** The test environment is a separate machine (or several machines) with business data but no development tools. This environment is used to test the application with data that simulates the live production environment. A test environment is used to test customizations, patches, and version upgrades, before applying them to the production environment.
- **Production.** The production environment is the live Siebel operational environment. A Production environment might be very similar to the Test environment at the end of testing, but ultimately it consists of hundreds or even thousands of users and live business data.

Figure 4 shows the typical content of each environment.

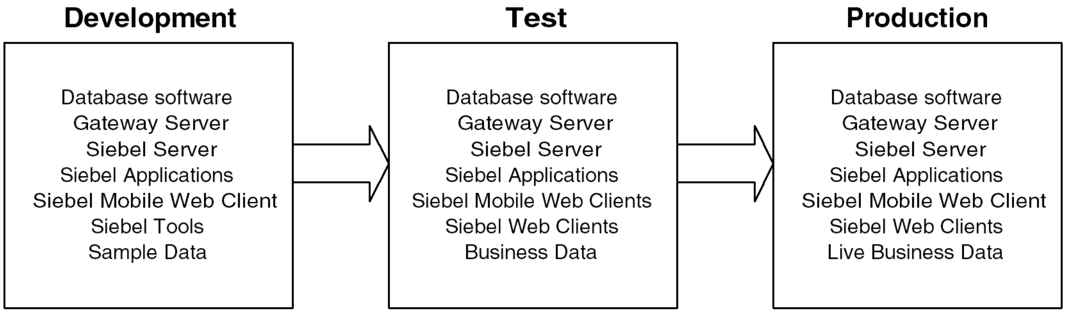


Figure 4. Software and Data in Each Environment

Each environment corresponds to a stage in the application rollout:

1 **Development**. Do your initial configuration in the Development environment.

2 **Test**. Copy the configuration data from the Development Environment to the Test environment and continue work there, using actual data for your organization. Here you will configure business rules and set up the data to simulate the production environment. After the set up is complete, testing occurs.

3 **Production or Deployment**. After you have thoroughly tested your work in the test environment, you migrate the configuration and user data to the production environment and roll out the product to all your users.

The subsequent parts of this guide correspond to these three environments and stages, each part telling you what NREC did at each stage.

The Development Environment

Table 5 lists and describes which products to install in the Development environment, on which machine the NREC installed each product, and where you can find installation instructions.

NREC installed Siebel Tools on two machines, one for each developer who will be configuring Siebel applications. You can have as many developers working simultaneously as you would like, as long you set up each developer's machine according to the instructions in Chapter 4, "Setting Up a Developer's Local Database."

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

NOTE: The NREC example assumes a Windows environment.

Table 5. Software Installed in the Development Environment

Product	Description	Machine Names	Where to Find Installation Instructions
Database software. For example, IBM DB2 Universal Database, Oracle, or SQL Server.	Third-party relational database software.	DEV_DB_server	Product documentation for the database application.
Siebel database schema	Installing and running a database script creates a database schema and populates some of the tables with "seed data."	DEV_DB_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel File System	A shared directory tree used for storing files not managed by the database software.	DEV_server	*Siebel Server Installation Guide for Microsoft Windows*

Table 5. Software Installed in the Development Environment

Product	Description	Machine Names	Where to Find Installation Instructions
Siebel Gateway Server	Software that handles connections between Siebel clients and servers.	DEV_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel Server	Application server software.	DEV_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel Dedicated Web Client	A client in which all application layers except the user interface reside on the server.	DEV_server	*Siebel Web Client Administration Guide*
Web server such as Microsoft Internet Information Server (IIS), IBM HTTP Server, or iPlanet	Web server required for the Siebel Web architecture.	DEV_server	Product documentation for the Web server.
Siebel Tools	The primary software for configuring Siebel client applications.	DEV_tools_A DEV_tools_B	*Siebel Tools Reference*
Siebel Mobile Web Client	A client in which all application layers reside on the user's personal computer.	DEV_tools_A DEV_tools_B	*Siebel Web Client Administration Guide*
Sample Database	The sample database that you can use to test, evaluate, and configure.	DEV_tools_A DEV_tools_B	*Siebel Web Client Administration Guide* and the release notes documentation for your Siebel application.
Microsoft Internet Explorer	The Web browser used to display the interface for the Siebel Web Client and the Mobile Web Client.	DEV_client_A DEV_tools_B	If this software is not pre-installed on the client computer, see the Microsoft Web site for installation instructions.

The Test Environment

Table 6 lists which products to install in the Test environment, on which machine NREC installed each product, and where you can find installation instructions.

For detailed information on supported software, see the system requirements and supported platforms documentation for your application.

NOTE: The NREC example assumes a Windows environment.

Table 6. Software Installed in the Test Environment

Product Installed by NREC	Machine Name	Where to Find Installation Instructions
Database software. For example IBM DB2 Universal Database, Oracle, or SQL Server.	TEST_DB_server	Product documentation for the database platform.
Siebel database schema	TEST_DB_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel File System	TEST_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel Gateway Server	TEST_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel Server	TEST_server	*Siebel Server Installation Guide for Microsoft Windows*
Siebel Mobile Web Client	TEST_client_A (and other client machines from which you want to test access)	*Siebel Web Client Administration Guide*
Web server such as Microsoft Internet Information Server (IIS), IBM HTTP Server, or iPlanet	TEST_server	Product documentation for the Web server.
Microsoft Internet Explorer	TEST_client_B (and other client machines from which you want to test access)	If this software is not pre-installed on the client computer, see the Microsoft Web site for installation instructions.

The Production Environment

The software you install in your Production environment is typically the same as the software installed in the Test environment (shown in Table 6 on page 42). Eventually, during rollout to users, you will copy the contents of one of the client machines to your mobile Web Client machines. Also, you will distribute a URL to your application to your Web Client users.

NREC installed the Production software on machines with names starting with PROD. For example, it installed the server-side software on PROD_server.

Getting Started with Siebel Tools 3

This chapter describes key concepts and common tasks that will get you started using Siebel Tools. You will refer back to these tasks as you work through the configuration tasks in subsequent chapters. For more information about Siebel Tools, see *Siebel Tools Reference*.

Read this chapter to:

- Understand Siebel object hierarchy
- Understand the generic object definition sequence
- Familiarize yourself with Siebel Tools
- Learn how to check in and check out projects
- Learn how to manage object definitions

Siebel Object Architecture

Siebel eBusiness architecture includes a core set of object definitions that are grouped into different layers depending on the object's function and characteristics. Additionally, there is a core set of HTML templates and style sheets that control the appearance of the user interface (see Figure 5 on page 47). You can modify object definitions and templates, or create new ones, to tailor Siebel applications to meet your organization's business requirements.

Siebel Web templates occupy the top layer of the architecture. Siebel object definitions are grouped into the middle three layers. The physical RDBMS database occupies the bottom layer. You modify Web templates and style sheets using a text editor or a raw code HTML editor. You modify Siebel object definitions using Siebel Tools.

Objects depend on objects defined in the layers below, but are insulated from each other. Changes to objects in one layer require little or no changes to the layers below. For example, you can control how data is presented by modifying objects in the user interface layer, without having to modify objects in the business logic layer. Likewise, you can change the color and other style characteristics of the user interface by modifying Web templates and style sheets, without having to modify object definitions.

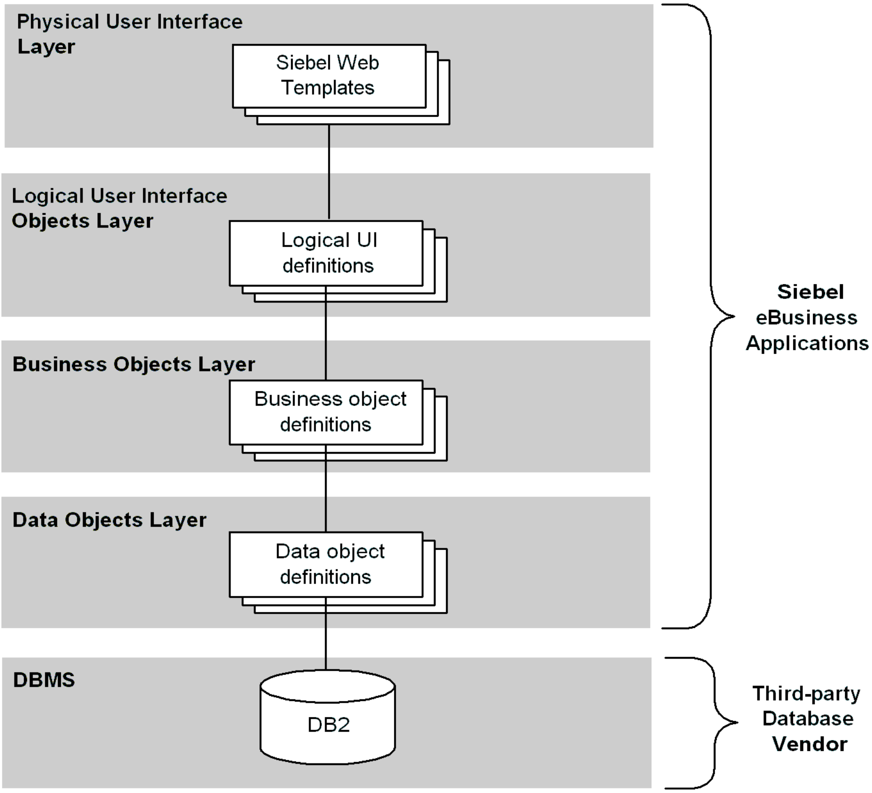

Physical User Interface Layer
Siebel Web Templates
Logical User Interface Objects Layer
Logical UI definitions
Business Objects Layer
Business object definitions
Data Objects Layer
Data object definitions
DBMS
DB2
Siebel eBusiness Applications
Third-party Database Vendor

Figure 5. Siebel Object-Based, Layered Architecture

- **Physical User Interface Layer**. This layer contains Siebel Web template files that control the style and structure of the user interface. Web templates consist of HTML tags and proprietary Siebel tags. Siebel tags are embedded within the HTML of template files and serve as placeholders for user interface objects defined in the repository, such as controls and applets. At runtime the Siebel Web Engine reads the tags, replaces them with interactive Web controls and values based on the UI object definitions, and renders the HTML that will be read by the user's browser.
- **Logical User Interface Layer.** Object definitions in this layer are the visual representation of objects in the Business Objects Layer. They define the interface presented to the user at run-time, and allow users to manipulate data. Examples of user interface objects include applets, views, and controls, such as buttons and check boxes. User interface objects also define the information that associates objects in the repository with the Siebel Web templates.
- **Business Objects Layer.** Object definitions in this layer describe individual business entities (such as Accounts, Contacts, or Activities) and the logical groupings and relationships among these entities. Business objects are based on data object definitions.
- **Data Objects Layer.** Object definitions in this layer provide a logical representation of the underlying physical database. For example, object definitions such as table, column, and index describe the physical database. These object definitions are independent of the installed RDBMS.
- **DBMS.** The third-party database management system manages the Data Objects Layer. It is not a part of the Siebel eBusiness Application.

Each layer of the Siebel object model contains several principal object types. Most of these object types contain child objects that further define the given object type.

For detailed information about the Siebel Object Architecture, see *Siebel Tools Reference*.

Understanding the Object Definition Sequence

When configuring Siebel applications it is useful to think of configuration tasks in terms of the Siebel object model hierarchy shown in Figure 5 on page 47. Sometimes you will work from the bottom up—data objects first, then business objects, and then user interface objects. Other times you will work in one layer only, modifying select objects as needed. The general sequence of tasks is shown in Figure 6.

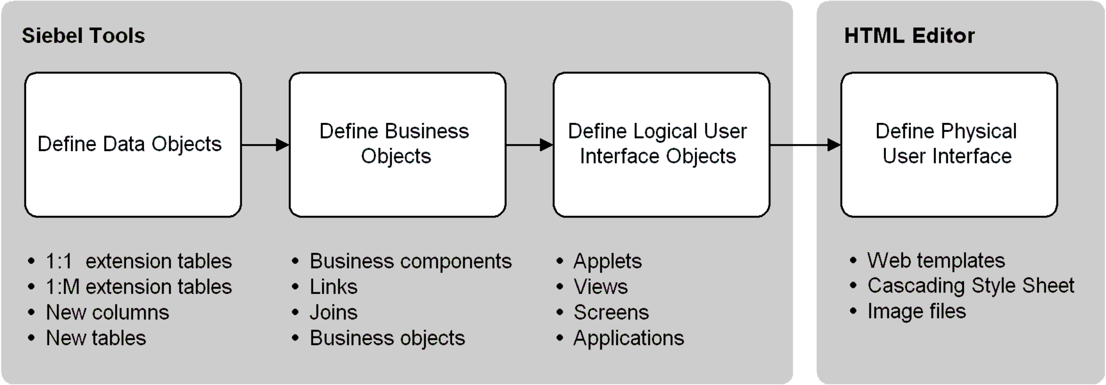


Figure 6. Development Sequence

Using Siebel Tools

Siebel Tools is a declarative software development environment that allows you to create and modify object definitions in the Siebel repository. Using Siebel Tools you can configure a standard Siebel application without modifying source code or SQL code.

NOTE: You use an HTML code editor, not Siebel Tools, to modify Web templates. However, you can also view a color-coded rendition of Web templates using the Web Template Window in Siebel Tools. You can also specify a default HTML editor to open from within the Siebel Tools application. For more information, see "Setting Tools Options" on page 56.

Windows in Siebel Tools

Most of the work you do in Siebel Tools will be through the Object Explorer Window, the Object List Editor, or the Object Properties Window. Each of these windows is described in the following sections.

Object Explorer

The Object Explorer is your starting point for working with object definitions. You use it to navigate through the Siebel object type hierarchy. It shows you the top level object types and lets you to expand them to reveal their child object types. You can also click the Flat tab to view object types listed alphabetically in a non-hierarchical structure. The Project drop-down list lets you display the object types for all locked projects or a particular project only.

Not all the top level objects appear in the Object Explorer by default. You can display or hide object types by setting your Development Tools Options. Choose View > Options and then click the Object Explorer tab. Add or remove objects from the list of visible objects. The Object Explorer is shown in Figure 7.

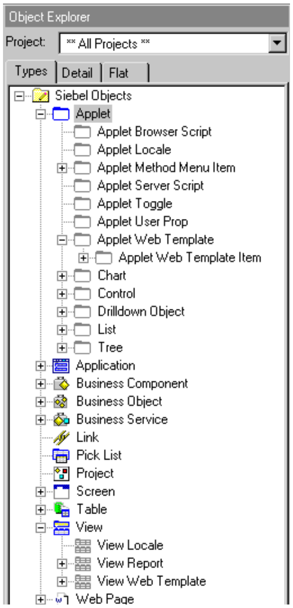


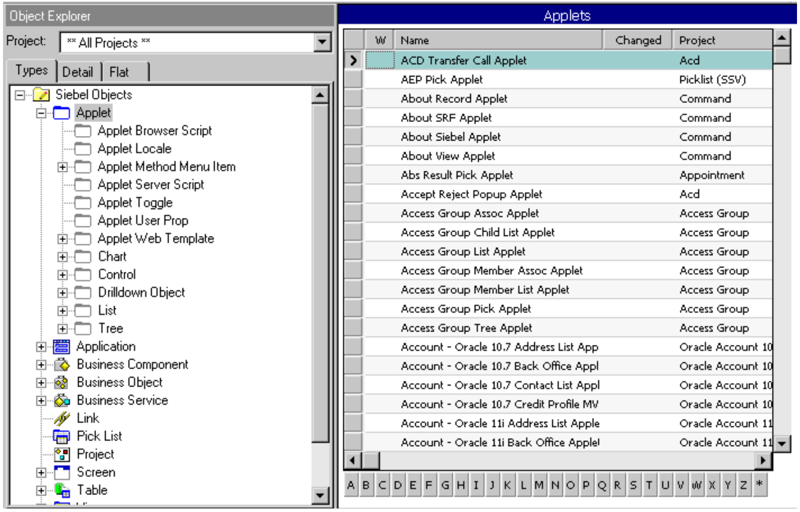
Figure 7. Object Explorer

Object List Editor

The Object List Editor appears to the right of the Object Explorer. It lists the object definitions for the object type selected in the Object Explorer. For example, in Figure 8 the Object List Editor is displaying the object definitions for the Applet object type selected in the Object Explorer. The Object List Editor is what you use to create, modify, or delete object definitions.

Object Explorer

Object List Editor



- Siebel Objects
  - Applet
    - Applet Browser Script
    - Applet Locale
    - Applet Method Menu Item
    - Applet Server Script
    - Applet Toggle
    - Applet User Prop
    - Applet Web Template
    - Chart
    - Control
    - Drilldown Object
    - List
    - Tree
  - Application
  - Business Component
  - Business Object
  - Business Service
  - Link
  - Pick List
  - Project
  - Screen
  - Table

W	Name	Changed	Project
	ACD Transfer Call Applet		Acd
	AEP Pick Applet		Picklist (SSV)
	About Record Applet		Command
	About SRF Applet		Command
	About Siebel Applet		Command
	About View Applet		Command
	Abs Result Pick Applet		Appointment
	Accept Reject Popup Applet		Acd
	Access Group Assoc Applet		Access Group
	Access Group Child List Applet		Access Group
	Access Group List Applet		Access Group
	Access Group Member Assoc Applet		Access Group
	Access Group Member List Applet		Access Group
	Access Group Pick Applet		Access Group
	Access Group Tree Applet		Access Group
	Account - Oracle 10.7 Address List App		Oracle Account 10
	Account - Oracle 10.7 Back Office Appl		Oracle Account 10
	Account - Oracle 10.7 Contact List Appl		Oracle Account 10
	Account - Oracle 10.7 Credit Profile MV		Oracle Account 10
	Account - Oracle 11i Address List Apple		Oracle Account 11
	Account - Oracle 11i Back Office Applet		Oracle Account 11

Figure 8. Object List Editor

Object Properties Window

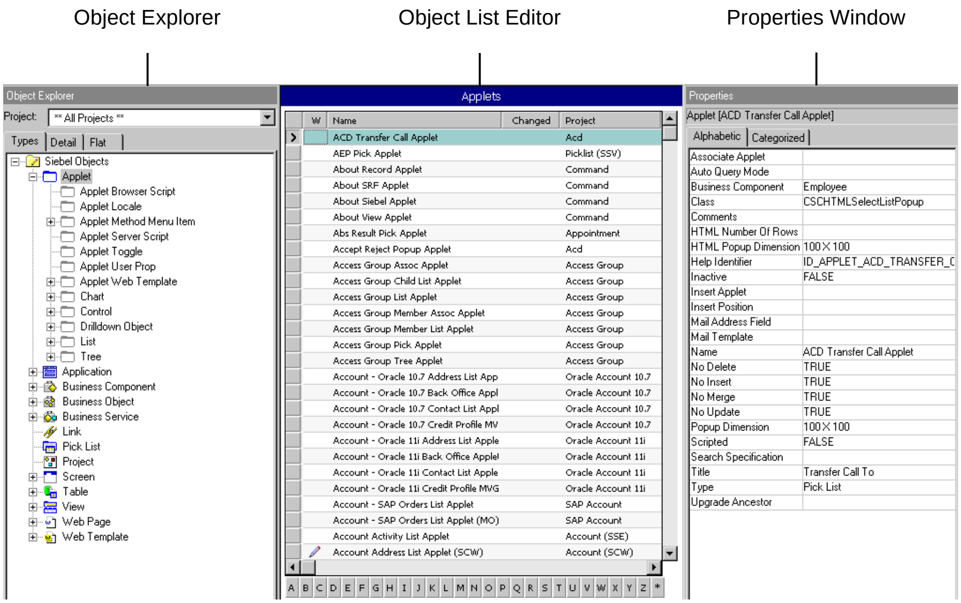
The Properties Window displays the property values for the object definition selected in the Object List Editor. It provides you with all property definitions at a glance. The property window is often more convenient than the Object List Editor when you want to view or edit objects that have many properties. You can show or hide the properties window by doing the following:

To show the Properties window

- Choose View > Windows > Properties window.

To hide the Properties window

- In the Properties window shown in Figure 9, right-click and then choose Hide.



W	Name	Changed	Project
	ACD Transfer Call Applet		Acd
	AEP Pick Applet		Picklist (SSV)
	About Record Applet		Command
	About SRF Applet		Command
	About Siebel Applet		Command
	About View Applet		Command
	Abs Result Pick Applet		Appointment
	Accept Reject Popup Applet		Acd
	Access Group Assoc Applet		Access Group
	Access Group Child List Applet		Access Group
	Access Group List Applet		Access Group
	Access Group Member Assoc Applet		Access Group
	Access Group Member List Applet		Access Group
	Access Group Pick Applet		Access Group
	Access Group Tree Applet		Access Group
	Account - Oracle 10.7 Address List App		Oracle Account 10.7
	Account - Oracle 10.7 Back Office Appl		Oracle Account 10.7
	Account - Oracle 10.7 Contact List Appl		Oracle Account 10.7
	Account - Oracle 10.7 Credit Profile MV		Oracle Account 10.7
	Account - Oracle 11i Address List Apple		Oracle Account 11i
	Account - Oracle 11i Back Office Applel		Oracle Account 11i
	Account - Oracle 11i Contact List Apple		Oracle Account 11i
	Account - Oracle 11i Credit Profile MVG		Oracle Account 11i
	Account - SAP Orders List Applet		SAP Account
	Account - SAP Orders List Applet (MO)		SAP Account
	Account Activity List Applet		Account (SSE)
✎	Account Address List Applet (SCW)		Account (SCW)

Associate Applet	
Auto Query Mode	
Business Component	Employee
Class	CSCHTMLSelectListPopup
Comments	
HTML Number Of Rows	
HTML Popup Dimension	100 X 100
Help Identifier	ID_APPLET_ACD_TRANSFER_C
Inactive	FALSE
Insert Applet	
Insert Position	
Mail Address Field	
Mail Template	
Name	ACD Transfer Call Applet
No Delete	TRUE
No Insert	TRUE
No Merge	TRUE
No Update	TRUE
Popup Dimension	100 X 100
Scripted	FALSE
Search Specification	
Title	Transfer Call To
Type	Pick List
Upgrade Ancestor	

Figure 9. Properties Window

Wizards

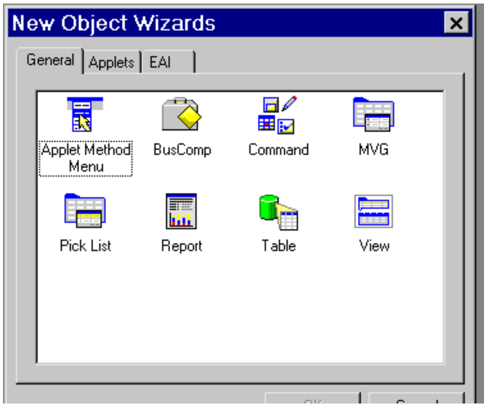
Siebel Tools includes many wizards that guide you through the process of creating new object definitions. Wizards assist you by prompting you for the key information necessary to define a particular object, such as an applet or a view. Then, based on the information entered, it creates the object definition and any related object definitions.

You do not have to use wizards to create objects. You can also create objects by adding records in the Object List Editor. But this requires thorough knowledge of the Siebel Object Model hierarchy. This guide uses wizards wherever possible.

To open a New Object Wizard

1 Choose File > New Object.

  The New Object Wizard dialog box appears.



2 Choose the wizard for the type of object you want to create.

  The selected wizard opens.

Web Layout Editor

The Web layout editor allows you to drag user interface objects such as applets and controls and drop them into place holders in Siebel Web templates. You will use the Web Layout Editor to design the layout of applets and views. You access the Web Layout Editor by selecting any of the following objects, right-clicking, and then selecting Edit Web Layout:

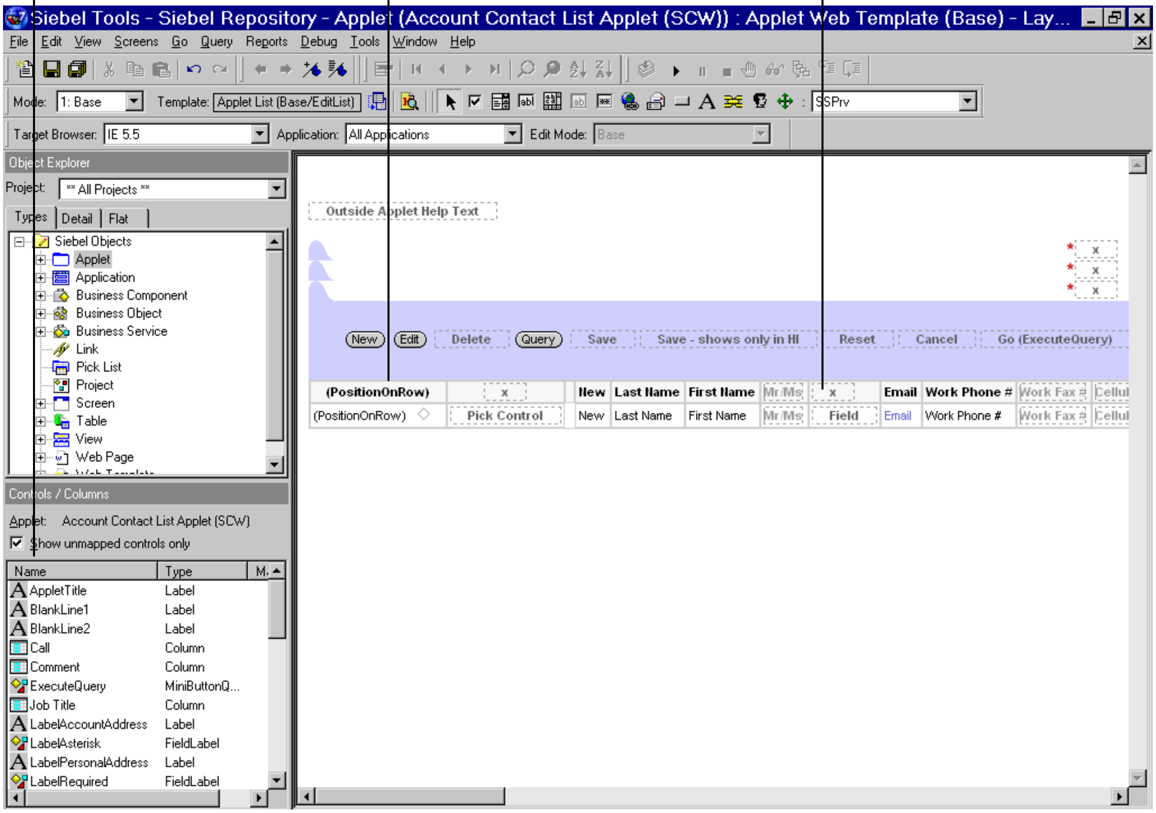
- Views and View Web templates
- Applets and Applet Web templates
- Web Templates

The Web Layout Editor is shown in Figure 10.

Drag controls listed in this window to empty placeholders on the Web template.

Example of a control mapped to a placeholder in the Web template

Empty placeholder



Siebel Tools - Siebel Repository - Applet (Account Contact List Applet (SCW)) : Applet Web Template (Base) - Lay...

Mode: 1: Base

Template: Applet List (Base/EditList)

SSPrv

Target Browser: IE 5.5

Application: All Applications

Edit Mode: Base

Object Explorer

Project: ** All Projects **

Controls / Columns

Applet: Account Contact List Applet (SCW)

☑ Show unmapped controls only

Name	Type	M.
AppletTitle	Label	
BlankLine1	Label	
BlankLine2	Label	
Call	Column	
Comment	Column	
ExecuteQuery	MiniButtonQ...	
Job Title	Column	
LabelAccountAddress	Label	
LabelAsterisk	FieldLabel	
LabelPersonalAddress	Label	
LabelRequired	FieldLabel	

(PositionOnRow)	x	New	Last Name	First Name	Mr/Ms	x	Email	Work Phone #	Work Fax #	Cellul
(PositionOnRow) ◇	Pick Control	New	Last Name	First Name	Mr/Ms	Field	Email	Work Phone #	Work Fax #	Cellul

Figure 10. Applet Web Layout Editor

Setting Tools Options

There are several options in Siebel Tools that you can set to facilitate your configuration work. These options are available by choosing View Options, and then choosing the tab for the options you want to configure. Some of the options are described below.

- **Language Setting Tab**. These settings define your working language in Siebel Tools. Objects can have locale-specific attributes. For example, text strings such as an applet title can be defined in several languages.

  The Tools Language Mode determines the locale-specific data that is:

  - Displayed in the Tools interface
  - Available to edit
  - Compiled to the .srf file
  - Transferred to and from the server during the check in/check out process

  You can see what your current Tools Language Mode is in the status bar located in the lower right corner of the Siebel Tools interface.

  The Enable Language Override check box allows you to edit locale-specific attributes. When you select this check box you can switch between the following two modes:

  - **Base.** When working in this mode, changes made to locale-specific attributes are stored in the base table. The changes apply to all languages.
  - **Language Override.** When working in this mode, changes are stored in a child locale table. That is, the attribute is overwritten for the current language.

  When the Enable Language Override check box is not selected, you can work in base mode only.

  For more information about language modes, see *Siebel Tools Reference*.

- **Check In/Out Tab.** This setting defines the development server database and your local client database to be used in the check in/check out process.

- **Web Template Editor Tab.** Use this setting to define a default Web template editor. When working in the Web Layout Editor you can click the Edit Template button on the toolbar to open the Web template using a default editor.

The Configuration Process

The typical process for configuration applications using Siebel Tools can be broken down into the following basic steps:

- "Checking Out Projects from the Server"
- "Making Configuring Changes" on page 60
- "Compiling Projects" on page 60
- "Testing Changes" on page 63
- "Checking In Projects" on page 65

Checking Out Projects from the Server

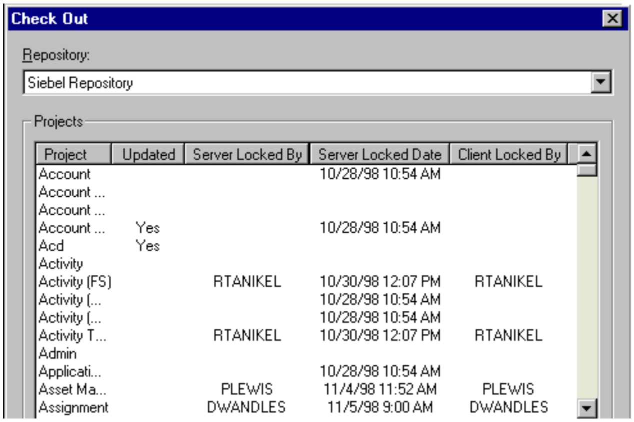
You typically do your configuration work in a local database, checking out object definitions (projects) from the development server as needed.

See "About the Local Development Environment" on page 73 for an overview of the development environment.

To check out a project

1 In Siebel Tools, choose Tools > Check Out.

The Check Out dialog box appears.



Check Out

Repository: Siebel Repository

Projects

Project	Updated	Server Locked By	Server Locked Date	Client Locked By
Account			10/28/98 10:54 AM	
Account ...				
Account ...				
Account ...	Yes		10/28/98 10:54 AM	
Acd	Yes			
Activity				
Activity (FS)		RTANIKEL	10/30/98 12:07 PM	RTANIKEL
Activity (...			10/28/98 10:54 AM	
Activity (...			10/28/98 10:54 AM	
Activity T...		RTANIKEL	10/30/98 12:07 PM	RTANIKEL
Admin				
Applicati...			10/28/98 10:54 AM	
Asset Ma...		PLEWIS	11/4/98 11:52 AM	PLEWIS
Assignment		DWANDLES	11/5/98 9:00 AM	DWANDLES

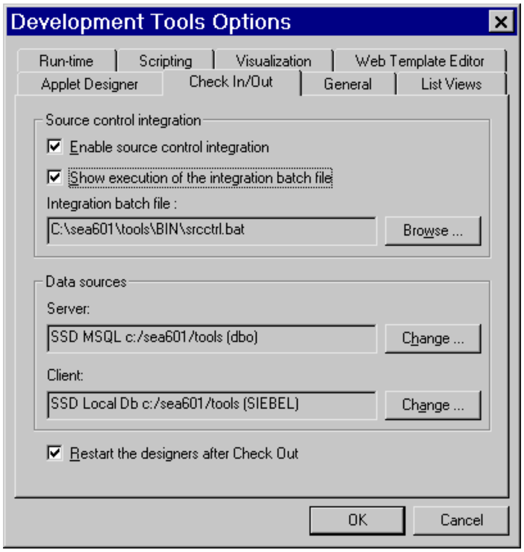
2 Make sure the correct repository is selected.

The default value is Siebel Repository.

3 Select the project or projects whose objects you want to check out.

4 Click the Options button.

The Development Tools Options window appears, with the Check In/Out tab selected.



Development Tools Options

Source control integration

☑ Enable source control integration

☑ Show execution of the integration batch file

Integration batch file : C:\sea601\tools\BIN\srcctrl.bat

Data sources

Server: SSD MSQL c:/sea601/tools (dbo)

Client: SSD Local Db c:/sea601/tools (SIEBEL)

☑ Restart the designers after Check Out

5 In the Development Tools Options window, make sure the Server and Client data sources are specified correctly.

6 Close the Development Tools Options window.

The Check Out dialog box appears.

7 In the Check Out dialog box, click Check Out.

Caution: Password encryption must be disabled when checking projects in and out. You disable password encryption in the Tools configuration file (tools.cfg).

Making Configuring Changes

Once a you check out a project, the project remains locked in the server database and in your local database. This prevents other developers from checking out the project. During this time, you can make configuration changes using Siebel Tools, such as modifying user interface objects, extending business logic, and extending the database.

NOTE: You can also lock projects locally, without checking them out from a server database. See "Locking Projects in Your Local Repository" on page 67. This is how the procedures in subsequent chapters are written.

Compiling Projects

After you have made your configuration changes, you must compile them into a Siebel repository file. Until you do so, your Mobile Web Client application that reads the repository file will not reflect the changes you have made.

There are various options for compiling the repository. You can compile at the project level—selected projects, all locked projects, all projects—or you can compile individual objects. Compiling individual objects is faster, but you must remember to do it for each object you modify.

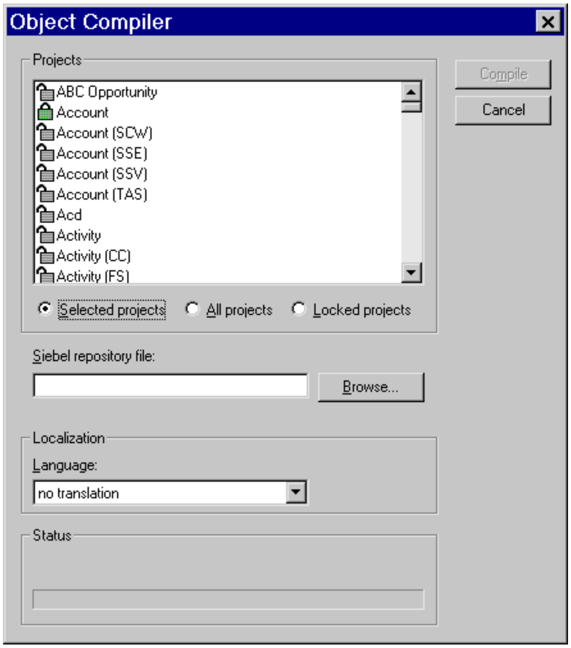
To compile projects

1 Exit Web client applications that are running on the .srf file to which you want to compile.

While running, a client application maintains a lock on the .srf file.

2 In Siebel Tools, choose Tools > Compile Projects.

The Object Compiler window appears.



Object Compiler

Projects

ABC Opportunity
Account
Account (SCW)
Account (SSE)
Account (SSV)
Account (TAS)
Acd
Activity
Activity (CC)
Activity (FS)

● Selected projects ○ All projects ○ Locked projects

Siebel repository file:

Localization

Language: no translation

Status

3 In the Object Compiler window, select whether you want to compile selected projects, locked projects, or all projects.

Compiling only the locked projects (those currently checked out) is faster than compiling all projects. Compiling locked projects is also often easier than selecting individual projects from the list.

4 In the Object Compiler window, click the Browse button to select the .srf file you want to compile.

The .srf file for your application is in the objects subdirectory of your Siebel application client directory. For example, a typical path is to the Mobile Web client installed for testing on a developer's machine is:

- `D:\sea702\client\OBJECTS\siebel.srf`

NOTE: Do not try to compile to the objects subdirectory of Tools. The .srf file in this directory is locked because the Siebel Tools program itself reads from it constantly as it runs. If you attempt to compile to this filename and path, you will receive an error and be prevented from compiling.

NOTE: Be sure to make a backup copy of the repository file, siebel.srf.

---

5 Click the Compile button in the Object Explorer window.

After compilation is successful, the .srf file you specified contains all the configuration changes you made.

To compile individual objects

1 In the Object Explorer, select the object type you want to compile.

You can only select top level objects such as, applets, views, business components, or business objects. You cannot compile child-level objects.

2 In the Object List Editor, select the object or objects you want to compile.

The objects can belong to different projects.

3 Right-click the mouse and choose Compile.

The Compile dialog box appears.

4 In the Compile dialog box, select the repository file (.srf) to which you want to direct your changes.

5 Click Compile.

After compilation is successful, the .srf file you specified contains all the configuration changes that you made.

Testing Changes

Once you have compiled objects into a new repository file, you can see the results of your changes in the client application. Typically developers will set up a Mobile Web client on their machine to use for testing and will point the Web client to the repository file that has been recompiled with configuration changes.

NOTE: Make sure you have the correct .srf file—the one you compiled your changes to—is defined for the application you want to test. You can check this by verifying the value of the repository file parameter in the application's configuration file.

Registering New Views in the Application

When you add a new view to an application, you must make it visible in the client application before you can test it. You do this by registering the new view in the application using the View Administration view and then associating it to a responsibility.

If you are working with existing views, these steps are not necessary. The views are already registered in the application.

To register views in the application

1 From the application-level menu, choose View > Site Map > Application Administration > Views.

The View list appears.

2 Click the menu button and then select New Record.

3 Enter the view name.

The view name you enter must match the name property defined for the view in Siebel Tools.

4 Enter a description and select whether you want the view to be available for local access in the Mobile Web client.

5 Click Save.

To associate views with a responsibility

1. From the application-level menu, choose View > Site Map > Application Administration > Responsibilities.

   The Responsibilities list appears.

2. In the Responsibilities list, select the Responsibility to which you want to associate the view.

   NOTE: You cannot edit the responsibilities that are part of the seed data that ships with the product. You must create a new responsibility to be able to associate new views to it. You can copy one of the sample responsibilities, such as Siebel System Administrator, and then customize it for your purposes.

3. In the View list, click the menu button and then select New Record.

4. From the Add Views dialog box select a view.

   The view is associated with the responsibility. When a user with the responsibility logs into the application, the view will appear in the user interface.

Verifying Your Changes

After you have made any new views available in the application, you can log on to your Web Client application to see the results of your configuration changes.

To verify your changes

1. Start your client application.
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 the applet appears where it should.
3. Note any differences between what you see and the expected appearance or behavior.
4. If you find problems, repeat the cycle:
   - **a** Return to Siebel Tools to make configuration changes necessary to fix the problem.
   - **b** Compile the project.
   - **c** Test the change.

Checking In Projects

After you confirm completion of your configuration changes, you need to check the project back into the development server. Here are some guidelines for checking in projects:

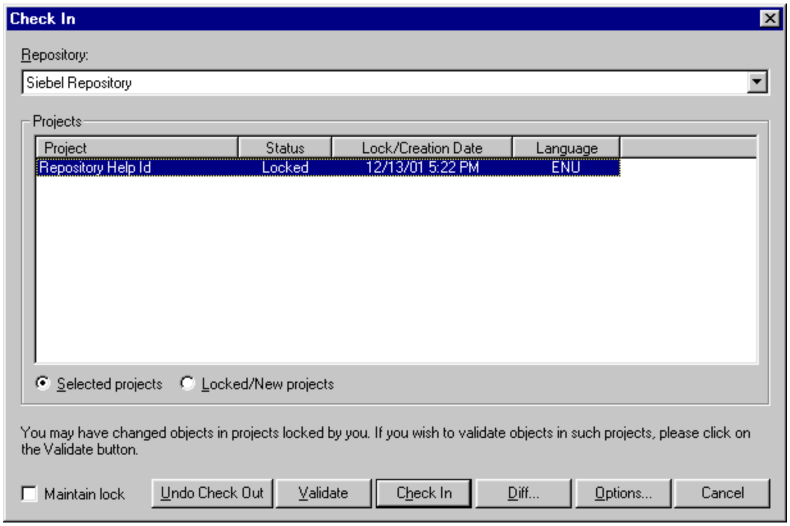
- Test objects before you check them in.
- Validate projects using the Validate button in the Check-In dialog box.
- Check in all dependent projects at the same time to make sure the configuration on the server remains consistent.
- Keep in mind that the work of other developers may depend on the objects you are configuring. In some cases, this may require you to check in projects before your work is complete because other developers may be dependent on a feature you have added to your project.

See "About the Local Development Environment" on page 73 for overview information about the development environment.

To check in changes

1 Choose Tools > Check In.

The Check In dialog box appears.



Repository: Siebel Repository

Projects

Project	Status	Lock/Creation Date	Language
Repository Help Id	Locked	12/13/01 5:22 PM	ENU

● Selected projects ○ Locked/New projects

You may have changed objects in projects locked by you. If you wish to validate objects in such projects, please click on the Validate button.

□ Maintain lock

2 Click the Options button.

The Development Tools Options dialog box appears.

3 As you did during check-out, make sure the server and client Data Sources are pointing to the correct databases.

4 Close the Development Tools Options dialog box.

5 Select the Locked/New Projects radio button.

6 Click Check In.

Other Key Tasks for Managing Object Definitions

This section covers other tasks that are useful for working with object definitions.

Viewing Project Differences Before Check In

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 committing your changes to the server.

To view project differences

- Click the Diff button in the Check In dialog box.

  All checked out projects that have been modified since the last check out are displayed.

Locking Projects in Your Local Repository

When you are prototyping ideas or are making changes to object definitions that you plan to discard, it is good practice to lock the project locally rather than checking it out from the server. This way the project remains unlocked on the server, allowing other developers to check out the project from the server, while still allowing you to modify and test changes in your local repository.

To lock a project locally

- In the Tools Object List Editor select the project or other object type you want to modify, then choose Tools > Lock Project.

  Tools locks the project in your local repository allowing you to make changes. It does not lock the project on the server database.

NOTE: Do not lock a project locally on the server repository.

Exporting Object Definitions

You can export object definitions making them available to import into another repository. This is useful when you are working in a multiple repository development environment and want to share object definitions.

To export object definitions

1. In the Object Explorer, select the object type you want to export.
2. In the Object List Editor, select the object or objects you want to export.
3. Choose Tools > Add to Archive.

   The Export to Archive File dialog box appears.

4. If you want to include additional objects in the export file:

   a. Do not click save.

   b. Return to the Object Explorer and select the object type you want to export.

   c. Repeat Step 2 and Step 3.

   NOTE: You can remove objects from the Export to Archive dialog box by selecting the object and pressing the Delete key on your keyboard.

5. Enter the path to the directory where you want to store the archive file.
6. Click Save.

   The object or objects are saved as files with a .sif extension.

   NOTE: This file overwrites any existing .sif files with the same name. You cannot add objects to an existing export file.

Importing Object Definitions

You can import object definitions from an archive file into your local repository. To import object definitions you need to have access to the archive files (.sif) and the target repository needs to be the active repository on your local machine.

NOTE: A local or server database can contain more than one repository. However, Siebel Tools only shows information from only one repository at any time. This is the active repository. You can change the active repository by selecting the Repositories object type in the Object Explorer window or by choosing File > Open Repository.

To import archive files

1 Lock all projects that will contain the objects you are importing.

NOTE: If you do not lock all projects affected by the import, the Import Wizard will prompt you to lock them on the second dialog box of the Wizard.

2 Choose Tools > Import From Archive.

The Select Archive File to Import dialog box appears.

3 Find the object archive file (.sif) that you want to import.

4 Click open.

The Import Wizard opens and the object archive file opens.

5 Select the conflict resolution option that you want to use.

6. Click next.

   The Import Wizard checks for differences between the object definitions in the current repository and the object definitions in the archive file.

7. If there are no differences, the object definition is imported.

   If there are differences, the Import Wizard displays the following:

   - **Conflicting Objects**. Shows the object types in which differences were found.
   - **Object Differences.** Shows whether the object exists in the archive file only, in the current repository, or both, and shows the conflict resolution action that has been defined. You can change the action if necessary.
   - **Attribute Differences.** Shows the object attributes that have different values. If the action for the object difference is merge, you can update the value in the Resolution column by selecting the attribute, right-clicking, and then choosing Repository or File.

Setting Up a Developer's Local Database 4

After you have installed software as described in Chapter 2, "Installing Siebel Applications," but before developers can do any configuration work, each developer must be set up with a local database. The process for setting up a local database includes tasks typically performed by a system administrator and tasks performed by each developer. For example, an administrator creates database users and runs Siebel Remote server tasks to generate local databases, then each developer initializes their local database by downloading repository data from the server. See Table 7 for a list of tasks and where in the application they are performed.

NOTE: This chapter gives you an overview of the sequence of tasks that need to be performed to set up a developer's local database. It is not necessary to complete these tasks to be able to follow the configuration examples in Chapter 5 through Chapter 9. In those chapters you can log on to the SAMPLE database using the SADMIN/SADMIN user name and password.

Table 7. Tasks for Setting Up a Developer's Local Database

Task	Where Performed
"Setting Up Database Users" on page 75	RDBMS Software
"Creating Positions" on page 76	User Administration Screen
"Associating Responsibilities" on page 77	User Administration Screen
"Setting Up Developers as Siebel Employees" on page 78	User Administration Screen
"Setting Up Developers as Mobile Web Clients" on page 80	Siebel Remote Administration Screen
"Generating a New Database Template" on page 82	Server Administration Screen

Table 7. Tasks for Setting Up a Developer’s Local Database

Task	Where Performed
“Extracting the Local Database” on page 84	Server Administration Screen
“Initializing Each Developer’s Local Database” on page 86	Siebel Tools
“Performing a Full “Get” Process” on page 87	Siebel Tools

About the Local Development Environment

Developers working with Siebel Tools work on a *local database*. A local database is a snapshot of the server database that is stored on a user's machine. The data in each local database is a subset of the server database, as determined by each user's visibility rules. The development environment is illustrated in Figure 11.

Developers check out object definitions from the server database, then after making and testing changes on their local machines, they check them back into the server database.

- **Check out.** Copies selected projects from the server database to the local database and locks the projects on the server. Locking the project on the server prevents other developers from checking it out.
- **Check in.** Copies selected projects (including any changes made by the developer) from the local database back to the server database and releases the lock on the server.

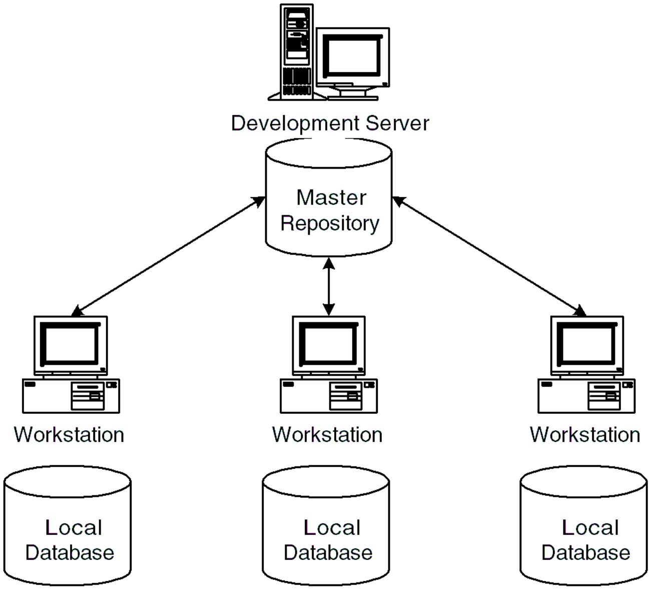


Figure 11. Development Environment

Working on a local database using check in and check out, rather than directly on the server database, provides the following benefits:

- Allows you to roll back unwanted changes without overwriting the work of other developers. Working on a local database gives you the option of not checking in changes to the server. 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.
- Allows you to work remotely because you are not required to be connected to the development server.

NOTE: Siebel Web templates are not included in the check in and check out mechanism used to control object definitions. You must manage Siebel Web template files using a third-party source control application or a manual process. See "Moving Modified Web Templates and Related Files" on page 225.

See "Checking Out Projects from the Server" on page 57 and "Checking In Projects" on page 65 for information on how to perform check in and check out.

Setting Up Database Users

Each user accessing a local database must be set up with a database user account. Work with your database administrator to add database accounts on the appropriate database and then add these accounts to the group SSE_ROLE. The exact steps for adding users and placing them in this role group depend on the database software you are using.

The username used for the account is the name that will appear in Siebel Tools when a developer locks a project. Make sure the username clearly identifies the developer. Establish a convention in choosing usernames, such as the first initial plus the last name.

For more information about setting up database users, see the product documentation for the specific database application your organization uses.

Creating Positions

Positions represent an actual job position in your organization. Positions determine which records are available to each user. A position can be any appropriate job title, such as the developer’s actual job title or simply the word “Developer.” For example, assume NREC set up positions for two developers, called Developer 1 and Developer 2.

There are several positions that are part of the seed data that ships with the Siebel product. However, none fit the role of the developer, so you will need to create a new position for the users on your development team.

For more information about positions, see *Applications Administration Guide*.

NOTE: The position information is used by the Siebel Remote’s routing rules and may affect the outcome of a database extraction. See *Siebel Remote and Replication Manager Administration Guide* for detailed information.

Associating Responsibilities

Responsibilities determine which views users have access to. For developers, the responsibility should include access to all views necessary to perform testing and development tasks. The easiest choice is to assign developers the Siebel Administrator responsibility. It is already defined as seed data and it provides access to *all* views.

Because the Siebel Administrator responsibility is already defined as part of seed data, you do not need to create a new one. You can associate the users to this responsibility when you register the developer as an employee in "Setting Up Developers as Siebel Employees."

An alternative to using a predefined responsibility would be to create your own.

For more information about responsibilities, see *Applications Administration Guide*.

Setting Up Developers as Siebel Employees

After developers have a database account, and the necessary positions and responsibilities are defined, you can set up each developer as an employee in your Siebel application.

To set up an employee user

1. Log on to the development server with a username and password that have administrator privileges, such as SADMIN/SADMIN.
2. From the application-level menu, choose View > Site Map > User Administration > Employees.

   The Employee list appears.

3 In the Employees list, add a new record and complete the following fields:

Field	Example Value	Comments
First Name	Noel	The developer’s first name.
Last Name	Adams	The developer’s last name.
User ID	NADAMS	The username to be used for logging on to Siebel Tools (and for identifying which user has checked out a project). The login name must match the user’s database account name. Enter all login names in uppercase letters for compatibility across database systems.
Password	NREC	The user’s password to the application is the same as the user’s password on the database account. This field is read-only.
Position	Developer_1	Positions determine which records are visible to users.
Responsibility	sadmin	Responsibilities determine which views a person has access to. Developers typically need access to all views for the purpose of testing and entering test data. For this reason, NREC chose to associate the Siebel Administrator responsibility to each developer. An alternative would be to create a new responsibility that only allows developers access to a particular set of views.

NOTE: The employee information for each developer resides on the development server. When you create a local database for the developer, this information is also included in each local database.

Setting Up Developers as Mobile Web Clients

After setting up developers as Siebel Employees, you must register developers as mobile Web clients. This provides Siebel Remote with the information it will need to create a local database file for the developer and populate it with the appropriate data.

To register a developer as a new mobile Web client

1 Log on to your Siebel application using an account (for example, SADMIN/SADMIN) with access to the Siebel Remote Administration screen.

> **NOTE:** Although Siebel Remote is not used for the check-in and check-out of projects, you must set up developers as mobile Web clients because Siebel Remote uses this information to retrieve the local database during initialization.

2 From the application-level menu, choose View > Site Map > Siebel Remote Administration > Mobile Clients.

The Mobile Client view appears with mobile Web clients for the Parent Server displayed.

3 In the Parent Server list, select the appropriate parent node.

4 In the Mobile Clients list, click New.

5 Complete the following fields:

Field	Example Value	Description
Mobile Client	NADAMS	Enter the mobile Web client name in uppercase letters, with a maximum of eight characters. It is recommended to use the User ID. The name cannot include spaces, periods, or other invalid characters for a DOS filename. It can contain only single-byte, alphanumeric characters, and the underscore (_) and hyphen (-) characters.
User ID	NADAMS	This value is used to log into the local database during initialization.
Routing Model	MOBILE CLIENT - EXTRACT ONLY	This value identifies the routing model to which the mobile Web client belongs. The MOBILE CLIENT - EXTRACT ONLY model does not allow synchronization. It is used simply for a snapshot of data, which is appropriate in the case of a developer.

For more detailed information about mobile Web clients, routing models, language preference, and Siebel Remote, see *Siebel Remote and Replication Manager Administration Guide*.

Generating a New Database Template

A database template is a cached representation of Siebel tables and indexes that are stored in a database file (DBF). You create a database template on the server, in this case the development server. Siebel Remote will use the template to create the local database for each developer. You create the database template by running the Generate New Database server component (GenNewDb).

NOTE: Before you can run server components, the component group to which it belongs, must be enabled at the enterprise-level. If you enabled Siebel Remote during the server installation process, this is already done for the Remote component group. However, if you did not select Siebel Remote during the installation process, you will need to enable it manually and then synchronize batch components. See *Siebel Server Administration Guide* for instructions on how to do this.

For detailed information about Siebel Remote, see *Siebel Remote and Replication Manager Administration Guide*.

Running the Generate New Database Server Component

The following procedure explains how to generate a new database template.

To generate a new database template (GenNewDb)

1. From the application level menu, choose View > Site Map > Server Administration > Enterprise Operations.
2. Click the Component Requests tab.
3. Click New.
4. In the Component/Job field, select Generate New Database.
5. Specify the server name in the Server field.

6. In the Component Request Parameters list, click New.
7. In the Name field, select SQL Anywhere Database from the pick list.

   The default value is typically sse_utf8.dbf and appears automatically.
8. From the Components Request menu, select Submit request.

   A new database file is generated. This may take a few minutes.

Extracting the Local Database

The Database Extract server component extracts data from the server database for each mobile user and temporarily store it in a compressed file. The data in this file is used to populate the user's local database during the database initialization process.

NOTE: The mobile user must have a valid position in the organization's reporting hierarchy for the database extract to be successful. For more information about positions and organizations, see *Applications Administration Guide*.

For detailed information about Siebel Remote, see *Siebel Remote and Replication Manager Administration Guide*.

To run a database extract for a mobile Web client

1. From the application-level menu, choose View > Site Map > Server Administration > Enterprise Operations.
2. Click the Component Requests tab, and click New.
3. In the Component/Job field, select Database Extract from the pick list.
4. In the Component Request Parameters list, click New and add the necessary parameters.

   The required parameter for Database Extract is Client Name.

   The value for the Client Name parameter is the name of the mobile Web client.
5. From the Component Requests menu, select Submit request.

   The mobile client database is extracted. This may take a few minutes.

Sample Directory Tree After Running Database Extract

Each registered mobile Web client requires a separate directory on the Siebel Remote server. The Database Extract program creates the appropriate directory and its subdirectories for each mobile Web client.

NOTE: The installation program also places a directory named txnproc in the docking subdirectory within the Siebel server root directory. Do *not* modify the contents of this directory under any circumstances.

The following example shows a portion of the server directory tree after you run Database Extract for mobile Web clients (and developers) named NADAMS and SSCOTT:

```
siebel
   docking
      nadams
         inbox
         outbox
      sscott
         inbox
         outbox
      txnproc
```

Running the Database Extract task creates a database snapshot for a given user, which consists of multiple files. These files contain all the data required to initialize the user's local client database, and are placed in the directory server*siebel_srvr_root*\docking\user\outbox. Each mobile user will download these files to create a local database and local file system (copies of literature files).

Initializing Each Developer's Local Database

After you extract each developer's local database, developers must now initialize their local databases. Initialization creates a local database file, called sse_data.dbf and stores it on the developer's local machine. Siebel Tools includes an initialization program that creates this file.

To initialize a developer's local database

1 Go to the machine on which Siebel Tools is installed.

   For the first developer, NREC used DEV_tools_A; for the second developer, NREC used DEV_tools_B. For information about the machines on which NREC installed different software, see Figure 4 on page 38.

2 Logon and connect to the Local database.

   The following message appears:

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

3 Click Yes.

   Siebel Tools connects to the Siebel Remote server and initializes the developer's local database.

   If initialization is successful, the sse_data.dbf file appears in the *tools_root*\local directory (where *tools_root* is the directory in which Siebel Tools has been installed, for example c:\siebdev).

Performing a Full "Get" Process

After initializing a local database, you must populate it with a read-only copy of all the projects and object definitions stored on the server database. This process is called a *full get*. A full get is equivalent to checking out all the projects from the 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.

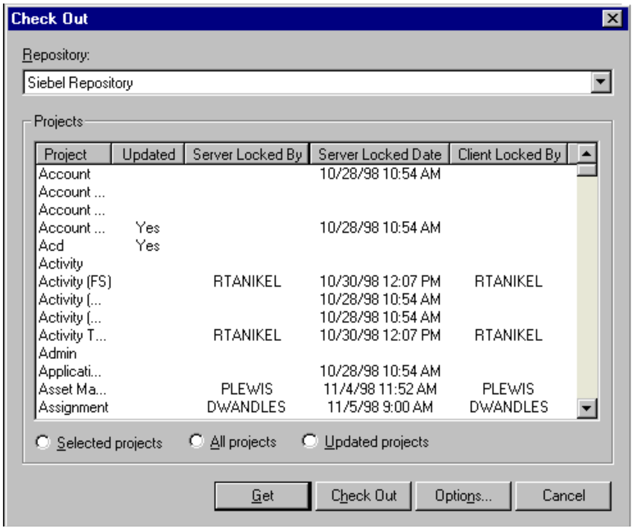
See "About the Local Development Environment" on page 73 for more information about the check in and check out process.

Having a read-only copy of the full repository on your local machine allows you to compile a Siebel repository file for your local database. This is required for testing changes locally.

To perform the initial full get of all projects from the repository

1. Start Siebel Tools, choosing Local in the Connect field.
2. Choose Tools > Check Out.

   The Check Out dialog box appears.



Repository: Siebel Repository

Projects

Project	Updated	Server Locked By	Server Locked Date	Client Locked By
Account			10/28/98 10:54 AM	
Account ...				
Account ...				
Account ...	Yes		10/28/98 10:54 AM	
Acd	Yes			
Activity				
Activity (FS)		RTANIKEL	10/30/98 12:07 PM	RTANIKEL
Activity (...			10/28/98 10:54 AM	
Activity (...			10/28/98 10:54 AM	
Activity T...		RTANIKEL	10/30/98 12:07 PM	RTANIKEL
Admin				
Applicati...			10/28/98 10:54 AM	
Asset Ma...		PLEWIS	11/4/98 11:52 AM	PLEWIS
Assignment		DWANDLES	11/5/98 9:00 AM	DWANDLES

○ Selected projects ○ All projects ○ Updated projects

3. Choose the name of your development repository from the Repository pick list.
4. Select the All Projects radio button.
5. Click the Options button.
6. In the Development Tools Options window, check to be sure that:

   - Your Server Data Source is pointing to your server development database.
   - Your Client Data Source is pointing to the local database you previously initialized and are currently running against.

   These data sources should match those shown in the Control Panel, under ODBC Data Sources.

7. Close the Development Tools Options window.
8. In the Check Out dialog box click Get.

   After the full get is complete, your currently open local repository has the same contents as the server repository from which you did the full get.

For more information about working with projects, see the projects chapter of the *Siebel Tools Reference*.

Getting Started at the User Interface Layer 5

This chapter gets you started defining and modifying object definitions using Siebel Tools. Following the NREC example, you will complete some simple configuration tasks focusing on the user interface level of the object hierarchy. Working through the tasks in this chapter will help you understand the hierarchy of the major user interface objects—applications, screens, views, and applets—and provide you with hands-on experience with basic configuration.

The NREC example helps place the tasks in context of making configuration changes to meet business needs. The chapter starts off by leading you through the process of narrowing down the out-of-the-box application to just those screens and views relevant to NREC. In the latter part of the chapter, you configure existing applets based on NREC requirements. You can follow these tasks like a tutorial, configuring against your sample database.

Creating an NREC Project

Before configuring object definitions, you should create a new project for NREC. You will use this project to group all new and modified object definitions. This will make it easier for you to find your work and will make it easy to export and import the object definitions.

To create a new project

1 In the Object Explorer, select the Project object type.

The Projects window opens in the Object List Editor.

2 In the Projects window, right-click and choose Add New Record.

3 Enter a new record with the following properties:

Property	Value
Name	NREC Configuration
Locked	TRUE

The project is now available for you to use. Moving all objects that you create or modify will help you keep track of your configuration changes.

Setting Your Target Browser

The Target Browser feature of Siebel Tools allows you to configure applications conditionally for different browsers, for example, Internet Explorer 5, Netscape 4.7, Netscape 6. Although you will not be configuring for different browsers in this guide, Siebel Tools requires that you have a target browser set. Otherwise, you will not be able to open the Web Layout Editor.

To set a target browser

1 Choose View > Toolbars > Configuration Context.

The Configuration Context toolbar appears.

2 Click the drop-down arrow in the Target Browser field, and then select Target Browser Config.

The Target Browser Configuration dialog box appears.

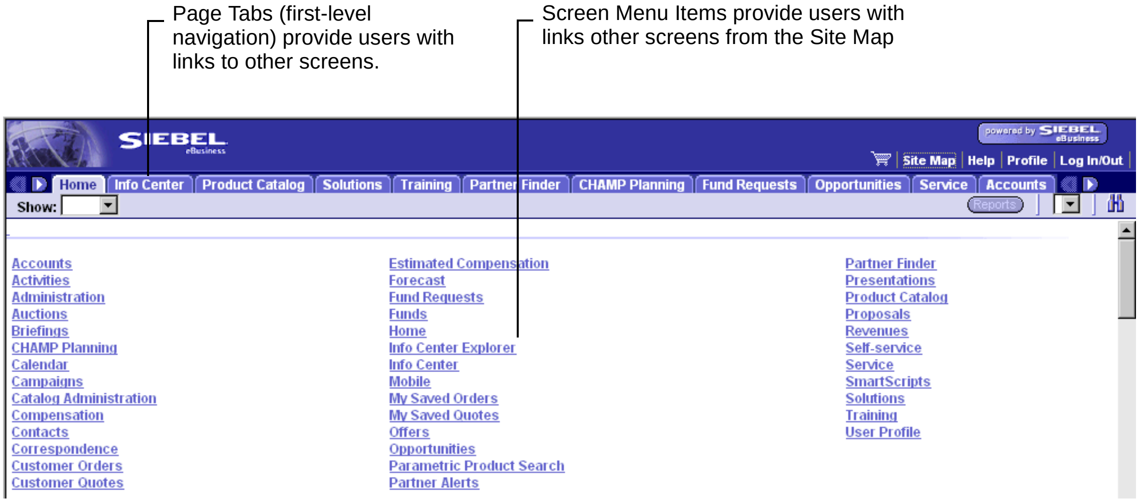
3 From the list of available browsers, select the ones you want as your target browsers and move them to Selected pane—for example, select IE 5.0.

Inactivating Screens

NREC’s implementation of the Siebel Partner Portal application does not include all the screens that come with the product, such as Service, Solutions, and Campaigns. These screens are not part of the system design and NREC does not plan to use them in the future. Therefore, NREC has decided to remove the screens from the user interface.

NREC has chosen to remove the screens by inactivating objects in the repository. Screens appear in the user interface as first-level navigation tabs and as links on the Site Map, as shown in Figure 12. These links are controlled by two child objects of the Application object, Page Tabs and Screen Menu Items. You can remove these links by making the object types inactive.

- Page Tabs control the first-level navigation tabs in the user interface.
- Screen Menu Items control the links that appear on the Site Map.



Page Tabs (first-level navigation) provide users with links to other screens.

Screen Menu Items provide users with links other screens from the Site Map

Figure 12. Page Tabs and Screen Menu Items

An alternative approach would be to create a responsibility that does not include any of the views within the screen and assign users to that responsibility. This would remove the screens from the user interface, but would leave the screen objects active in the repository. This approach allows you to add the screens back in the user interface without having to recompile the repository file. However, because these screens are not part of the design, assume NREC has chosen to inactivate the Screen Page Tab and Screen Menu Items in the repository.

To inactivate screen page tabs and screen menu items

1. In the Object Explorer, expand the Application object type.
2. In the Object List Editor, find and select the Siebel eChannel Application.

   NOTE: The application object named Siebel eChannel is the application called Partner Portal. The employee application is Siebel Partner Manager.

3. Lock the project locally by choosing Tools > Lock Project.

   The project to which the application belongs (Siebel eChannel) is locked and a Pencil Icon appears in the W column indicating that you can modify the record.

4. Change the value for the Project property from Siebel eChannel to NREC Configuration.

   This will move the application object to the NREC project making it easier for you to track changes.

5. In the Object Explorer, select the Page Tab object type (child of Application).

   The Page Tabs window appears in the Object List Editor.

6 Inactivate unnecessary screens listed in the Page Tabs window by selecting the check box in the Inactive field for each record.

Inactivate all Page Tab records *except* the following:

- Accounts Screen (SCW)
- Activities Screen (SCW)
- Contacts Screen (SCW)
- Opportunities Screen (SCW)

NOTE: These are the screens that should remain active in NREC's application. All others are not part of the system design.

The inactive Page Tabs turn red in the repository and will not appear as first-level navigation tabs user interface at run time.

7 In the Object Explorer, select the Screen Menu Item object type (child of Application).

The Screen Menu Items window appears.

8 Inactivate the unnecessary screens listed under Step 6 on page 94 in the Screen Menu Item window by selecting the Inactive field for each record.

Inactive Screen Menu Items do not appear on the Site Map at run time.

9 Compile and test your changes.

a Choose Tools > Compile Projects.

b In the Object Compiler dialog box, select the Selected Projects or Lock Projects radio button and then define the path to the Siebel repository file (.srf) to which you are compiling your changes.

Typically, this is the .srf file used by the Mobile Web client installed on your machine for testing.

For example, `D:\sea702\client\OBJECTS\ENU\siebel.srf`

NOTE: Be sure to make a back up of the siebel.srf file before you compile any changes.

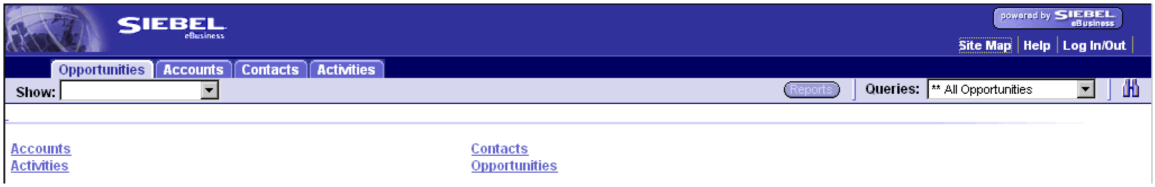
c Click Compile.

Siebel Tools compiles the object definitions for the locked project to the repository file.

See "Compiling Projects" on page 60 for more information about compiling.

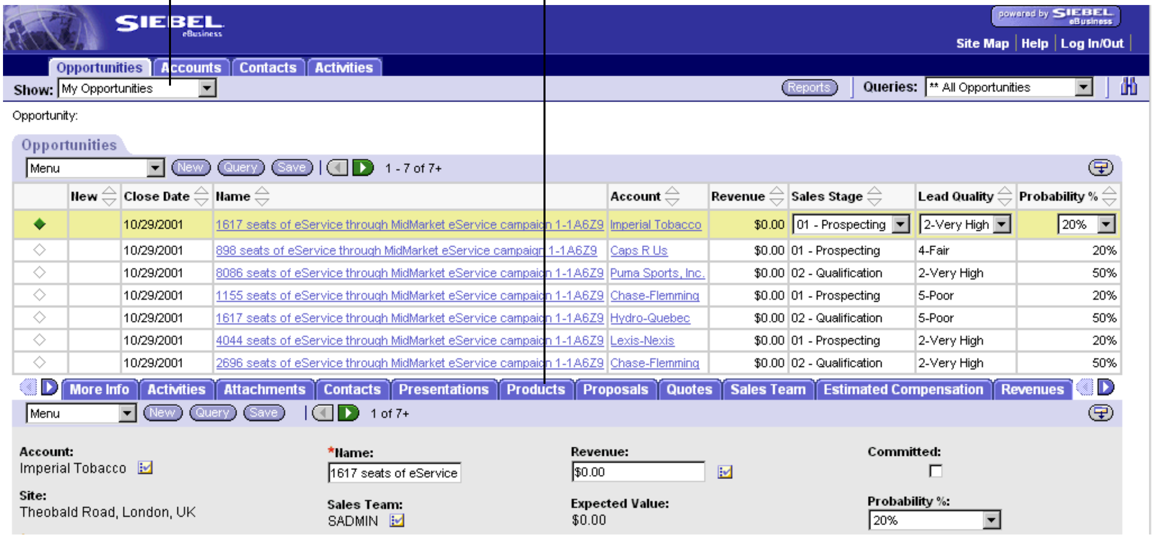
d Open the Partner Portal application.

The results of your changes appear when you start the Partner Portal application. Notice that there are now only five tabs and five links in the Site Map.



Inactivating Views

Each screen in the Siebel application is associated with a set of related views. Views are associated to a screen using the Screen View object type, which is a child object of the Screen object type. These views appear in the user interface in the Show drop-down list and in the second row of tabs as shown in Figure 13.



Opportunities Screen, set to show the My Opportunities View

Views associated with the Opportunities screen (Screen View object type)

Show: My Opportunities

Queries: ** All Opportunities

Opportunity:

Opportunities

1 - 7 of 7+

	New	Close Date	Name	Account	Revenue	Sales Stage	Lead Quality	Probability %
◆		10/29/2001	1617 seats of eService through MidMarket eService campaign 1-1A6Z9	Imperial Tobacco	$0.00	01 - Prospecting	2-Very High	20%
◇		10/29/2001	898 seats of eService through MidMarket eService campaign 1-1A6Z9	Caps R Us	$0.00	01 - Prospecting	4-Fair	20%
◇		10/29/2001	8086 seats of eService through MidMarket eService campaign 1-1A6Z9	Puma Sports, Inc.	$0.00	02 - Qualification	2-Very High	50%
◇		10/29/2001	1155 seats of eService through MidMarket eService campaign 1-1A6Z9	Chase-Flemming	$0.00	01 - Prospecting	5-Poor	20%
◇		10/29/2001	1617 seats of eService through MidMarket eService campaign 1-1A6Z9	Hydro-Quebec	$0.00	02 - Qualification	5-Poor	50%
◇		10/29/2001	4044 seats of eService through MidMarket eService campaign 1-1A6Z9	Lexis-Nexis	$0.00	01 - Prospecting	2-Very High	20%
◇		10/29/2001	2696 seats of eService through MidMarket eService campaign 1-1A6Z9	Chase-Flemming	$0.00	02 - Qualification	2-Very High	50%

1 of 7+

Account: Imperial Tobacco

Site: Theobald Road, London, UK

*Name: 1617 seats of eService

Sales Team: SADMIN

Revenue: $0.00

Expected Value: $0.00

Committed: □

Probability %: 20%

Figure 13. Default Opportunity View

NOTE: There are four levels of navigation in Siebel Applications. The first level is the row of page tabs across the top of the frame that allows users to navigate to other screens. The second level is the Show drop-down list that allows users to select different context views (for example, My Contacts, All Contacts, My Team's Contacts). The third level is the second row of tabs that allows users to navigate to other views. The fourth level (not shown in Figure 13) is a drop-down list that allows users to select different grandchild views. For more information about configuring navigation levels, see *Siebel Tools Reference*.

Many of the views associated with screens show data or contain functionality that is not part of NREC's system design. For example, many of the views associated with the Opportunities Screen, such as Attachments, Presentations, Proposals, Quotes, and Sales Teams, are not part of NREC's Partner Portal solution.

NREC has chosen to inactivate the unnecessary views associated with each of the remaining, active screens. The following table summarizes which views NREC has determined should be inactive.

Table 8. Active Screens and Views in NREC's eChannel Application

For these screens...	Inactivate these views
Accounts Screen (SCW)	Account Detail - Account Team View (SCW)
	Account Detail - Assets View (SCW)
	Account Detail - Orders View (SCW)
	Account Detail - Quotes View (SCW)
	Account Detail - Projects View
	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)
Activities Screen (SCW)	FS Activity Measurements View
	FS Activity Part Movements
	FS Activity Recommended Parts and Tools
	FS Activity Time Sheet View
	FS Activity View
	FS Activity Expense View
	FS Instructions
Contacts Screen (SCW)	Contact Detail - Service Request View (SCW)

Table 8. Active Screens and Views in NREC's eChannel Application

For these screens...	Inactivate these views
Opportunities Screen (SCW)	Incentive Compensation Opportunity Revenue Estimated Compensation List View 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
eChannel Home Screen	Not applicable.

There is more than one approach to controlling the views that appear in the user interface. One option would be to create a responsibility for NREC that includes only the relevant views for NREC's solution. Another option would be to use Siebel Personalization to show or hide views based on user profile attributes. However, because these views are not part of NREC's system design, NREC has chosen to inactivate unnecessary views in the repository.

See *Applications Administration Guide* for more information about responsibilities.

See *Personalization Administration Guide* for more information about controlling views with Personalization.

The following procedure uses the Accounts Screen (SCW) as an example. To complete the NREC example, you would need to inactivate the unnecessary views for the screens listed in Table 8 on page 98.

To inactivate Screen Views associated with a Screen

1. In the Object Explorer, expand the Screen object type.

   The Screens window appears in the Object List Editor.

2. In the Screens window, select the Accounts Screen (SCW).
3. Lock the project locally by choosing Tools > Lock Project.
4. Change the Project property from Opportunity (SCW) to NREC Configuration.
5. In the Object List Editor, select the Screen View object type (child of Screen).

   The Screen View window appears in the Object List Editor.

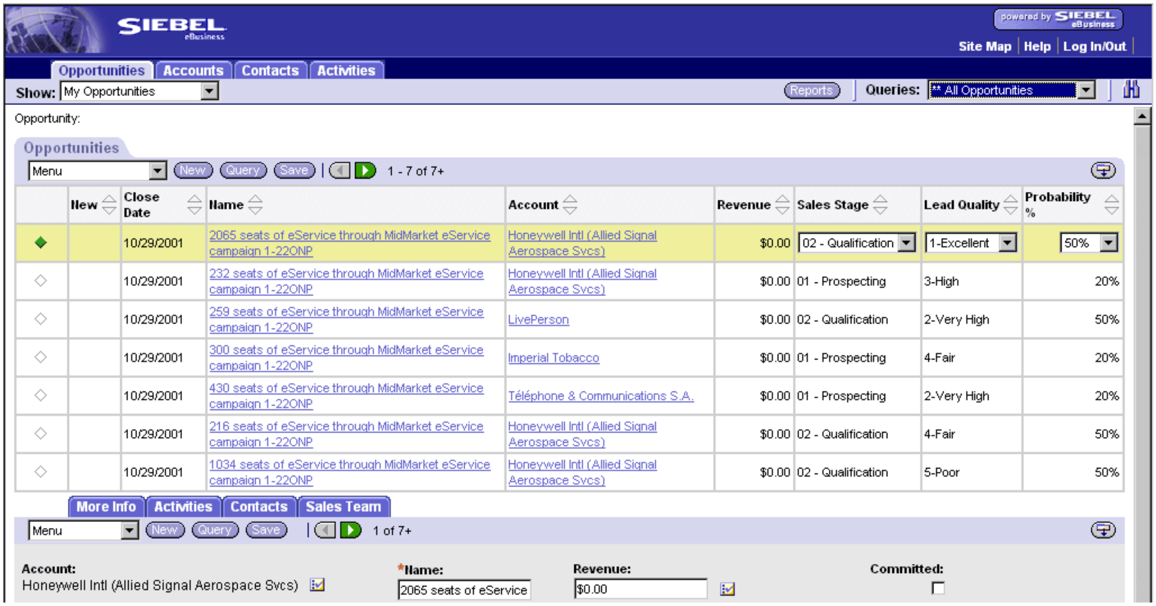
6. Select the Inactive field for the views that are not part of NREC's system design.

   See Table 8 on page 98 to determine which views to inactivate.

   NOTE: To move the Inactive field next to the Name field, right-click in Screen View window, and then choose Columns Displayed.

7. Repeat steps 2 through 7 for each active screen in the NREC example.
8. Compile and Test.

   See "Compiling Projects" on page 60, and "Testing Changes" on page 63.


SIEBEL eBusiness
powered by SIEBEL eBusiness
Site Map | Help | Log In/Out
Opportunities | Accounts | Contacts | Activities
Reports
Opportunities
Menu
New
Query
Save
More Info | Activities | Contacts | Sales Team
Menu
New
Query
Save

Show: My Opportunities

Queries: ** All Opportunities

Opportunity:

1 - 7 of 7+

	New	Close Date	Name	Account	Revenue	Sales Stage	Lead Quality	Probability %
◆		10/29/2001	2065 seats of eService through MidMarket eService campaign 1-22ONP	Honeywell Intl (Allied Signal Aerospace Svcs)	$0.00	02 - Qualification	1-Excellent	50%
◇		10/29/2001	232 seats of eService through MidMarket eService campaign 1-22ONP	Honeywell Intl (Allied Signal Aerospace Svcs)	$0.00	01 - Prospecting	3-High	20%
◇		10/29/2001	259 seats of eService through MidMarket eService campaign 1-22ONP	LivePerson	$0.00	02 - Qualification	2-Very High	50%
◇		10/29/2001	300 seats of eService through MidMarket eService campaign 1-22ONP	Imperial Tobacco	$0.00	01 - Prospecting	4-Fair	20%
◇		10/29/2001	430 seats of eService through MidMarket eService campaign 1-22ONP	Téléphone & Communications S.A.	$0.00	01 - Prospecting	2-Very High	20%
◇		10/29/2001	216 seats of eService through MidMarket eService campaign 1-22ONP	Honeywell Intl (Allied Signal Aerospace Svcs)	$0.00	02 - Qualification	4-Fair	50%
◇		10/29/2001	1034 seats of eService through MidMarket eService campaign 1-22ONP	Honeywell Intl (Allied Signal Aerospace Svcs)	$0.00	02 - Qualification	5-Poor	50%

1 of 7+

Account:	*Name:	Revenue:	Committed:
Honeywell Intl (Allied Signal Aerospace Svcs)	2065 seats of eService	$0.00	□

Configuring NREC’s Activity Applets

This section gets you started working with applets. You will add and remove fields from the user interface, reposition controls, and modify text labels. After completing all the procedures, you will compile and test your changes.

About Applets and Applet Web Templates

Applets are rendered in the user interface by combining objects definitions stored in the Siebel repository with layout and formatting information contained in applet Web templates. Objects such as controls and list columns are mapped to place holders in Web templates. At runtime, the Siebel Web Engine creates these objects, places them in the appropriate spot in the Web template, retrieves the relevant data from the Siebel database, and uses the HTML contained in the Web template to display the applet in the user interface.

Modifying or creating applets is primarily done in Siebel Tools. Tasks include defining object properties in the Object List Editor and mapping controls to place holders in Web templates using the Web Layout editor. However, you may also want to change style aspects of an applet, such as the color scheme, in which case you would work with an external cascading style sheet (see Chapter 5, “Getting Started at the User Interface Layer”). Most of the time, you do not need to modify Web templates themselves.

For more information about Web templates, see *Siebel Tools Reference*.

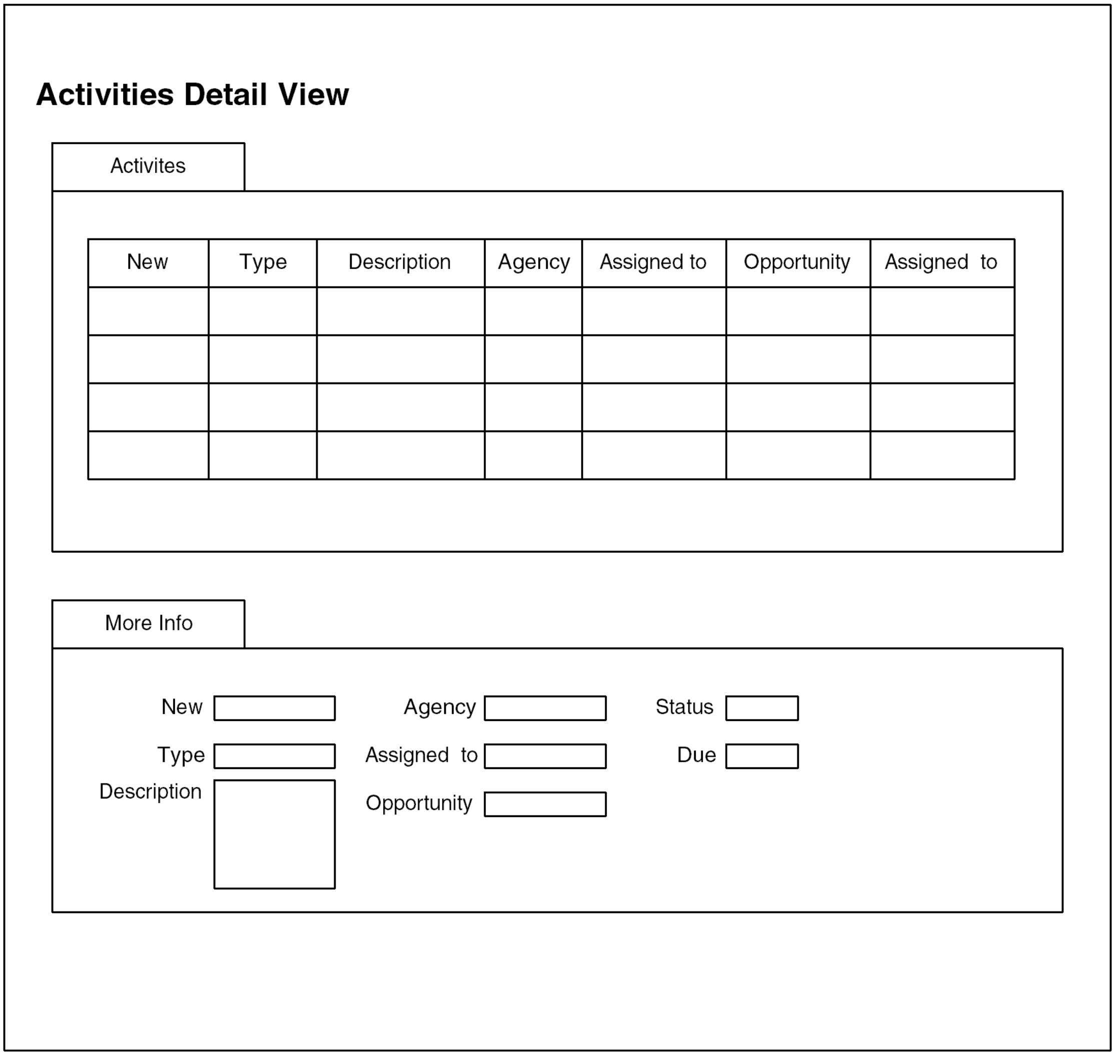
NREC’s Business Requirements

Two applets appear on NREC’s Activities view: Activity List Applet and the Activity Detail Applet. These applets need to be modified to meet NREC’s business requirements. The business requirements and related tasks are listed below.

- To enhance usability NREC requires that all unnecessary fields be removed from the user interface. Additionally they require that the fields appear in a particular order. You will remove fields such as Campaign, Fund Request, and Associated Cost, and reorder the remaining fields based on NREC’s design. See “Removing Fields (List Columns and Controls) from the User Interface” on page 105.
- Partner agents need to have the ability to track which opportunity is associated with each activity. This functionality is not available out of the box. Therefore, you will add the opportunity field to both applets. See “Exposing Fields in the User Interface” on page 107.

These tasks are all in the user interface layer of the Siebel Object Model. No work is required in the underlying business object layer or data object layer. The tasks in this section give you some hands-on practice configuring applets.

The design team prepared a view mock-up that shows what the view and applets should look like after they are complete.



Activities Detail View

Activites

New	Type	Description	Agency	Assigned to	Opportunity	Assigned to

More Info

New

Type

Description

Agency

Assigned to

Opportunity

Status

Due

Figure 14. Activity Detail View Mock-up

Removing Fields (List Columns and Controls) from the User Interface

You can remove fields from the user interface using the Web Layout Editor. For each field (list columns or controls) that you do not want to be displayed, you simply delete it from the Web template. This unbinds the list column or control from the Web template. The objects remain intact as child objects of the applet, but they are not rendered in the user interface at runtime.

Applets generally appear in several modes. Each mode has a different Web template associated with it. The default mode is controlled by the view in which the applet appears. The three most common modes are:

- **Base.** Use this mode to display the applet as read-only.
- **Edit.** Use this mode to display the applet as a form and allow users to modify values. This is useful for adding new records, editing records, and queries.
- **Edit List.** Use this mode to display a list applet in an editable mode, allowing users to edit directly in the list. Not applicable for form applets.

When modifying applets you need to modify each mode. The following procedure uses the Base mode of the Activity Detail Applet (SCW) as an example. The NREC design requires fields to be removed from all modes of the Activity List Applet (SCW) and the Activity Detail Applet (SCW).

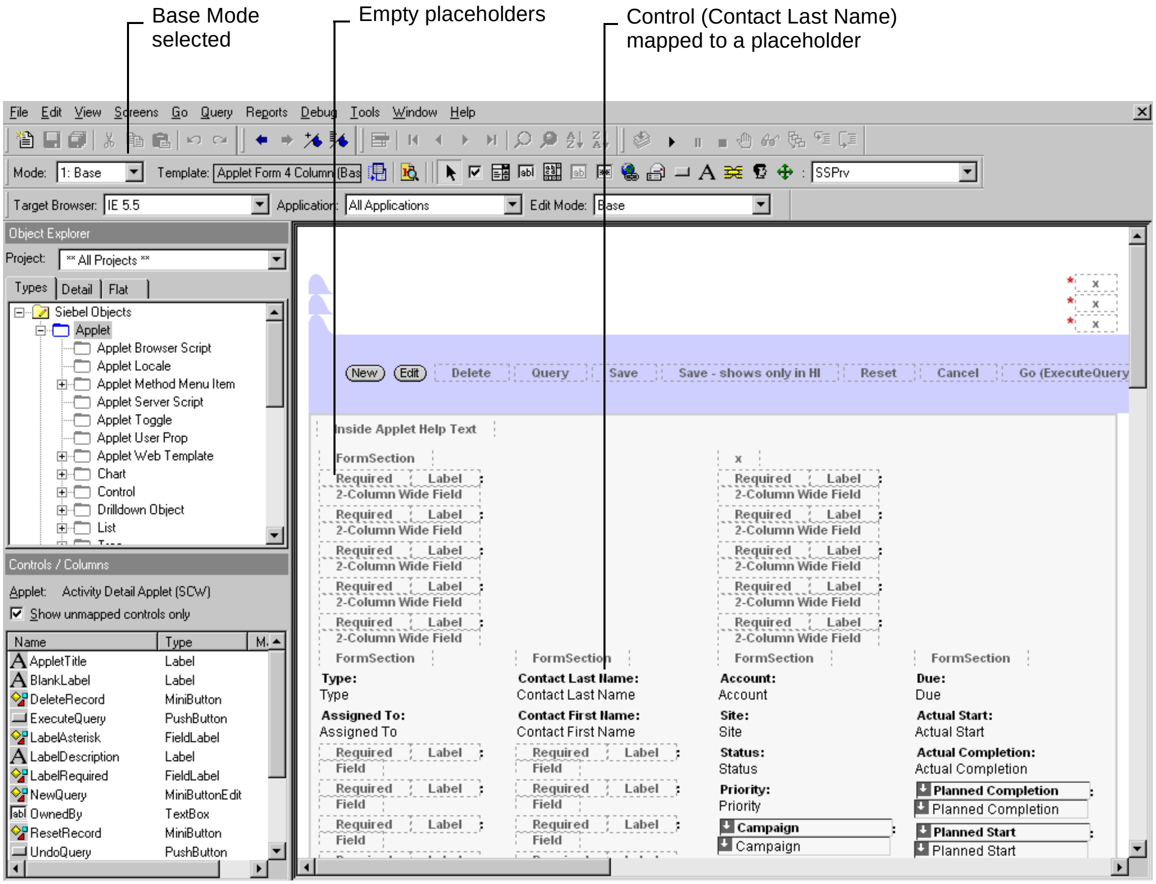
To remove fields from the user interface

1. In the Object Explorer, select the Applet object type.
2. In the Object List Editor, select Activity Detail Applet (SCW).
3. Lock the project locally by choosing Tools > Lock Project.
4. Change the Project property from Activity (SCW) to NREC Configuration.

5 Right-click and then choose Edit Web Layout.

The Web Layout Editor appears with the Base mode of the applet displayed.

NOTE: You need to have your target browser set before you can open the Web Layout Editor. See “Setting Your Target Browser” on page 91.



Base Mode selected

Empty placeholders

Control (Contact Last Name) mapped to a placeholder

Mode: 1: Base
Template: Applet Form 4 Column (Bas
Target Browser: IE 5.5
Application: All Applications
Edit Mode: Base
Project: ** All Projects **
Applet: Activity Detail Applet (SCW)
☑ Show unmapped controls only

Name	Type	M.
AppletTitle	Label	
BlankLabel	Label	
DeleteRecord	MiniButton	
ExecuteQuery	PushButton	
LabelAsterisk	FieldLabel	
LabelDescription	Label	
LabelRequired	FieldLabel	
NewQuery	MiniButtonEdit	
OwnedBy	TextBox	
ResetRecord	MiniButton	
UndoQuery	PushButton	

6. In the Web Layout Editor, delete the controls Programs/Other Comments, Campaign, Fund Request, and Associated Cost.

   a. Point to a control with the mouse.

   b. Click the mouse to select it.

      You can select multiple controls by holding down the Ctrl key while selecting with the mouse.

   c. Right-click and then choose Delete.

      Use the mock-up shown in Figure 14 on page 104 to determine where to place fields.

7. Choose File > Save.

8. In the Web Layout Editor, right-click and then choose Preview to see how the applet will look when it is rendered in the user interface.

9. Select Edit from Mode drop-down list on the Configuration Context toolbar and repeat Step 6 through Step 8.

10. Choose File > Close.

Exposing Fields in the User Interface

NREC's partner agents must be able to associate an Opportunity record with an Activity record. However, the Activity List Applet and the Activity Detail Applet do not include the Opportunity field by default. Although the field doesn't appear by default, you can reconfigure the applets to display it. The Opportunity field is already defined on the underlying business component—the Action business component—so all that is required is to expose the field in the user interface.

The tasks for exposing the opportunity field in the user interface differ slightly for a list applet and form applet.

- List applets require that the field be defined as a list column object.
- Form applets require that the field be defined as a control object.

NOTE: The Activity Detail Applet (SCW) is a form applet.

After list column or control objects are defined, the objects are mapped to place holders in Web templates.

See *Siebel Tools Reference* for detailed information about configuring applets.

Adding the Opportunity Field to the Activity List Applet (SCW)

On list applets, fields are defined as List Column object types. For the Activity List Applet (SCW) you need to define the Opportunity field as a List Column before mapping it to a Web template.

To define a list column object for a list applet

1. In the Object Explorer, expand the Applet object type.
2. In the Object List Editor, find and select the Activity List Applet (SCW).
3. If the project is not locked, do the following:
   - **a** Choose Tools > Lock Project.
   - **b** Change the Project property from Activity (SCW) to NREC Configuration.
4. In the Object Explorer, expand the List object type (a child object of Applet).
5. Expand the List object type, and then select the List Column object type (a child object of List).

   The List Columns window appears in the Object List Editor.

6 In the List Columns window, add a new record using the following properties:

Property	Value	Description
Field	Opportunity	The values available from the drop-down list are the fields defined on the parent business component of the Applet (in this case, the Action business component).
Display Name	Opportunity	Text that appears in the user interface.
Runtime	TRUE	When the property is set to TRUE, the system makes a run-time check to see if a picklist, calculator, calendar, or MVG pop-up button will be provided.

Adding the Opportunity Field to a Activity Detail Applet (SCW)

For form applets, fields are defined as Control object types. For the Activity Detail Applet (SCW) you will define the Opportunity field as a control before mapping it to a Web template.

To add a control object to a form applet

1 In the Object Explorer, expand the Applet object type.

2 In the Object List Editor, find and select the Activity Detail Applet (SCW).

3 In the Object Explorer, select the Control object type (child of Applet).

4 In the Controls window of the Object List Editor, add a new record with the following properties:

Property	Value	Description
Name	Opportunity	Unique name of the control object.
Caption	Opportunity	Text that appears in the user interface.
Field	Opportunity	Field defined on the parent business component. In this case, the parent business component is Action.
HTML Type	Field	HTML Control Type.

Mapping List Columns or Controls to Web Templates

After you have defined the list column and control object types for the applets, you are ready to map them to place holders in a Web template. You do this using the Web Layout Editor. The process for mapping list columns or controls to Web templates is the same for both list applets and form applets.

NOTE: It is also possible to create controls while working in the Web Layout Editor. To do this you drag a control type from the toolbar, drop it onto the template, and then define the necessary properties. This method is an alternative to defining the controls in the Object List Editor as described in the previous section.

The following procedure uses the Opportunities list column on Activity List Applet (SCW) as an example. Note that NREC's design would require that you follow the same procedure for mapping the Opportunity control to the Web templates for the Activity Detail Applet (SCW) as well. For both applets, you must map the control to the Web templates for each applet mode.

To map fields (controls and list columns) from an applet to a Web template

1. In the Object Explorer, select the Applet object type.
2. In the Object List Editor, select the Activity List Applet (SCW).
3. Right-click and choose Edit Web Layout.

   The Web Layout Editor appears with the applet displayed in base mode.

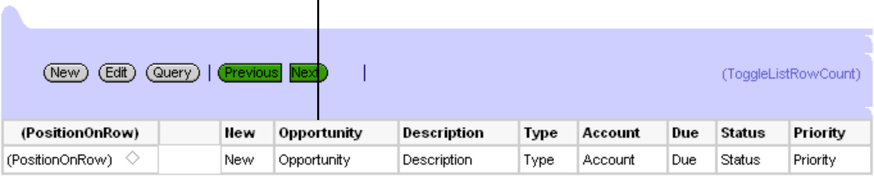
4. Make room for the Opportunity field by rearranging the controls that are already mapped to the Web template.

   You can move controls by dragging and dropping them or by using the Shift to Previous Placeholder and Shift to Next Placeholder buttons on the toolbar.

5. From the Controls/Columns window, drag the Opportunity list column and drop it into an empty place holder in the web template.

   NOTE: If the Controls/Columns window is not open, choose View > Windows > Controls.

6 Right-click and then choose Preview to see a how the applet will look rendered in the user interface.



(PositionOnRow)		New	Opportunity	Description	Type	Account	Due	Status	Priority
(PositionOnRow) ◇		New	Opportunity	Description	Type	Account	Due	Status	Priority

Opportunity List Column added.

NOTE: You can also export the preview to an HTML file by choosing File > Export, and then choosing a file name and location in the Save As dialog.

7 Make adjustments as necessary.

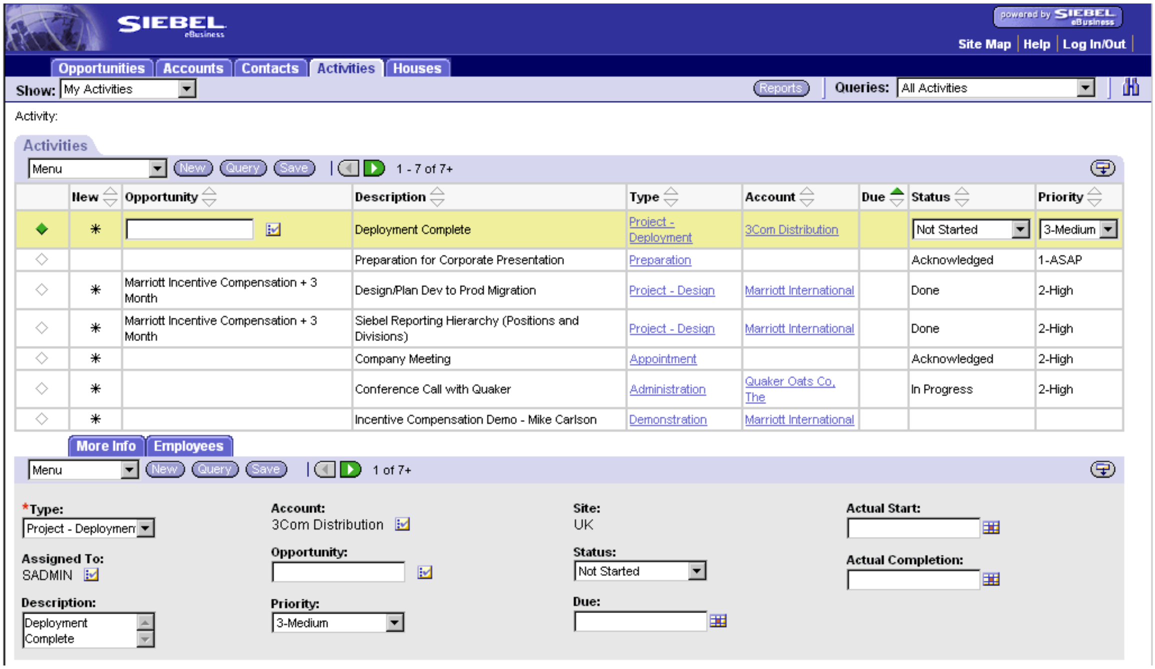
8 Choose File > Save.

9 Repeat the procedure for each of the mode of the Activity List Applet (SCW) and the Activity Form Applet (SCW).

You can switch applet modes using the Mode drop-down list in the Configuration Context Toolbar.

Reviewing the Results

In the previous sections you removed the unnecessary fields from applets, added a new field, and changed the display name and caption of a field. After you compile your changes, the applets should appear as shown in Figure 15.



Show: My Activities

Queries: All Activities

Activity:

1 - 7 of 7+

	New	Opportunity	Description	Type	Account	Due	Status	Priority
◆	✳		Deployment Complete	Project - Deployment	3Com Distribution		Not Started	3-Medium
◇			Preparation for Corporate Presentation	Preparation			Acknowledged	1-ASAP
◇	✳	Marriott Incentive Compensation + 3 Month	Design/Plan Dev to Prod Migration	Project - Design	Marriott International		Done	2-High
◇	✳	Marriott Incentive Compensation + 3 Month	Siebel Reporting Hierarchy (Positions and Divisions)	Project - Design	Marriott International		Done	2-High
◇	✳		Company Meeting	Appointment			Acknowledged	2-High
◇	✳		Conference Call with Quaker	Administration	Quaker Oats Co, The		In Progress	2-High
◇	✳		Incentive Compensation Demo - Mike Carlson	Demonstration	Marriott International			

1 of 7+

*Type: Project - Deploymen

Assigned To: SADMIN

Description: Deployment Complete

Account: 3Com Distribution

Opportunity:

Priority: 3-Medium

Site: UK

Status: Not Started

Due:

Actual Start:

Actual Completion:

Figure 15. Activity Applets

Configuring the House and Opportunity Entities 6

In this chapter you will work in three layers of the object model—data objects layer, business objects, and user interface object—with the goal of constructing four views based on NREC's design specifications. This will involve both modifying existing object definitions and creating new ones, and includes the following tasks:

- Creating and modifying views
- Creating and modifying applets
- Creating and modifying business components
- Modifying business objects
- Adding columns to an existing base table
- Using 1:M extension tables

The tasks in each section start at the database layer then progress up to the user interface layer. Working through the material this way will give you a good understanding of the object definition sequence and give you plenty of practice with many of the common configuration tasks.

Tasks are covered in detail where they first appear. If a task also occurs later in the chapter, it is presented at a higher level with a cross reference to the detailed procedure.

You can follow the tasks in this chapter like a tutorial, configuring against the sample database.

Configuring the House Detail View

NREC's House Detail view needs to be able to 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 NREC's needs reasonably well. However, attributes such as square feet and number of bedrooms are not part of the Siebel data model.

You can extend the standard data model to meet your unique business requirements. For example, assume NREC's design team has decided to extend the internal products base table by adding additional columns. These columns will be used to store the following additional attributes for each house.

- Address
- City
- State
- ZIP Code
- Square Feet
- Price
- Number of Bedrooms
- Number of Bathrooms

NOTE: Although List Price exists as standard field on the Internal Product business component, assume that NREC's design team has decided to add a column to S_PROD_INT to keep all the house attributes stored in one table.

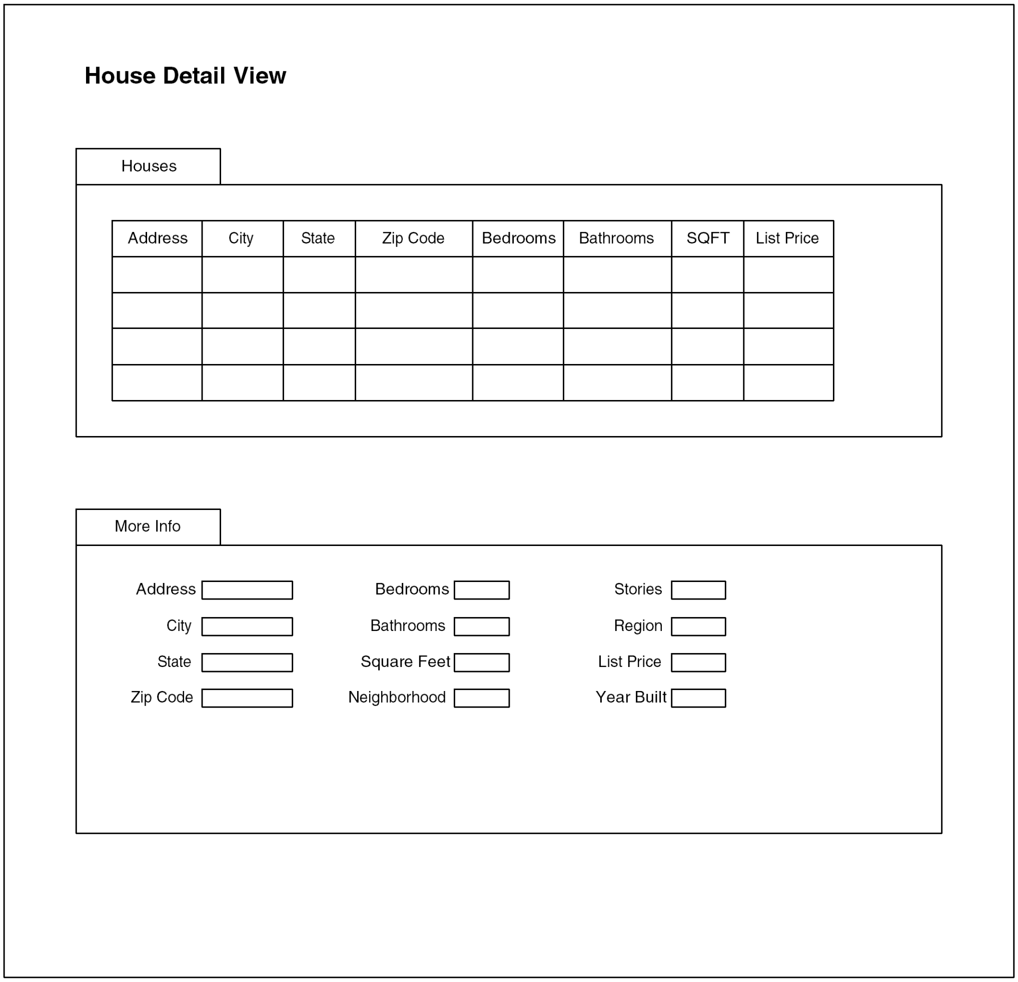
For detailed information about extending the data model, see *Siebel Tools Reference*.

In this section you will add columns to the internal products table, and then modify objects in the business object layer and the user interface layer to accommodate the columns changes. You will modify existing objects as well as create new objects where appropriate.

The tasks for accomplishing this are:

- "Extending the Database by Adding New Columns to the Base Table" on page 117
- "Configuring the Internal Product Business Component" on page 121
- "Modify Existing Product Applets to Display NREC Attributes" on page 123
- "Creating the House Detail View" on page 127
- "Creating the Houses Screen" on page 128
- "Compiling and Testing" on page 130

Figure 16 shows a mock-up of NREC's House Detail View. This is a new list-form view using existing applets.



House Detail View

Houses

Address	City	State	Zip Code	Bedrooms	Bathrooms	SQFT	List Price

More Info

Address		Bedrooms		Stories	
City		Bathrooms		Region	
State		Square Feet		List Price	
Zip Code		Neighborhood		Year Built	

Figure 16. House Detail View Mock-Up

Extending the Database by Adding New Columns to the Base Table

You add columns to a base table using the Database Extension Designer. This involves adding columns to the base table using the Object List Editor in Siebel Tools and then applying the changes your local database schema.

Adding New Columns to the Products Table

First you must add the columns to the base table. You do this by creating additional Column object definitions for the base table object. For example, you are adding columns to the products table, S_PROD_INT.

To add a new column to a base table

1. In the Object Explorer, expand the Table object type.
2. In the Tables window, Find and select S_PROD_INT.
3. Lock the project locally by choosing Tools > Lock Project.

   NOTE: The project field is read-only for tables. You cannot change the project to from Newtable to NREC Configuration.

   Notice that the table is of type Data (Public). Only public tables can be extended.

4. In the Object Explorer, select Column object type (child of table).

   The names of the columns for the S_PROD_INT table appear in the Object List Editor.

5 In the Columns window, add the following records.

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

NOTE: Names of extension columns begin with X_ (for example, X_ADDRESS). The User Names of extension columns end with Ext (for example, X_ADDRESS Ext). Siebel Tools automatically enforces these conventions.

The logical database schema is changed based on the information you entered, but you still need to physically apply the changes to your local database.

Applying Schema Changes to Your Local Database

After you have added columns to the base table as described in the previous section, you have to physically apply the schema changes to your local database.

To apply schema changes to the local database

1 In the Object Explorer, select the Table object type.

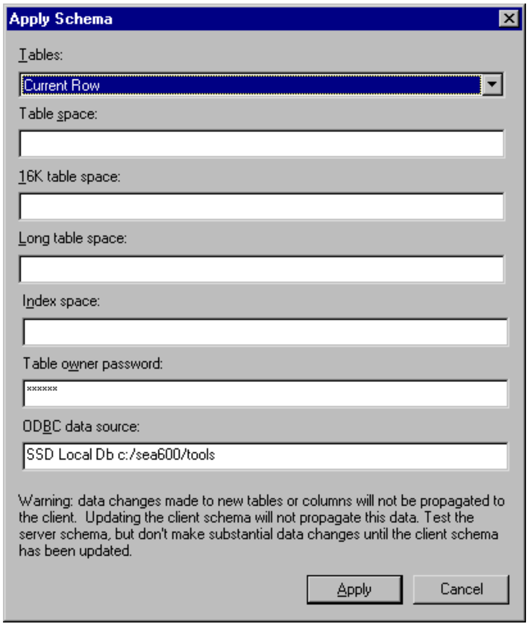
2 In the Tables window, select S_PROD_INT.

3 Click the Apply button.

A warning appears saying that you are connected to a local database and asking if you want to continue.

4 Click OK.

The Apply Schema dialog box appears.



Tables: Current Row

Table space:

16K table space:

Long table space:

Index space:

Table owner password: ******

ODBC data source: SSD Local Db c:/sea600/tools

Warning: data changes made to new tables or columns will not be propagated to the client. Updating the client schema will not propagate this data. Test the server schema, but don't make substantial data changes until the client schema has been updated.

5 In the Tables pick list, select Current Row.

This will update the database to reflect the schema change to the current selected row only (S_PROD_INT).

Other options are:

- **All.** Update the database to reflect all changes made to the dictionary. This option forces each database object to be compared with the data dictionary, and updated if required.
- **Current Query option.** Update the database to reflect modifications made to the tables in the current query only.

6 Enter the password for the Siebel database owner in the Password text box.

NOTE: The default password is SIEBEL.

Do not specify a table space or index space.

7 Verify that the ODBC (Open DataBase Connectivity) connection specified in the ODBC Data Source text box is correct for your environment.

NOTE: You cannot apply schema changes to any database other than the one to which you are currently connected. If you are connected to the Sample database, be sure that the OBDC Data Source points to the correct driver. For example, if your Sample database is installed in your client directory on the D drive, the connect string is `SEAW Samp Db D:/sea702/client.`

8 Click Apply to apply the new columns to the physical schema on your local database.

After this process has been completed, the columns you added to the logical schema as described in "Extending the Database by Adding New Columns to the Base Table" on page 117 now physically exist on your local database and are available to use in your configuration.

NOTE: Typically, after you have tested changes in the local database environment, you would need to apply these changes to the server database. Checking in a project copies configuration changes to the server, but this process does not apply physical database extension changes. Therefore, you would need to go through a separate process to apply database extension changes to the server database. For information on this process, see *Siebel Tools Reference*.

Configuring the Internal Product Business Component

After you have applied the extension columns to your local database schema, you can add the fields to the business component that will use the columns to store data. In the NREC example, you will be adding fields to the Internal Product business component. Additionally, you need to modify certain properties of the business component to allow users to update records.

The following procedure uses the square feet attribute as an example. To complete the NREC example, you need to add fields for all the attributes listed at the beginning of this section as described in "Configuring the House Detail View" on page 114.

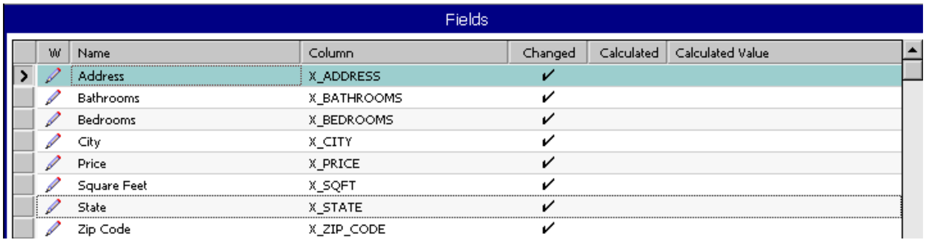
To add fields to the Internal Products business component

1. In the Object Explorer, expand the Business Component object type.
2. In the Business Components window, select the Internal Product business component.
3. Lock the project locally by choosing Tools > Lock Project.
4. Change the value of the Project property from Product to NREC Configuration.
5. In the Object Explorer, select the Field object type (child of Business Component).
6. In the Fields window, right click and enter a record with the following properties:

Field	Value
Name	Square Feet
Column	X_SQFT

7. Repeat Step 6 on page 121 for each field you want to add.

   For example, NREC entered fields shown below.



Fields

W	Name	Column	Changed	Calculated	Calculated Value
✎	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	✓		

To change the default properties of the Internal Products business component

1. Select the Internal Product business component in the business component window.

2. Choose View > Windows > Properties Window.

3. Change the values of the following properties from the default value of TRUE to FALSE.

Field	Value
No Delete	FALSE
No Insert	FALSE
No Merge	FALSE
No Update	FALSE

   Changing the values of these properties will allow users to enter records using applets based on this business component.

Modify Existing Product Applets to Display NREC Attributes

After the fields are added to the business component, you can display them in the user interface. Assume that NREC's design team has determined that configuring the following two applets would be easier than creating new ones.

- Product List Applet
- Product Form Applet

These applets are already based on the Internal Product business component and meet NREC's needs reasonably well.

The following procedures cover exposing fields in the user interface at a more abstract level than was covered in the previous chapter. For detailed procedures, see "Exposing Fields in the User Interface" on page 107.

Configuring the Product List Applet

The following procedure covers configuring the Product List Applet.

To expose fields in the Product List Applet

1. Navigate to the Product List Applet.
2. Lock the project locally by choosing Tools > Lock Project, and then change the value of the Project property from Product (SSE) to NREC Configuration.

3 Add List Column objects for each of the new fields.

Note that the object hierarchy is Applet > List > List Column.

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

NOTE: The Show In List property determines whether the field will be displayed by default in the list applet. Fields with a Show In List property set to FALSE will appear gray in the Web Template layout editor. These fields will be hidden by default in the user interface. However, users can add them to the display by choosing Columns Displayed from the applet-level menu.

4 Change the display name of existing list columns:

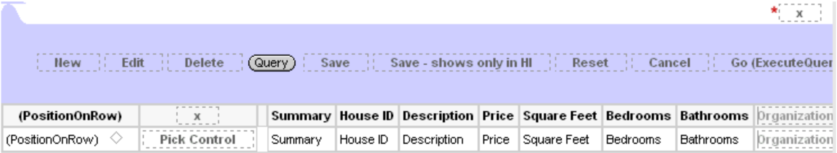
- Change the display name of the Part # list column to House ID.
- Change the display name of the Name list column to Summary.

5 Select the Product List Applet record and right-click to open the Web Layout Editor.

6 In the Web Layout Editor, delete the unnecessary columns and add the new columns to the Web template for the Base mode of the applet.

- For example, delete the following: Orderable, Service Product, Lead Time, Compensable, Unit of Measure, Product Line, Serialized, Class, Equivalent Product, Product Line, Serialized, and Revision.

- Add the fields that you defined in Step 3 on page 124, including the fields that will be hidden by default.



(PositionOnRow)	x		Summary	House ID	Description	Price	Square Feet	Bedrooms	Bathrooms	Organization
(PositionOnRow) ◇	Pick Control		Summary	House ID	Description	Price	Square Feet	Bedrooms	Bathrooms	Organization

7 In the Web Layout Editor, delete the unnecessary columns and add the new columns to the Web Template for the Query and Edit mode of the applet.

- Delete Vendor and Class.

- Add the fields that you defined in Step 3 on page 124, including the fields that will be hidden by default.

NOTE: You can switch applet modes using the Mode drop-down list on the Configuration Context Toolbar.

Configuring the Product Form Applet

The following procedure covers configuring the Product Form Applet.

To configure the Product Form Applet

1 Navigate to the Product Form Applet.

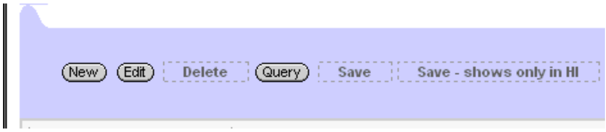
2 Lock the project locally by choosing Tools > Lock Project, and then change the value of the Project property from Product (SSE) to NREC Configuration.

3. Add Control objects (child of Applet) for each of the new fields.

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

4. Modify the caption property of the existing controls:
   - Change the caption of the Part # control to House ID.
   - Change the caption of the Name control to Summary.
5. Select the Product Form Applet record, and then right-click and choose Edit Web Layout.
6. In the Web Layout Editor delete the unnecessary controls and add map the new controls for each applet mode.

   Use Figure 16 on page 116 to determine what controls to add and delete.
7. In the Web Layout Editor, map the NewRecord and EditRecord controls to the appropriate place holders in the applet banner for each applet mode.



The NewRecord and EditRecord controls are not mapped to the web templates for the standard Product Form Applet. Mapping them to the applet will allow users to create and edit records.

Creating the House Detail View

NREC wants to create a new view to display the Product List and Product Form Applets. They want to have the Products List Applet on top and the Products Form Applet below. This type of view is the standard list-form view. You will use the View Wizard to create the view using the applets modified in the previous sections.

To create a new view

1. Choose File > New Object.
2. The New Object dialog box appears.
3. Under the General tab, select View and then click OK.

   The New View Wizard opens.

4. Complete the wizard providing the following information as you go.

Field	Value
Project	NREC Configuration
Name	NREC House Detail View
Business Object	Internal Product
Title	House Detail View
Web Template	View Detail (Parent with Pointer)
Selected Applets	Product List Applet Product Form Applet

The wizard creates the view and related objects and then opens the view in the Web Layout Editor for you to review.

NOTE: When creating views with special visibility properties, such as Organization, Manager, or Sales Rep, set the Visibility Applet Type property of the View object not the View Web Template Item object. The only exception to this rule is the Home Page. Set the Home Page visibility using the View Web Template Item object.

Creating the Houses Screen

After creating the view, you must add it to a screen so that it will appear in the user interface. However, in this case, NREC has chosen to create a new screen for Houses. This screen will be used to group all the views related to the Houses entity. After creating the screen, you will need to define the a Screen View child object to associate the screen with the House Detail View.

To create the Houses Screen

1 In the Object Explorer, expand the Screen Object type.

The Screens window opens in the Object List Editor.

2 In the Screen window, right-click and select New Record.

3 Enter a new record with the following properties:

Field	Value
Name	NREC House Screen
Project	NREC Configuration
Viewbar text	Houses

4 In the Object Explorer, select the Screen View object type (child of Screen).

The Screen View window opens in the Object List Editor. Screen Views associate views to a screen.

5 In the Screen View window, add a new record using the following values.

Field	Value	Description
View	NREC House Detail View	Name of associated view.
Viewbar Text	More Info	Text that appears in the second row of tabs.
Sequence	20	Determines the sequence in which view tabs are displayed.

6 In the Object List Editor, select the Screen object type again.

7 In the Screen window, set the Default View property to House Detail View.

Defining Page Tab and Screen Menu Items

For screens to appear in the user interface, you must define Page Tab and Screen Menu Items. These objects are child objects of the Application object type. Page tabs appear as the tabs across the top of the application and Screen Menu Items appear in the site map.

To define Page Tabs and Screen Menu Items

1 In the Object Explorer, expand the Application object type.

2 In the Application window, select the Siebel eChannel application.

3 In the Object Explorer, select the Page Tab (child of Application) object type.

4 In the Page Tab window, right-click and enter a record with the following properties:

Field	Value	Description
Name	NREC House Screen	Unique name for the Page Tab.
Screen	NREC House Screen	Screen object associated with the page tab.
Sequence	125	Determines the sequence in which the page tabs appear.
Text	Houses	Text that appears in the tab.

5 In the Object Explorer, select the Screen Menu Item object type (child of Application).

6 In the Screen Menu Item window, add a new record with the following properties:

Field	Value	Description
Screen	NREC House Screen	Screen associate with the Screen Menu Item.
Sequence	15	Determines the sequence that the item appears.
Text	Houses	Text that appears as the link on the Site Map.

Compiling and Testing

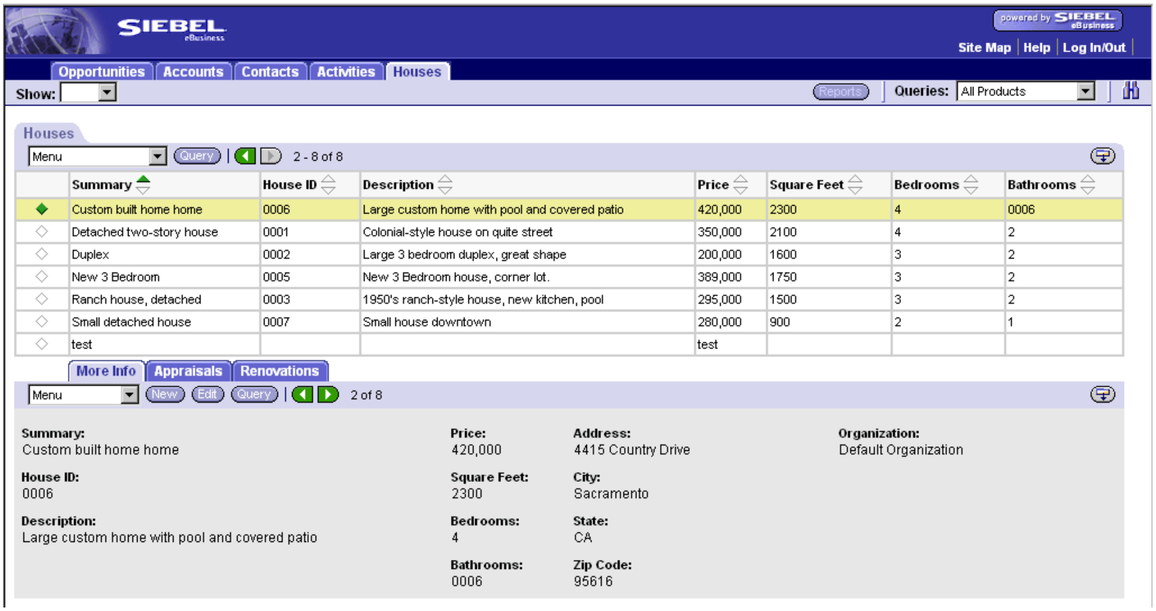
Now you are ready to compile and test your changes. After compiling the changes to a repository file, but before testing the changes, you must perform the following tasks using a Siebel employee application, such as Siebel Call Center or Siebel Sales:

- Register the House Detail View in the application.
- Associate the House Detail View to a responsibility.

NOTE: You cannot edit the responsibilities that are part of the seed data that ships with the product. You must create a new responsibility to be able to associate new views to it. You can copy one of the sample responsibilities, such as Siebel System Administrator, and then customize it for your purposes. You can create a new responsibility for testing, call it NREC Test, and then associate the SADMIN user to the responsibility. When you log on as SADMIN, you will see the new view.

You must complete these steps any time you create a new view. See "Testing Changes" on page 63 for detailed procedures.

Figure 17 shows what the applets and view look like after configuring them.



	Summary	House ID	Description	Price	Square Feet	Bedrooms	Bathrooms
◆	Custom built home home	0006	Large custom home with pool and covered patio	420,000	2300	4	0006
◇	Detached two-story house	0001	Colonial-style house on quite street	350,000	2100	4	2
◇	Duplex	0002	Large 3 bedroom duplex, great shape	200,000	1600	3	2
◇	New 3 Bedroom	0005	New 3 Bedroom house, corner lot.	389,000	1750	3	2
◇	Ranch house, detached	0003	1950's ranch-style house, new kitchen, pool	295,000	1500	3	2
◇	Small detached house	0007	Small house downtown	280,000	900	2	1
◇	test			test			

Summary:
Custom built home home

House ID:
0006

Description:
Large custom home with pool and covered patio

Price:
420,000

Square Feet:
2300

Bedrooms:
4

Bathrooms:
0006

Address:
4415 Country Drive

City:
Sacramento

State:
CA

Zip Code:
95616

Organization:
Default Organization

Figure 17. NREC House Detail

Configuring the House Detail - Appraisals View

The House Appraisals view is designed to provide a list of appraisal details (date, appraisal value, appraiser) for a particular house. This is a one-to-many relationship between the Products entity (Houses) and the Appraisals entity. Each house can be associated with many appraisal records. Review the ER diagram in "Data Layer" on page 31 for more information about NREC data entities and their relationships.

The standard Siebel data model does not include an Appraisals entity. Therefore, assume that the NREC design team has decided to create a new business component for Appraisals, which will store its data in a standard one-to-many extension table. In the NREC example, you will use the 1:M extension table for S_PROD_INT, which is named S_PROD_INT_XM.

For more information about extension tables, see *Siebel Tools Reference*.

The NREC design team has also determined that a master-detail view is required to display this data. The view should display a detailed 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 as described in "Modify Existing Product Applets to Display NREC Attributes" on page 123. However, you will need to create a new applet to display the list of Appraisals in the bottom applet. See Figure 18 on page 143 to see the end result of the configuration work.

Displaying data from a 1:M extension table requires the following steps:

- "Creating the Appraisals Business Component" on page 134
- "Creating a Link Between Houses and Appraisals" on page 137
- "Adding the Appraisals Business Component to a Business Object" on page 138
- "Create a New Appraisals List Applet" on page 139
- "Creating the House Detail - Appraisals View" on page 141
- "Adding Additional Columns to the Opportunity Base Table" on page 146
- "Compiling and Testing" on page 143

About Standard 1:M Extension Tables

One-to-many extension tables are predefined tables that have one-to-many relationships with base tables. They have generic columns that you can use to store additional data. They allow you to extend the data model for your purposes and track entities that are not part of the standard Siebel data model. And because the extension tables themselves are already part of the data model, you do not need to modify the database schema.

When using a one-to-many extension table to store data, you use the TYPE column to group records. You will create a new business component for the entity you want to track. The business component must have a Type field that defaults to a unique value and a search specification that finds only those records that contain this value. For details of how this is accomplished, see "Creating the Appraisals Business Component." This practice allows you to use a single one-to-many extension table to store data for multiple business components. However, this practice should also be followed when the extension table is used by only one business component.

NOTE: There are more than 20 one-to-many extension tables in the standard data model. The names of one-to-many extension tables contain the suffix *_XM*.

Creating the Appraisals Business Component

To display data from a one-to-many extension table you must define a new business component with fields that map to the generic columns in the extension table (ATTRIB_01, ATTRIB_02, and so on), as well as three fields that provide the user key and map to the following columns:

- **PAR_ROW_ID.** This column should map to the foreign key field using in the one-to-many link.
- **NAME.** This column should map to the field being tracked in the business component. The value in the name needs to make the record unique for each parent record.
- **TYPE.** This column will be used to group records in the extension table. This column should be mapped to the Type field. You will set a default value for the Type field and then configure the business component to automatically search for those records in the extension table that contain this value.

NOTE: The combination of NAME, TYPE, and PAR_ROW_ID will be unique in order to satisfy the U1 index of the _XM table.

The following procedure walks you through creating the appraisals business component for the NREC example.

To create the appraisals business component

1. Choose File > New Object.

   The New Object Wizards dialog box appears.

2. Under the General tab, select the BusComp icon and then click OK.

   The New Business Component wizard appears.

3 Enter the following information, and then click Next.

Field	Value
Project	Product (SSE)
Name	Appraisals
Table	S_PROD_INT_XM

The Single Value Fields dialog box appears.

4 Enter the following fields and columns, and then click Finish.

Column in Base Table	Field Name	Description
ATTRIB_01	Date	Date the appraisal is done.
ATTRIB_02	Appraisal Value	Appraised value of the property.
ATTRIB_03	Comments	Comments.
PAR_ROW_ID	Par Row Id	Parent Row ID.
NAME	Name	Used to store the name of the Appraiser.
TYPE	Type	Used to group records. Will be set to a unique predefault value.

The Business Component Wizard creates the business component based on the information you entered.

To set the Search Specification for the Business Component and the Pre Default Value of the Type field

1 In the Object Explorer, expand the Business Component Object type.

The Business Component window appears in the Object List Editor.

2 In the Business Component window, find and select the Appraisals business component.

3 Add the following Search Specification property:

- `[TYPE]= ‘Appraisals’`

The business component will retrieve only those records in which the TYPE field contains the value Appraisals.

4 In the Object Explorer, select the Field object type (child of Business Component).

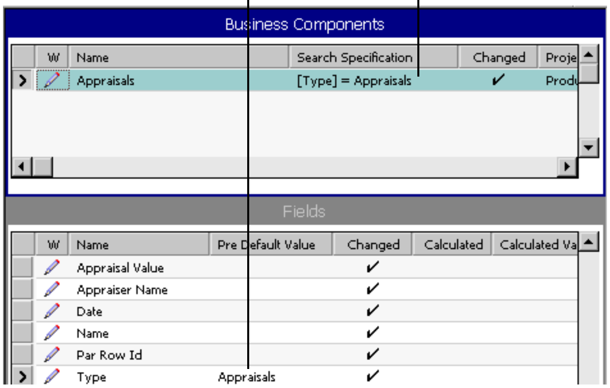
The list of fields that you created in the previous procedure appear in the Fields window of the Object List Editor.

5 Select the Type field and then enter `Appraisals` as the Pre Default Value.

The Type field will always default to Appraisals.

Pre Default Value for the Type field

Search Specification for the Appraisals business component



Business Components

W	Name	Search Specification	Changed	Proje
	Appraisals	[Type] = Appraisals	✓	Produ

Fields

W	Name	Pre Default Value	Changed	Calculated	Calculated Va
	Appraisal Value		✓		
	Appraiser Name		✓		
	Date		✓		
	Name		✓		
	Par Row Id		✓		
	Type	Appraisals	✓		

Creating a Link Between Houses and Appraisals

The relationship between Houses and Appraisals is one-to-many—for each house record, there can be many Appraisal records. The link object establishes the one-to-many relationship between the parent business component (Internal Products) and the child business component (Appraisals). You must create a Link object to establish this relationship. Once the relationship exists, you will be able to create a master-detail view to display the data.

To create a link between Internal Product and Appraisals

1. In the Object Explorer, select Link.
2. In the Links list applet, right-click and choose New Record.
3. Enter values for the properties as shown below.

Field	Example Value	Comment
Name	Internal Product/Appraisals	The convention is to use the name of the parent and child business component in the name of the link.
Project	NREC Configuration	Name of the project.
Parent Business Component	Internal Product	Name of the parent business component.
Child Business Component	Appraisals	Name of the child business component.
Destination Field	Par Row ID	Foreign key value in the child table.
Source Field	Row ID	Primary key value in the parent table.

Adding the Appraisals Business Component to a Business Object

Business objects group related business components together. They also gather the Link objects that associate two business components with one another. Generally, business objects correspond to a screen in Siebel Applications. For example, the Opportunities business object is analogous to the Opportunities screen.

Internal Products business object is the business object on which the Houses screen is based. You must associate the Appraisals business component that you create, as described in "Creating the Appraisals Business Component" on page 134, to the Internal Products business object.

To add the business component to a business object

1. In the Object Explorer, expand the Business Object object type.
2. In the Object List Editor, select the Internal Product business object.
3. In the Object Explorer, select the Business Object Component object type (child of business object).

   The Business Object Component window appears in the Object List Editor.
4. In the Business Object Component window, enter a new record using the following values:

Field	Value	Comments
BusComp	Appraisals	This is the business component you created as described in "Creating the Appraisals Business Component" on page 134.
Link	Internal Product/Appraisals	The link defines the master-detail relationship between the parent and child business components.

Create a New Appraisals List Applet

Now that you have configured the business object layer, you are ready to expose the appraisal data in the user interface. You will do this by creating a new applet. The type of applet called for in this case is a list applet. It will display a list of appraisal records for a single house record.

You can use the List Applet Wizard to create the applet. The wizard will make sure you define all the correct properties and will automatically map the necessary list columns and controls to the Applet Web Templates.

To create the appraisal list applet

1 Choose File > New Object.

The New Objects Wizard appears.

2 Under the Applets tab, select List Applet icon and then click OK.

3 In the General dialog box, enter the following and then click Next.

Field	Value
Project	NREC Configuration
Business Component	Appraisals
Applet Name	Appraisals List Applet
Display Title	Appraisals

4 In the Web Layout - General dialog box, select the Web templates to use, and then click Next.

Field	Value
Web template for base mode	Applet List (Base/EditList)
Web template for edit mode	Applet List Edit (Edit/New/Query)
Web template for edit list mode	Applet List (Base/EditList)

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

   - Date
   - Name
   - Appraisal Value

6. In the second Web Layout - Fields dialog box, leave the controls selected by default and then click Next.

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

   The New Applet Wizard creates a new list applet based on the information you entered and displays the applet in Edit Web Layout Mode.

8. You can preview the applet by right-clicking in the Layout window and then choosing Preview.

9. Choose File > Close.

Creating the House Detail - Appraisals View

Next you need to create a new view to display the Product Form applet and the Appraisals List applet. Use the New View wizard to walk you through the steps.

To create the Appraisals view

1 Choose File > New Object.

The New Object Wizard appears.

2 Under the general tab, select the View icon.

The New View Wizard opens.

3 In the New View dialog box, enter the following information and then click Next.

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

4 In the View Web Layout - Select Template dialog box, select the following Web template and then click next.

View Detail (Parent with Pointer)

5 In the Web Layout - Applets dialog box, select the following two 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 and the necessary supporting objects and opens the applet in the Edit Web Layout mode.

7 You can preview the applet by right-clicking in the Layout window and then choosing Preview.

8 Choose File > Close.

Adding the House Detail - Appraisals View to the Houses Screen

After you have created the view, you need to associate it with a screen. In this case, NREC will add the House Detail - Appraisals View to the Houses Screen.

See "Creating the House Detail View" on page 127 for the detailed procedure.

To associate the House Detail - Appraisals view to the Houses Screen

1 In the Object Explorer, expand the Screen object type.

The Screens window appears in the Object List Editor.

2 In the Screens window, select the Houses Screen.

3 In the Object Explorer, select the Screen View object type (child of Screen).

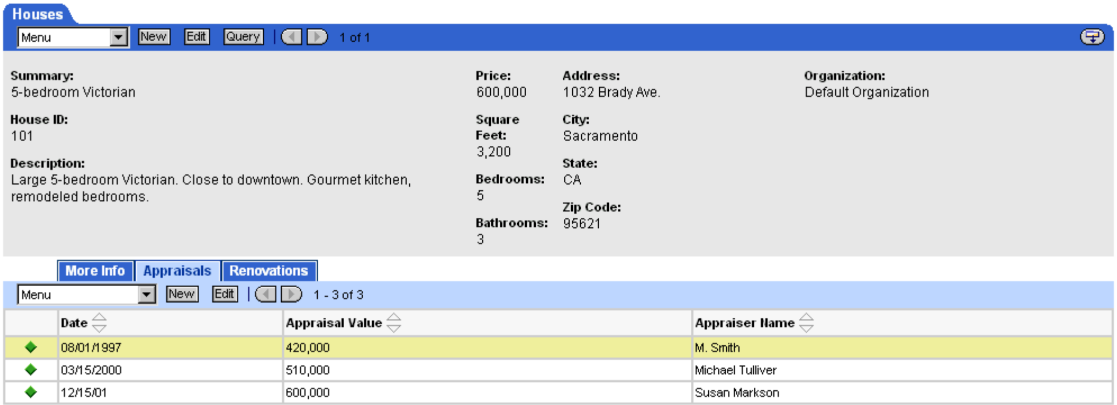
4 In the Screen View window, add the following record.

Field	Value
View	House Detail - Appraisals View
Menu Text	Appraisals
Viewbar Text	Appraisals

Compiling and Testing

After you have completed your configuration work, do the following:

- Compile your changes.
- Register the House Detail - Appraisals View in the application.
- Add House Detail - Appraisals View to a responsibility that you can use to test.
- Test your work.



Summary: 5-bedroom Victorian

House ID: 101

Description: Large 5-bedroom Victorian. Close to downtown. Gourmet kitchen, remodeled bedrooms.

Price: 600,000

Square Feet: 3,200

Bedrooms: 5

Bathrooms: 3

Address: 1032 Brady Ave.

City: Sacramento

State: CA

Zip Code: 95621

Organization: Default Organization

	Date	Appraisal Value	Appraiser Name
◆	08/01/1997	420,000	M. Smith
◆	03/15/2000	510,000	Michael Tulliver
◆	12/15/01	600,000	Susan Markson

Figure 18. House Detail - Appraisals View

Configuring the Opportunity Details View

NREC will use Partner Portal's standard Opportunity Details view to allow partner agents to enter and display information about opportunities. However, many of the attributes that NREC needs to track for each opportunity are not part of the standard Siebel data model. For example, the standard data model doesn't include the number of bedrooms or bathrooms that a potential buyer may be interested in. The additional attributes are:

- Price Range
- Square Feet
- Number of Bedrooms
- Number of Bathrooms
- ZIP Code

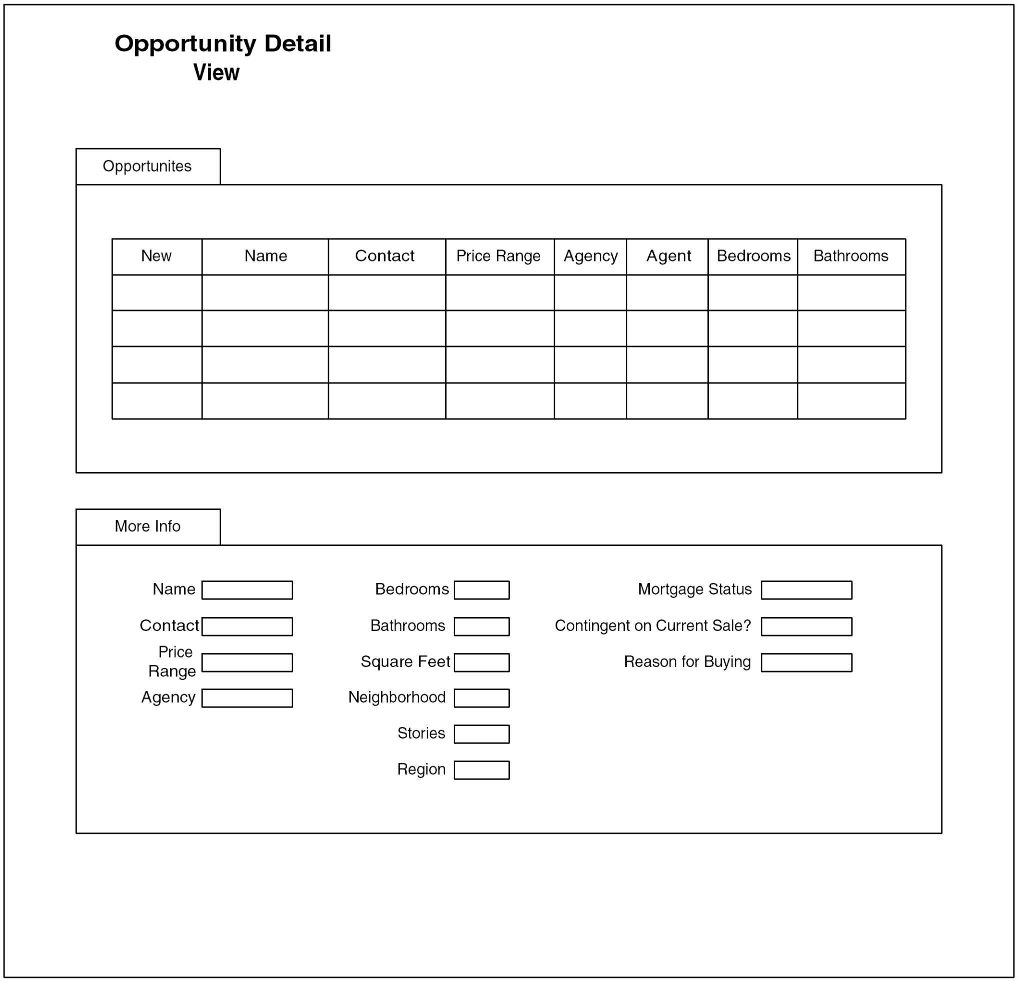
NREC will follow an approach similar to what they did for the Houses entity—add columns to the base table and then modify the objects at the business logic layer and user interface layer accordingly.

For detailed information about extending the database, see *Siebel Tools Reference*.

The high-level configuration tasks are:

- "Adding Additional Columns to the Opportunity Base Table" on page 146
- "Applying Schema Changes to Your Local Database" on page 118
- "Adding Fields to the Opportunities Business Component" on page 146
- "Modifying Applets to Display Additional Attributes" on page 147
- "Compiling and Testing" on page 151

Figure 19 shows a mock-up design for the Opportunity Detail View. The Opportunities Detail view is the default view for the Opportunities screen. It is a list-form view. The Opportunities List Applet is the master applet. The Opportunity Form applet is the detail applet. The mock up shows both the standard fields and the fields added for NREC.



New	Name	Contact	Price Range	Agency	Agent	Bedrooms	Bathrooms

Figure 19. Opportunity Detail View Mock Up

Adding Additional Columns to the Opportunity Base Table

The steps for adding columns to the base table are the same steps you followed to add additional columns to the internal products table. The procedure is covered here at a high-level. For detailed instructions, review the tasks covered in "Extending the Database by Adding New Columns to the Base Table" on page 117.

To add additional columns to the Opportunity base table

1 Navigate to the Opportunity base table S_OPTY.

2 Add the following column records:

Name	Type	Length
X_PRICE_RANGE	Varchar	20
X_SQUARE_FEET	Varchar	20
X_BEDROOMS	Varchar	20
X_BATHROOMS	Varchar	20
X_ZIP_CODE	Varchar	30

3 Apply the schema to the local database.

Adding Fields to the Opportunities Business Component

After extending the database to support NREC's additional attributes, you need to define the fields in the business object layer. You do this by adding the fields to the Opportunity business component. The fields will map to the extension columns in the S_OPTY. The steps covered in this procedure are at a high-level. For detailed instructions, review the procedures covered in "Configuring the Internal Product Business Component" on page 121.

To add fields to the Opportunity business component

1. Navigate to the Opportunity business component.
2. Lock the project locally, and then change the value of the Project property to NREC Configuration.
3. Add the following fields.

Name	Column
ZIP Code	X_ZIP_CODE
Price Range	X_PRICE_RANGE
Square Feet	X_SQUARE_FEET
Number of Bathrooms	X_BATHROOMS
Number of Bedrooms	X_BEDROOMS

These fields are now available to expose in the user interface.

Modifying Applets to Display Additional Attributes

Now that the fields are added to the business component, the next step is to expose them in the user interface. NREC's design is to modify the default applets that appear on the Opportunities Detail View to include the new fields. These two applets are:

- Opportunities List Applet (SCW)
- Opportunities Form Applet (SCW)

An alternative would be to create new applets. However, modifying existing objects is typically a preferable method because it requires less configuration work and causes less problems with future upgrades.

The following procedures are the high-level steps required for modifying the existing Opportunities list and form applets to display new fields added to the opportunities business component. As discussed in the previous chapter, the procedures differ slightly for form and list applets. See "Exposing Fields in the User Interface" on page 107.

To expose the new fields on the Opportunity List Applet (SCW)

1. Navigate to the Opportunities List Applet (SCW).
2. Add the following List Columns.

Field	Display Name	Show in list
ZIP Code	ZIP Code	FALSE
Price Range	Price Range	TRUE
Square Feet	SQFT	TRUE
Number of Bedrooms	Bedrooms	TRUE
Number of Bathrooms	Bathrooms	TRUE

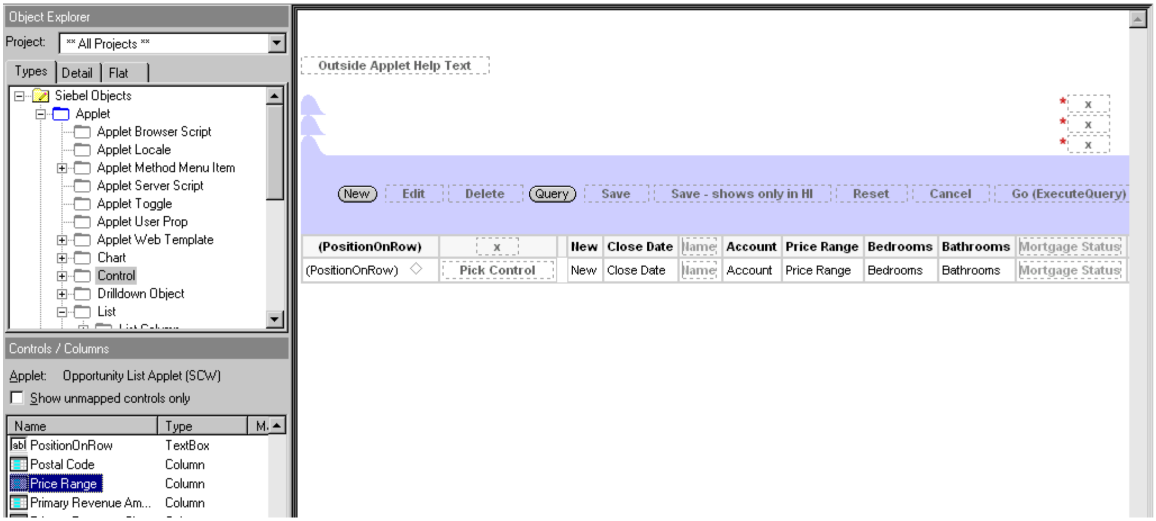
   The list columns are now available to map to applet Web templates. The Show in list property determines whether the field will appear by default in the list applet. If the Show in list property is not set to true, users can add it to the list applet at run-time by choosing Display Columns from the applet-level menu.

3. Use the Web Layout Editor, to delete the following fields from the applet Web template.
   - Revenue, Sales Stage, Committed, Lead Quality, Probability, Probability %, Reason Won/Lost, Currency, Created, Description, Site, Address, City, State, ZIP, and Country.

NOTE: Hold down the Ctrl key and select multiple controls on the applet Web template.

4 Map the list columns defined in Step 2 to the Web template.

5 Repeat Step 3 and Step 4 on page 149 for each applet mode.



Applet: Opportunity List Applet (SCW)

To expose new fields on the Opportunity Form Applet (SCW)

1 Navigate to the Opportunities Form Applet (SCW).

2 Add the following Controls.

Name	Caption	Field
ZIP Code	ZIP Code	ZIP Code
Price Range	Price Range	Price Range
Square Feet	SQFT	Square Feet
Number of Bedrooms	Bedrooms	Number of Bedrooms
Number of Bathrooms	Bathrooms	Number of Bathrooms

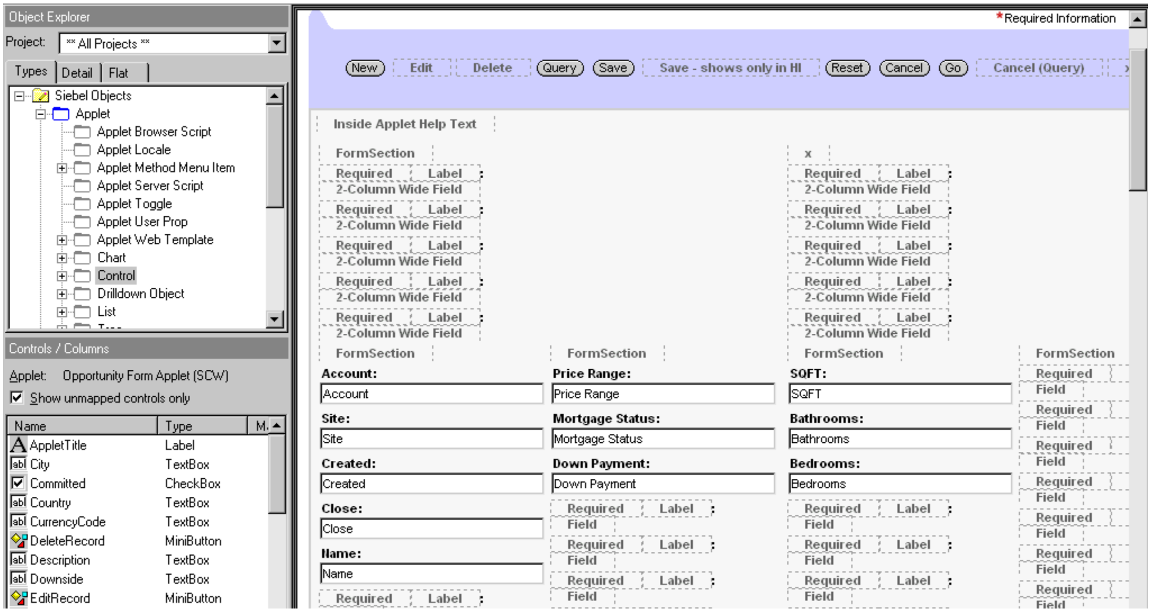
3 Using the Web Layout Editor, delete the following fields from the applet Web template.

- Site, Created, Address, City, State, ZIP, Country, Probability, Committed, Sales Team, Lead Quality, Sales Method, Sales Stage, Lead Source, Revenue, Expected Value, Upside, and Downside.

NOTE: Hold down the Ctrl key and select multiple controls on the applet Web template.

4 Map the controls you created in Step 2 on page 149 to the Web template.

5 Repeat Step 3 and Step 4 for each applet mode.



Project: ** All Projects **

Applet: Opportunity Form Applet (SCW)

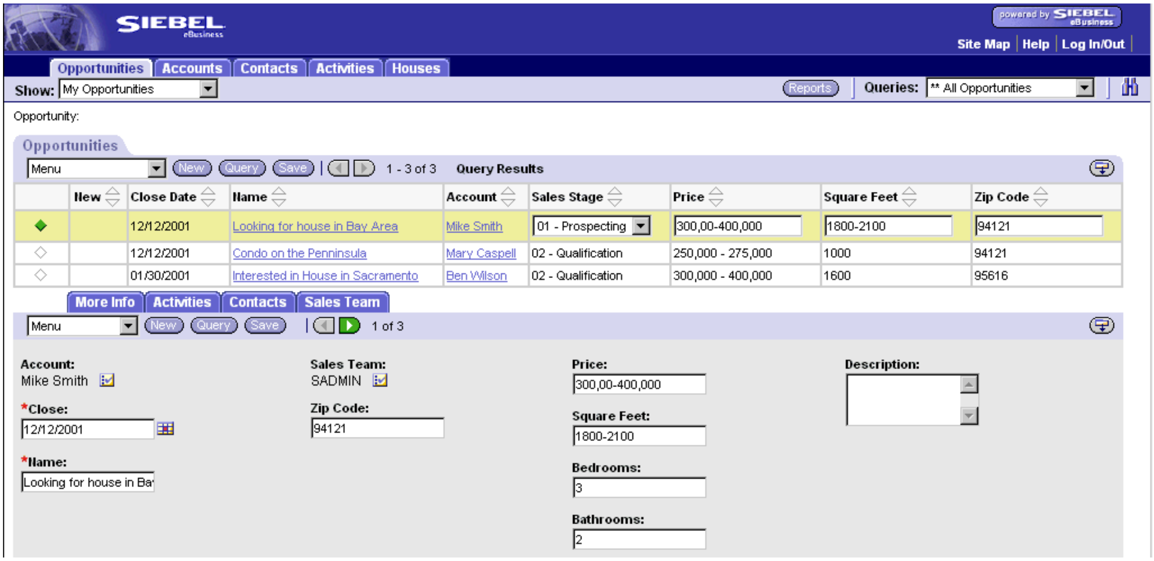
☑ Show unmapped controls only

Name	Type
AppletTitle	Label
City	TextBox
Committed	CheckBox
Country	TextBox
CurrencyCode	TextBox
DeleteRecord	MiniButton
Description	TextBox
Downside	TextBox
EditRecord	MiniButton

Account: Account
Price Range: Price Range
SQFT: SQFT
Site: Site
Mortgage Status: Mortgage Status
Bathrooms: Bathrooms
Created: Created
Down Payment: Down Payment
Bedrooms: Bedrooms
Close: Close
Name: Name

Compiling and Testing

After you have made your changes you are ready to compile and test your results. Figure 20 shows what the applets look like after being modified.


SIEBEL eBusiness
powered by SIEBEL eBusiness
Site Map | Help | Log In/Out
Opportunities | Accounts | Contacts | Activities | Houses
Reports
Opportunity:
Opportunities
Menu
New
Query
Save
Query Results
More Info | Activities | Contacts | Sales Team
Menu
New
Query
Save

Show: My Opportunities

Queries: ** All Opportunities

1 - 3 of 3

	New	Close Date	Name	Account	Sales Stage	Price	Square Feet	Zip Code
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting	300,00-400,000	1800-2100	94121
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification	250,000 - 275,000	1000	94121
◇		01/30/2001	Interested in House in Sacramento	Ben Wilson	02 - Qualification	300,000 - 400,000	1600	95616

1 of 3

Account: Mike Smith

*Close: 12/12/2001

*Name: Looking for house in Ba

Sales Team: SADMIN

Zip Code: 94121

Price: 300,00-400,000

Square Feet: 1800-2100

Bedrooms: 3

Bathrooms: 2

Description:

Figure 20. Modified Opportunity List and Form Applets

Pick Lists, Drilldowns, and MVGs 7

This chapter will take a second pass at configuring. You will work with some new objects as well as some of the objects that you configured in previous chapters. You can follow the tasks in this chapter using your sample database:

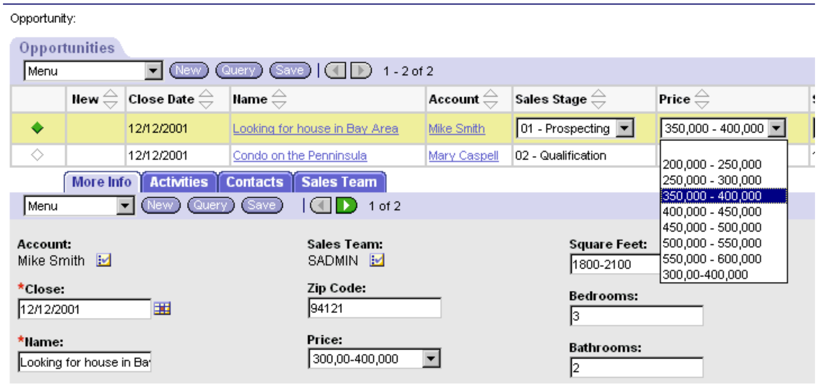
- Creating dynamic and static pick lists
- Creating drilldowns
- Creating MVGs

Configuring Pick Lists

NREC wants to take advantage of pick lists to standardize data entry and minimize mistakes. Pick lists allow users to populate a field by selecting values from a list rather than typing them in fields. There are two types of pick lists: static and dynamic. Static pick lists draw their data from the Siebel list of values table. The list of values table is maintained by an administrator. Dynamic pick lists draw their data from user-maintained tables.

Static Pick Lists

A static pick list is a list of predefined values that the user invokes from a field in an applet. When the user clicks the drop-down arrow to the right of the field, a single-column pick list appears. The user selects a value from the list, and then clicks Save to enter the value for the field. The values in the pick list are predefined by an administrator or developer and stored in the list of values table.



New	Close Date	Name	Account	Sales Stage	Price
◆	12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting	350,000 - 400,000
◇	12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification	

200,000 - 250,000
250,000 - 300,000
350,000 - 400,000
400,000 - 450,000
450,000 - 500,000
500,000 - 550,000
550,000 - 600,000
300,00-400,000

Account: Mike Smith
Sales Team: SADMIN
Square Feet: 1800-2100
*Close: 12/12/2001
Zip Code: 94121
Bedrooms: 3
*Name: Looking for house in Ba
Price: 300,00-400,000
Bathrooms: 2

Figure 21. Static Pick List

A pick list can be bounded or unbounded. A bounded pick list allows the user to select values from the list only. An unbounded pick list allows users to select values from the list or type values directly into the field.

NREC's requirement is to have pick lists for the following fields on the Opportunity business component.

- Price Range
- Square Feet
- Bedrooms
- Bathrooms

The following procedure uses the Price Range field on the Opportunities business component as an example. To complete the NREC configuration, you would create pick lists for the remaining fields as well.

To create a new static pick list

1 Choose File > New Object.

The New Objects Wizard appears.

2 Under the General tab, select the Pick List icon, and then click OK.

The Pick List wizard opens.

3 On Pick List dialog box, enter the following and then click Next:

Field	**Value**
Project	NREC Configuration
Business Component	Opportunity
Field	Range

The field is one you created in "Adding Additional Columns to the Opportunity Base Table" on page 146.

4 In the Pick List Definition dialog box, select the Static Pick List radio button and then click Next.

The Pick List Type dialog box displays all the controls that map to the field you selected in Step 3.

5 In the Pick List dialog box, select the New radio button and then click Next.

6. In the next Pick List Definition dialog box, enter Price Range as the name of the pick list, select the Create New List of Values radio button, and then click Next.

   If there is a list of values already defined for the pick list, the wizard allows you to select them here.

7. In the List of Values dialog box, enter a name for the list and then, for each value that you want to appear in the list, enter the value and then click Enter.

Field	Value
Name	Price
Value	100,000 - 150,000
	150,000 - 200,000
	200,000 - 250,000
	...and so on

8. In the next Pick List Definition dialog box, enter a comment in the Comment field and leave the Search Specification and Bounded Pick List check boxes blank.

   A bounded pick list forces the user to enter a value that's been predefined on the list. Not selecting the Bounded check box allows the user to type in a value or select one from the list.

9. In the Finish dialog box, review the information and then click Finish.

   The wizard creates the Pick List object and related objects and adds the list of values to the database.

10. Compile and Test.

    The pick list appears as a drop-down list on the Price fields in the user interface. See Figure 21 on page 154. Notice that the pick list is available on the Price field in both the list and the form applet.

Dynamic Pick Lists

Like static pick lists, dynamic pick lists allow the user to populate fields by selecting values from a list. However, rather than drawing the values from the list of values table, a dynamic pick list draws its values from another user-maintained business component. Fields that use dynamic pick lists are typically joined fields displaying data from a table other than the business component's base table. The dynamic pick list allows users to update the joined field.

NREC design includes a dynamic pick list on the Activity Detail Applet. The opportunity field that you exposed on the applet (see "Adding the Opportunity Field to a Activity Detail Applet (SCW)" on page 109) is a joined field. Creating a dynamic pick list on this field will allow users to update the activity record by pulling in the opportunity name from the Opportunity table.

The Pick List Wizard walks you through the process of creating a dynamic pick list and related objects, which include:

- **Pick List.** Object that defines the properties of the pick list, including the originating business component and the pick business component.

  NOTE: The originating business component is the one on which you are creating the dynamic pick list. In the current example, it is the Action business component. The pick business component is the one from which you are picking values to display to the user. In the current example, it is the Opportunity business component.

- **Pick Maps**. Child object of a business component field that map the source field in the pick business component with the target field in the originating business component.
- **Pick Applet.** Pop-up applet that allows you to display the a list of records from which the user can select.

Before creating a dynamic pick list you must lock the projects for both the pick business component and the originating business component. For the current example lock the Action project locally and then change the value of the Project property to NREC Configuration. In previous chapters, you should already have associated the Opportunity business component with the NREC Configuration project, which should be locked.

To create a dynamic pick list

1. Choose File > New Object.
2. Under the General Tab, select the Pick List icon, and then click OK.
3. In the Pick List dialog box, enter the following.

Field	Value	Description
Project	Activity	Project to which the pick list will belong.
Business Component	Action	Business component in the project that contains the field for which you are defining the pick list.
Field	Opportunity	Field for which you are defining the pick list.

4. In the Pick List Type dialog box, select the Dynamic radio button.
5. In the Pick List Definition dialog box, select the Create new Pick List radio button.
6. In the next Pick List Definition dialog box, enter the following values and then click Next:

Field	Value	Description
Pick Business Component (business component you want to base the pick list on)	Opportunity	This is the business component from which you are drawing values to display to the user.
Field to sort by	Created	Records are sorted by this field in the pick applet.
Name	Opportunity NREC	Name of the Pick List.

7. In the Pick List Specifications dialog box, accept the defaults; leave all of the check boxes unselected.

8 In the Pick Map dialog box, use the drop-down lists to select the following and then click Add:

Field in Originating BusComp	Field in Pick BusComp
Opportunity	Opportunity

This information is used to define the Pick Maps for the Pick List. They are the mappings between the source field and the target field.

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

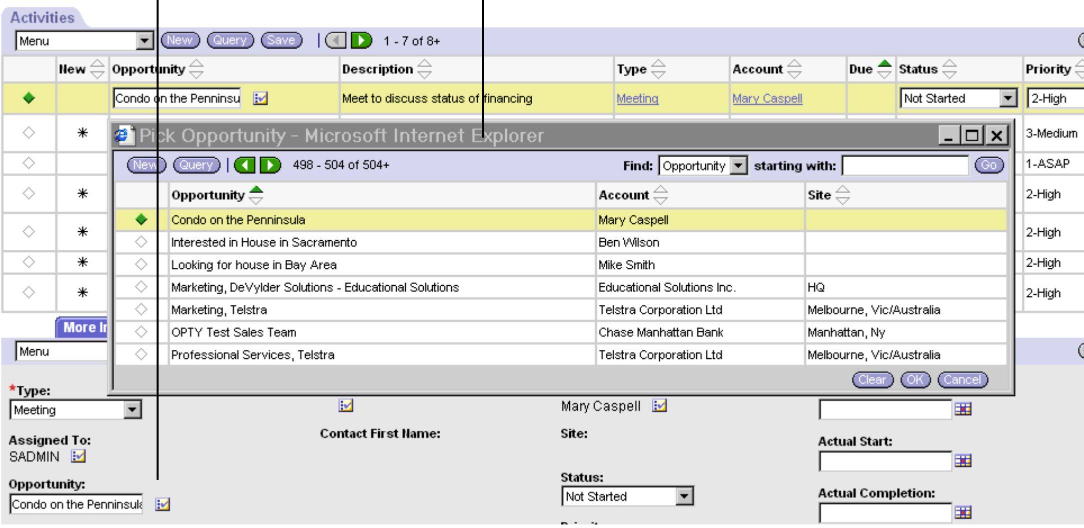
The Pick List wizard creates the pick list, pick maps, and checks to see if a pick applet already exists that can be used to display the records. For the NREC example, there is already an Opportunity pick applet defined, so the New Applet Wizard does not open. If an appropriate pick applet did not exist, then the New Applet Wizard automatically opens.

10 Compile and test.

Be sure to compile the Activity project.

Dynamic pick list on the Opportunity field

Opportunity Pick Applet displays records from the Opportunity business component



New	Opportunity	Description	Type	Account	Due	Status	Priority
	Condo on the Penninsu	Meet to discuss status of financing	Meeting	Mary Caspell		Not Started	2-High
*							3-Medium
							1-ASAP
*							2-High
*							2-High
*							2-High
*							2-High

Opportunity	Account	Site
Condo on the Penninsula	Mary Caspell	
Interested in House in Sacramento	Ben Wilson	
Looking for house in Bay Area	Mike Smith	
Marketing, DeVylder Solutions - Educational Solutions	Educational Solutions Inc.	HQ
Marketing, Telstra	Telstra Corporation Ltd	Melbourne, Vic/Australia
OPTY Test Sales Team	Chase Manhattan Bank	Manhattan, Ny
Professional Services, Telstra	Telstra Corporation Ltd	Melbourne, Vic/Australia

*Type: Meeting

Assigned To: SADMIN

Opportunity: Condo on the Penninsula

Contact First Name:

Mary Caspell

Site:

Status: Not Started

Actual Start:

Actual Completion:

Constraining a Pick List

You can filter a pick applet to display only records that have field values that match corresponding fields in the originating business component's records. For example, you can constrain the Opportunity pick applet, configured in the previous section, to display only the opportunities for the account associated with the activity.

To constrain a pick list you create another pick map with a Constrain property set to TRUE. This pick map defines fields on the originating business component and a field on the pick business component. These two fields must match for the record to be displayed in the pick applet. You can create more than one constrain pick map. Constrain pick maps do not copy the values of the field from the originating business component to the pick business component. They serve as a filter to determine which records will appear in the pick applet.

Following the NREC example, you will constrain the list of records displayed in the Opportunity Pick Applet based on the Account field.

To create a constrain pick map

1. In the Object Explorer, expand the Business Component object type.
2. In the Object List Editor, select the Action business component.

   The Action business component is the originating business component.

   NOTE: The Opportunity business component is the pick business component.

3. In the Object Explorer, expand the Field object type (child of Business Component).
4. In the Object List Editor, select the Opportunity field.

   The Opportunity Field is the field from which the pick applet is invoked.

5. In the Object Explorer, select the Pick Map object type (child of Field).

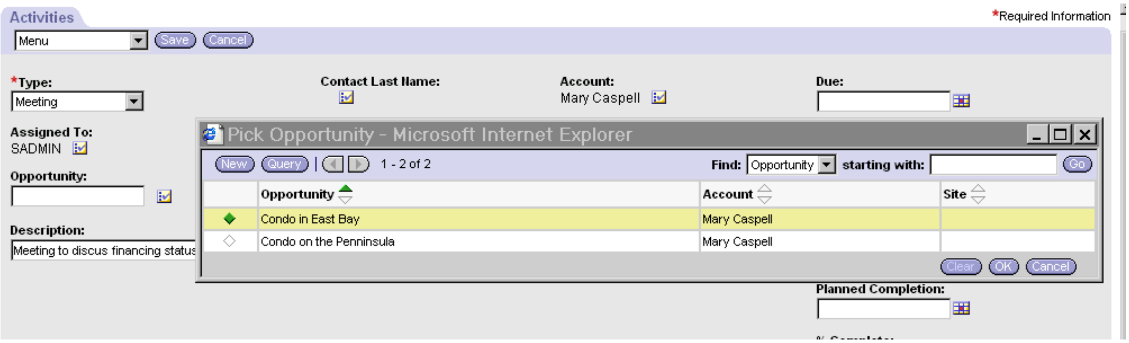
6 In the Object List Editor, enter the following record.

Field	Pick List Field	Constrain
Account Name	Account	TRUE

The Account Name field on the originating business component must match the Account field on the pick list business component for the record to be displayed in the pick applet.

7 Compile and test.

NOTE: Notice that the only opportunities that appear are those associated with the Account record for Mary Caspell. To verify your results you will need to enter test data. If no account record is selected for the activity, then no opportunities will appear in the pick list.



*Type: Meeting

Assigned To: SADMIN

Account: Mary Caspell

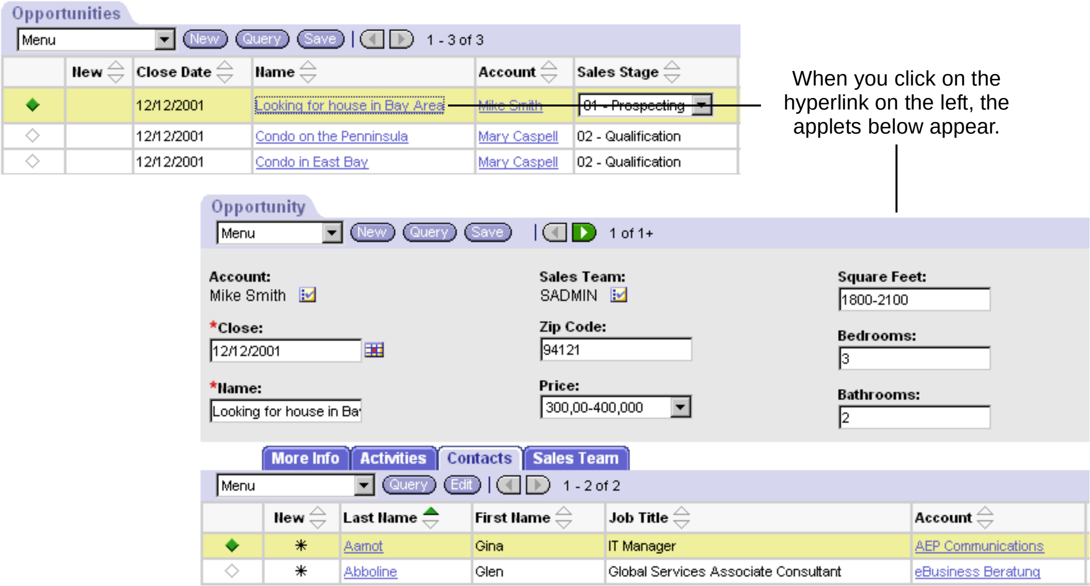
Description: Meeting to discus financing status

Find: Opportunity

	Opportunity	Account	Site
◆	Condo in East Bay	Mary Caspell	
◇	Condo on the Penninsula	Mary Caspell	

Creating Drilldowns

Drilldowns allow users to click a field and be taken to another view that displays more information about the field. Fields with drilldowns appear as hyperlinks in the user interface. For example, when you click on the Name field in the Opportunity List Applet, you drill down to the Opportunity Detail - Contacts view as shown in Figure 22.



	New	Close Date	Name	Account	Sales Stage
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification
◇		12/12/2001	Condo in East Bay	Mary Caspell	02 - Qualification

When you click on the hyperlink on the left, the applets below appear.

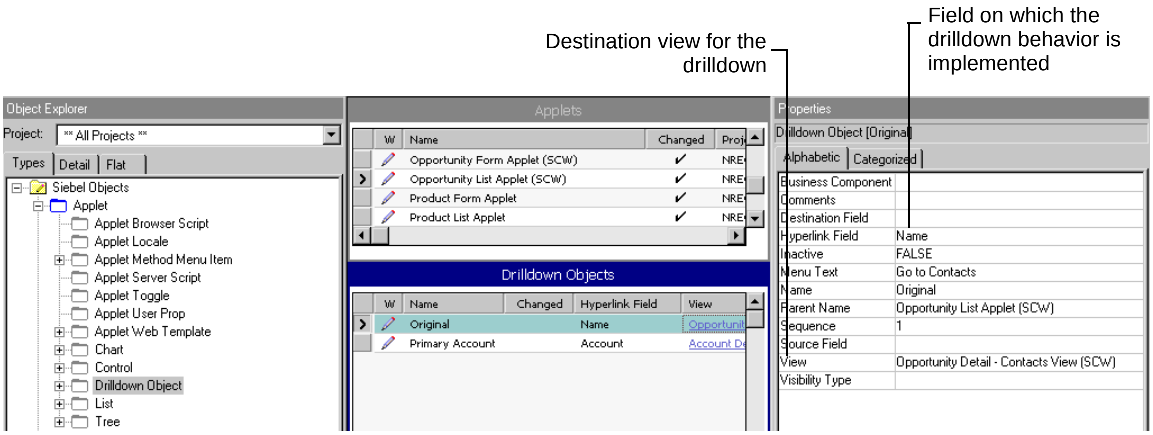
Account: Mike Smith
Sales Team: SADMIN
Square Feet: 1800-2100
*Close: 12/12/2001
Zip Code: 94121
Bedrooms: 3
*Name: Looking for house in Ba
Price: 300,00-400,000
Bathrooms: 2

	New	Last Name	First Name	Job Title	Account
◆	*	Aamot	Gina	IT Manager	AEP Communications
◇	*	Abboline	Glen	Global Services Associate Consultant	eBusiness Beratung

Figure 22. Drilldown Example

Drilldowns can be either static or dynamic. A static drilldown always takes the user to the same view. A dynamic drilldown can take the user different views depending on certain conditions, such as the value of a field.

You configure drilldowns in Siebel Tools. The Drilldown object type is a child object of the Applet object type. It defines the field on which the drilldown behavior is to be implemented and the view that appears when the user clicks the field. For example the drilldown object definition for the Name field of the Opportunity List applet as shown in Figure 23.



Destination view for the drilldown

Field on which the drilldown behavior is implemented

W	Name	Changed	Proj
	Opportunity Form Applet (SCW)	✓	NRE
	Opportunity List Applet (SCW)	✓	NRE
	Product Form Applet	✓	NRE
	Product List Applet	✓	NRE

W	Name	Changed	Hyperlink Field	View
	Original		Name	Opportunit
	Primary Account		Account	Account De

Business Component	
Comments	
Destination Field	
Hyperlink Field	Name
Inactive	FALSE
Menu Text	Go to Contacts
Name	Original
Parent Name	Opportunity List Applet (SCW)
Sequence	1
Source Field	
View	Opportunity Detail - Contacts View (SCW)
Visibility Type	

Figure 23. Static Drilldown

Creating a Dynamic Drilldown

A dynamic drilldown takes a user to a different view depending upon certain conditions, such as the value of a field. For example, NREC wants to configure the drilldown shown in Figure 22 on page 163 so that it is dynamic. The requirement is the following:

- If the opportunity sales stage is *Prospecting, Qualification, or Closing,* go to the Opportunity Detail - Activities view.
- Otherwise, go to the Opportunity Detail - Contacts view. Note that this is the behavior of the existing drilldown object.

Dynamic drilldowns require one or more dynamic drilldown destination object types. Dynamic drilldown destination objects define the conditions that determine when to go to a particular drilldown object. They are child objects of a drilldown object.

For a dynamic drilldown, you define each candidate view by creating a drilldown object. You express the conditions under which each drilldown object should be activated by defining one or more dynamic drilldown destinations. Dynamic drilldown destinations are defined as child objects of the default drilldown object. When all conditions expressed in the dynamic drilldown destinations are false, the parent drilldown acts as the default.

For the current example, you need to do the following:

- Define a drilldown object for the Opportunity Detail - Activities view.
- Define destination drilldown objects as children of the *Original* drilldown object.

NOTE: Original is the existing drill drilldown object. It displays the Opportunity Detail - Contact view. This drilldown will serve as the default when all conditions expressed in child drilldown destination objects are false.

To create a dynamic drilldown object

1. In the Object Explorer, expand the Applet object type.
2. In the Object List Editor, select the Opportunity List Applet (SCW).

   This is the applet from which you want the user to be able to drill down.
3. In the Object Explorer, select the Drilldown Object object type.
4. In the Drilldown Object list, enter a new record using the following values:

Field	Values for First Drilldown
Name	Activities
Hyperlink Field	Name
View	Opportunity Detail - Activities (SCW)

To create Drilldown Destination Objects

1. Select the drilldown object that you want to serve as the parent (default) of your dynamic drilldown destinations.

   For the current example, select the drilldown object named *Original*.

   The target view defined in this drilldown object will serve as the default for when all conditions in the Drilldown Destination objects are false.

2. In the Object Explorer, select Dynamic Drilldown Destination object type (a child object of Drilldown Object).

3. In the Dynamic Drilldown Destinations applet, enter the following records:

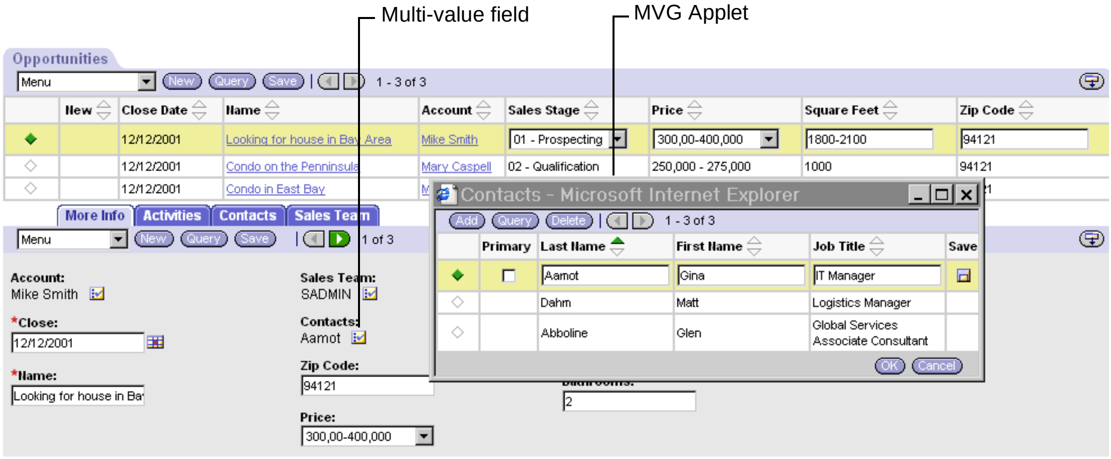
Name	Filed	Value	Destination Drilldown Object	Sequence
Prospecting	Sales Stage	01 - Prospecting	Activities	1
Qualification	Sales Stage	02 - Qualification	Activities	2
Closing	Sales Stage	03 - Closing	Activities	3

4. Compile and test.

   Navigate to the Opportunity List Applet. When you drill down on the name column it takes you to Opportunity Detail - Contacts (SCW) view or Opportunity Detail - Activities (SCW) depending on the value of the Sales Stage field.

Configuring Multi-Value Groups

Multi-value groups (MVGs) allow you to incorporate child data into an applet. They provide you with a way to display multiple child records for a single parent record, yet they do not require a master-detail view. Instead, the user clicks the Select icon in a multi value field, and an MVG applet pops up displaying a set of child records. For example, NREC's requirement is to create a MVG on the Opportunity Form Applet (SCW) to display the contacts associated with each opportunity. The results of the configuration work are shown in Figure 24.



	New	Close Date	Name	Account	Sales Stage	Price	Square Feet	Zip Code
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting	300,00-400,000	1800-2100	94121
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification	250,000 - 275,000	1000	94121
◇		12/12/2001	Condo in East Bay	M				1

Account:
Mike Smith

*Close:
12/12/2001

*Name:
Looking for house in Ba

Sales Team:
SADMIN

Contacts:
Aamot

Zip Code:
94121

Price:
300,00-400,000

2

	Primary	Last Name	First Name	Job Title	Save
◆	□	Aamot	Gina	IT Manager	
◇		Dahm	Matt	Logistics Manager	
◇		Abboline	Glen	Global Services Associate Consultant	

Figure 24. Multi-Value Field

Creating an MVG

You can use the MVG Wizard to create the objects necessary for MVGs. Based on the information you enter, the Wizard creates the following objects that are necessary to implement the MVG:

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

To configure a multi value field using the MVG Wizard

1. Choose File > New Object.

   The New Object Wizards dialog box appears.

2. Under the General tab, click the MVG icon and then click OK.

   The Multi Value Group wizard appears.

3. In the Multi Value Group dialog box, enter the following and then click Next:

Field	Value	Description
Project	NREC Configuration	Project to which the MVG will belong.
Master Business Component	Opportunity	The name of the master business component. For example Opportunity is the parent business component in the Opportunity/Contact Link object.

4. In the second Multi Value Group dialog box, enter the following and then click Next:

Field	Value	Description
Detail Business Component	Contact	This is the name of the detail business component. For example, Contact is the child business component defined in the Opportunity/ Contact link object.
Multi Value Link Name	NREC Contact	Unique name for the Multi Value Link object.

5. In the Direct Links dialog box, select the Opportunity/Contact link and then click Next.

   This dialog box displays all links that describe a master-detail relationship between the business components defined in Step 3 and Step 4.

6 In the Primary ID dialog box, leave all options unselected.

NOTE: Although this procedure does not cover implementing primaries, doing so can have performance benefits by reducing the number of SQL queries that are executed to populate the MVG for each record.

---

7 In the Multi Value Link dialog box, accept the default for Multi Value Link properties; leave all the check boxes unselected.

8 In the Multi Value Field dialog box, use the drop-down list to select a field from the detail business component to use to create the multi value field in the master business component.

a From the drop-down list, select the Last Name field.

b Use the default value of NREC Contact Last Name for the Multi Value Field Name.

c Click Add and then Click Next.

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

The Multi Value Group Wizard uses the information you entered to create the necessary objects for a multi value group. It also searches the repository for an existing multi value group applet that can be used to display the data in the user interface. If a suitable applet is not found, the MVG Applet Wizard opens allowing you to define one.

In the current example, the Contacts MVG Applet already exists, so the MVG Applet wizard does not open.

Exposing the MVG in the User Interface

After you create the underlying objects necessary to display the MVG, you can expose the multi value field in the user interface.

To expose an MVG in the user interface

1 Select the Opportunity Form (SCW) applet.

2 Create a List Column object, accepting the default values.

3 Add List Column object type based on the multi value field you created in the previous procedure.

For the current example:

Field	Value
Field	NREC Contact Last Name
Display Name	Contacts
MVG Applet	Contact MVG Applet
Runtime	True

NOTE: The Contacts MVG is suitable for NREC's purposes. You could also have created a new MVG applet using the MVG Applet Wizard.

4 Map the new List Column to the Web templates for each mode.

5 Compile and test.

The multi value field appears in the applet with a Select button next to it. When the user clicks the select button, the MVG applet appears. See Figure 24 on page 167.

Creating a Virtual Business Component 8

This chapter walks you through creating a virtual business component. It follows the NREC example, building on work you did in Chapter 6, “Configuring the House and Opportunity Entities,” for configuring the House entity.

Read this chapter for a general overview of virtual business components and the steps for configuring them. Keep in mind that this example is relatively simple. It uses a simple file as an external data source rather than an external database or external application.

For more information about how virtual business components can be used to integrate external data, see *Overview: Siebel eBusiness Application Integration Volume I* and *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

Understanding Virtual Business Components

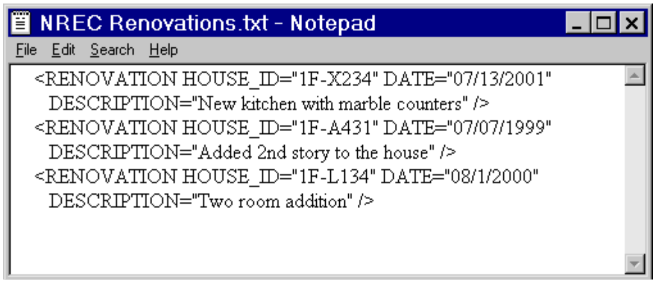
Virtual business components (VBCs) allow you to bring data from external data sources into your Siebel application user interface. For example, you could create a VBC to represent data from data sources such as:

- Back office applications
- Legacy applications
- Transaction services
- Web sites

Virtual business components are based on business services. Business services determine the behavior of the VBC and define how the VBC manipulates data. You can base VBCs on predefined business services provided by Siebel Systems, such as the XML Gateway, or you can create your own business service. You create both virtual business components and business services in Siebel Tools.

NREC's Virtual Business Component

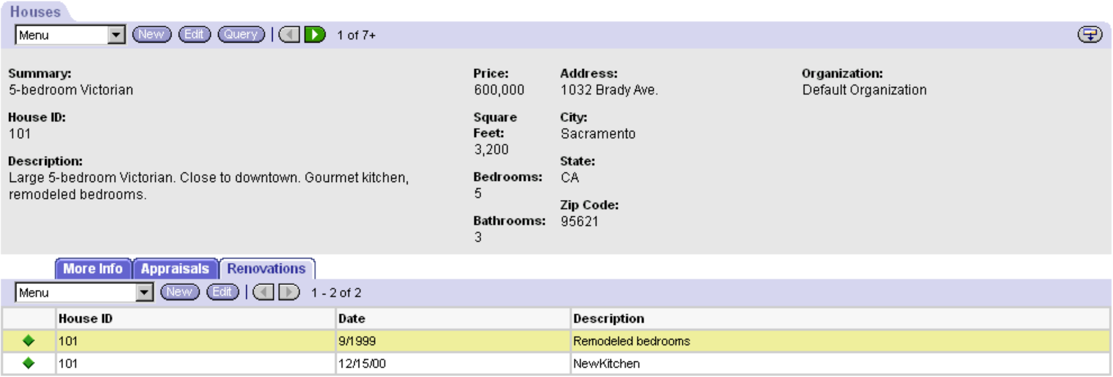
Assume that NREC stores house renovation data in a comma-delimited text file that a third-party vendor provides to NREC. NREC's design includes a virtual business component used to retrieve the data from the file and display it in the Partner Portal application. It will allow partner agents and NREC employees to see a history of renovation information for each house stored in NREC's Siebel database. Figure 25 shows NREC's external file of renovation data. Figure 26 shows how the data from this file is displayed in the user interface.



```
<RENOVATION HOUSE_ID="1F-X234" DATE="07/13/2001"
  DESCRIPTION="New kitchen with marble counters" />
<RENOVATION HOUSE_ID="1F-A431" DATE="07/07/1999"
  DESCRIPTION="Added 2nd story to the house" />
<RENOVATION HOUSE_ID="1F-L134" DATE="08/1/2000"
  DESCRIPTION="Two room addition" />
```

Figure 25. NREC's External File of Renovation Data

NREC's requirement is to display this data in a standard master-detail view showing a house record in the top applet and a list of renovations in the bottom applet.



Summary:
5-bedroom Victorian

House ID:
101

Description:
Large 5-bedroom Victorian. Close to downtown. Gourmet kitchen, remodeled bedrooms.

Price:
600,000

Square Feet:
3,200

Bedrooms:
5

Bathrooms:
3

Address:
1032 Brady Ave.

City:
Sacramento

State:
CA

Zip Code:
95621

Organization:
Default Organization

	House ID	Date	Description
◆	101	9/1999	Remodeled bedrooms
◆	101	12/15/00	NewKitchen

Figure 26. Applet Displaying Data from a VBC

The steps for configuring a virtual business component for the NREC's scenario are:

- "Creating a Business Service" on page 175
- "Creating a Virtual Business Component" on page 179
- "Creating Fields for the Virtual Business Component" on page 180
- "Defining User Properties for the Virtual Business Component" on page 181
- "Creating a Link" on page 182
- "Updating the Business Object" on page 183
- "Exposing the Fields in the User Interface" on page 184
- "Creating a New View" on page 185

Creating a Business Service

Business services are objects that define sets of functionality. For example, a business service can perform tasks such as moving data or converting data formats. Like business components, business services are objects stored in the Siebel repository. However, rather than interacting with tables and data as business components do, business services interact with other objects.

Virtual business components use business services to provide the functionality that allows them to manipulate external data. Business services define the methods, properties, and states that determine the behavior of virtual business components.

There are many predefined business services provided for specialized needs, such as using XML to display data in a Siebel application. You can also create your own business services. In the NREC example, you will use Siebel Tools to create a business service that will read and write data to a comma-delimited text file.

For information more information about business services, see *Siebel Object Interfaces Reference* and *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

In the NREC example, the steps for creating the business service are:

- "Defining the Business Service" on page 176
- "Defining Business Service Scripts" on page 176

Defining the Business Service

You create business service objects using Siebel Tools.

To define a business service

1. In the Object Explorer, expand the Business Service object type.
2. In the Business Services window, enter a new record with the following properties:

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

Defining Business Service Scripts

After defining the Business Service, you can define the business service scripts that actually do the processing associated the with the NREC External Text Handler business service object. Sample code for this example is shown in "Code Sample" on page 188.

Business Service methods to implement are listed in Table 9.

Table 9. Business Service Methods

Method	Custom Method Name in Code Sample	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 into the external data source.

Table 9. Business Service Methods

Method	Custom Method Name in Code Sample	Description
PreInsert	Pal_PreInsert	Provides any default values for new rows. In the current example, there are no default values.
Query	Pal_Query	Queries the external datasource and returns rows that match a given value. In the current example, the method queries the text file and returns rows that have a matching value for the current House_ID field.
Service_PreInvokeMethod	Service_PreInvokeMethod	Handles requests that come into the business service and calls other functions as appropriate.
Update	Not used in sample code	Updates an existing row in an external datasource.
Delete	Not used in sample code	Deletes a row in the external datasource.

For more information about business service methods, see *Siebel Object Interfaces Reference*.

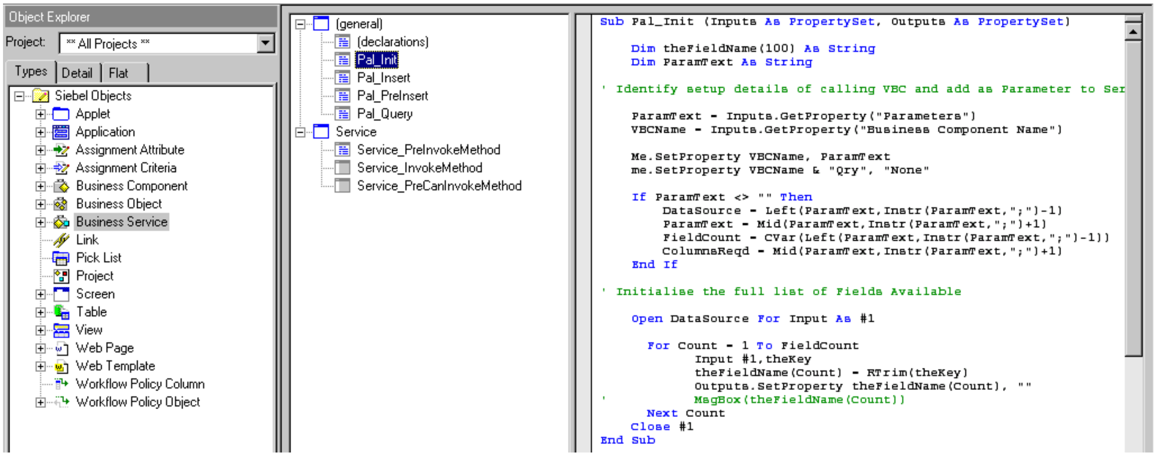
To define and write the business service script for NREC External Text Handler

1. Select the NREC External Text Handler business service.
2. With the business service selected, right-click, and then choose Edit Server Scripts.
3. Select Visual Basic as the scripting language and then click OK.

   The script editor appears with Service_PreInvokedMethod selected as the service.

4 Enter the custom functions by doing the following:

- **a** Expand the (general) icon in the script editor and then select the (declarations) icon.
- **b** In the text entry window, enter the code for the custom function.
- **c** After finishing the function and saving your work, an icon representing the function appears in the left pane.
- **d** Repeat steps Step a on page 178 through Step c for each custom function.

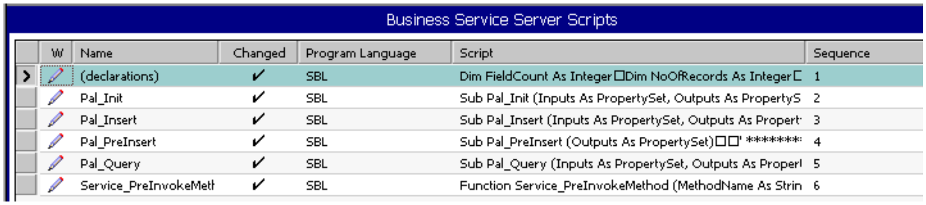


```
Sub Pal_Init (Inputs As PropertySet, Outputs As PropertySet)

    Dim theFieldName(100) As String
    Dim ParamText As String

' Identify setup details of calling VBC and add as Parameter to Ser

    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

' Initialise 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), ""
'           MsgBox(theFieldName(Count))
      Next Count
    Close #1
End Sub
```

5 Select the Service_PreInvokeMethod icon and then enter the code in text entry window.

6 Select the (declarations) icon and then type the code in the text entry window.

After saving your work, Business Service Server Script objects are automatically created.



Business Service Server Scripts

	W	Name	Changed	Program Language	Script	Sequence
>		(declarations)	✓	SBL	Dim FieldCount As Integer□Dim NoOfRecords As Integer□	1
		Pal_Init	✓	SBL	Sub Pal_Init (Inputs As PropertySet, Outputs As PropertyS	2
		Pal_Insert	✓	SBL	Sub Pal_Insert (Inputs As PropertySet, Outputs As Propert	3
		Pal_PreInsert	✓	SBL	Sub Pal_PreInsert (Outputs As PropertySet)□□' *******	4
		Pal_Query	✓	SBL	Sub Pal_Query (Inputs As PropertySet, Outputs As Proper	5
		Service_PreInvokeMeth	✓	SBL	Function Service_PreInvokeMethod (MethodName As Strin	6

Creating a Virtual Business Component

When creating a virtual business component you do not use the Business Component Wizard because it forces you to select a table on which the business component should be based and does not set the class to CSSBCVExtern.

To create a virtual business component

1. In the Object Explorer, select the Business Component object type.
2. Right-click in the Object List Editor and choose New Record.
3. Enter Values for the following properties.

Field	Value	Comments
Name	NREC Renovations VBC	Unique name for the virtual business component
Project	NREC Configuration	Choose a locked project
Class	CSSBCVExtern	Provides the virtual business component functionality

Creating Fields for the Virtual Business Component

After creating the virtual business component you need to create the fields that you want to display in the Siebel application. These fields will display data retrieved from the House Renovations file shown in Figure 25 on page 173.

NREC's requirement is to define the following fields for the business component:

- House_ID
- Date
- Description

To add fields to the virtual business component

1. In the Object Explorer, expand the Business Component Object type.
2. In the Object List Editor, select NREC Renovations VBC.
3. In the Object Explorer, select Single Value Field object type (child of Business Component).
4. Enter each of the following records.

Name	**Type**	**Description**
House_ID	DTYPE_TEXT	House Identification Number
Date	DTYPE_TEXT	Date the renovation was recorded
Description	DTYPE_TEXT	Description of the renovation

Defining User Properties for the Virtual Business Component

The user properties required by the NREC Renovations VBC are Service Name and Service Parameters. The Service Name user property defines which business service the VBC will invoke. The Service Parameters define the parameters that are passed into the business service upon initialization. In this example, the parameters are defined in the script attached to the business service. See the Pal_Init function in the "Code Sample" on page 188.

To define virtual business component user properties

1. Select the NREC Renovations VBC business component.
2. In the Object Explorer, select the Business Component User Properties object type (child of business component).
3. Add the following user properties records.

User Property Name	Value	Comments
Service Name	NREC External Text Handler	Name of the business service.
Service Parameter	D:\NREC\nrec.txt;3;House_ID, Date,Description	The script for the current example specifies the service parameters as the full path to the file name, number of columns, and column names. The parameters must be separated by semicolons, and there can be no spaces in the comma delimited strings. Service parameters will vary depending on the business service being used.

Creating a Link

The relationship between houses and house renovations is one to many; there can be many renovations recorded for one house. Therefore, you need a link to create the parent-child relationship between the Internal Product business component and the NREC Renovations VBC business component. Internal Product is the business component that NREC is using to store houses. See NREC’s entity relationship diagram shown in Figure 2 on page 31.

To create a link between parent and child business components

1 Select the Link object type.

2 In the Links window, add a new record using the following values:

Field	Value	Description
Name	Internal Product/NREC Renovations VBC	Unique name for the link.
Project	NREC Configuration	Project to which the object belongs.
Parent Business Component	Internal Product	Parent business component.
Child Business Component	NREC Renovations VBC	Child Business component.
Source Field	Part #	Unique ID on the parent record. Often this field will be Row Id. In this case, the unique field is a manually entered value.
Destination Field	House_ID	Column in child record that stores the parent ID.

Updating the Business Object

Like any business component, a virtual business component must be grouped together with objects of similar focus using a business object. Now that you have a link defined for the Internal Product/NREC Renovations VBC relationship, you need to add the Renovation VBC business component to the Internal Product business object.

To add the Renovations business component to the Internal Product business object

1. Expand the Business Object object type.
2. Select the Internal Product business object.
3. Select the Business Object Component object type (child of business object).
4. In the Business Object Components window, add a new record:

Field	Value
Business Component	NREC Renovations VBC
Link	Internal Product/NREC Renovations VBC

The NREC Renovations VBC business component is added to the Internal Product business object.

Exposing the Fields in the User Interface

Once the business object layer has been configured, you can expose the fields defined in the NREC Renovations VBC in the user interface. The steps to do this include:

- "Creating a New List Applet"
- "Creating a New View" on page 185
- "Adding the View to a Screen" on page 186

Creating a New List Applet

To display the data from the NREC Renovations VBC you must create a new applet. The NREC design specifies a standard list applet. You will use the List Applet New Object Wizard to create it. In the next section, you will create a master-detail view to contain the applet.

To create a new list applet to display renovation data

1 Choose File > New Object.

The New Object dialog box appears.

2 Click the Applets tab and then select List Applet.

3 Complete the wizard entering the following information as you go.

Field	Value
Project	NREC Configuration
Business Components	NREC Renovations VBC
Applet Name	Renovations List Applet
Display Title	Renovations
Web Templates	Base: Applet List (Base/List Edit) Edit: Applet List Edit (Edit/New/Query) Edit List: Applet List Edit (Edit/New/Query)
Fields	House_ID, Date, Description
Controls	Accept default controls selected for the applet.

Creating a New View

To display the Renovations list applet you need to add it to a view. NREC requires a standard master-detail view, showing the House Form applet on top and the Renovations list applet on the bottom. This will allow partner agents to see all the renovation records for each house record.

To create a new view

1. Choose File > New Object.
2. In the New Objects Wizard dialog box, select View.

   The New View Wizard opens.
3. Complete the wizard providing the necessary information as you go.

Field	Value
Project	NREC Configuration
Name	House Details - Renovations View
Business Object	Internal Product
Title	Renovations
Web Template	View Detail (Parent w/ Pointer)
Applets	Product Form Applet Renovations List Applet

Adding the View to a Screen

The next step is to add the new view to a screen. For the current example, you will add the House Details - Renovations View to the NREC House screen.

See "Creating the Houses Screen" on page 128 for information about the Houses Screen.

To add a view to a screen

1. Expand the Screen Object type.
2. Select the NREC Houses Screen.
3. Select the Screen View object type (child of Screen).
4. In the Screen View window, add a new record using the following properties:

Field	Value
View	House Details - Renovations View
Viewbar Text	Renovations
Sequence	30

Compiling and Testing

After you have completed your configuration work, perform the following tasks:

- Compile your changes. Be sure to compile changes for all projects that contain objects you modified. For example, the Internal Product business object belongs to the Product project by default.
- Register the House Details - Renovations View in the application.
- Add House Renovations to a responsibility that you can test.
- Test your work.

  Review Figure 25 on page 173 to see what the results should look like.

Code Sample

The following is the code breakdown for the NREC's example business service. See "Defining Business Service Scripts" on page 176 for the steps for defining the code and associating it with the business service.

```
(declaration) :-

  Dim FieldCount As Integer
  Dim NoOfRecords As Integer
  Dim DataSource As String
  Dim ColumnsReqd As String
  Dim VBCName as String

Pal_Init : -

  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

  ' Initialise 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), ""
        'MsgBox(theFieldName(Count))
        Next Count
     Close #1

  End Sub

Pal_Insert :-

  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
```

```
'
******************************************************************
****************

' * 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

  'MsgBox OutputStr

'
******************************************************************
****************

' * Create the Output String In the DataSource File

'
******************************************************************
****************
```

```
    Open DataSource For Append As #1

     Print #1,OutputStr

    Close #1

  End Sub

Pal_PreInsert :-

  Sub Pal_PreInsert (Outputs As PropertySet)

'
*******************************************************************
***************

' * A Pre Insert Method is Required before any insert is applied,
Its use to provide

' * Default Values, This Service does not provide any defaults, But
must return a

' * populated property set, so this points to the first field by
default

'
*******************************************************************
***************

    dim newRow As PropertySet
    set newRow = TheApplication.NewPropertySet()

    newRow.SetProperty Left(ColumnsReqd,Instr(ColumnsReqd,",")-
1),""

    Outputs.AddChild newRow

  End Sub


Pal_Query :-

  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 persistant store 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)
    'MsgBox 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)
  '        MsgBox 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 Persistant 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
```

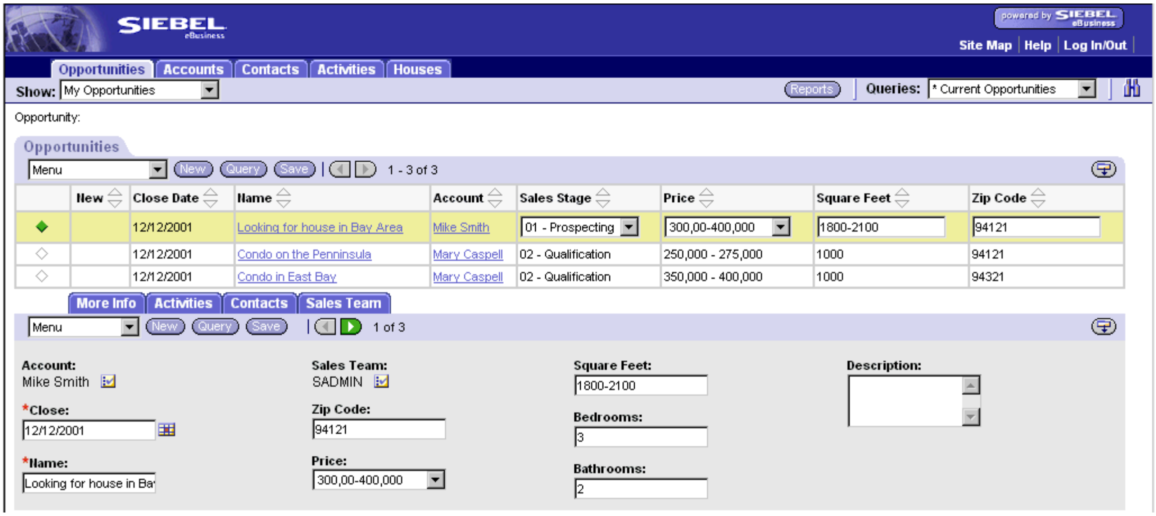
```
'
*****************************************************************
***************

  If DoQuery = 1 Then

   Open DataSource For Input As #1
    For Count = 1 To FieldCount
      Input #1,theKey
      If Instr(ColumnsReqd,theKey) <> "0" Then
        theFieldName(Count) = RTrim(theKey)
        OutColCount = OutColCount + 1
      Else
        theFieldName(Count) = "CVNR"
      End If
    Next Count

  me.SetProperty VBCName & "ColCnt", CStr(OutColCount)

'
*****************************************************************
***************

' * Retreveal 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 Persistant 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
     Row(i).SetProperty ColList, ColData
    Outputs.AddChild Row(i)
   Next i
  End If

End Sub



Service_PreInvokedMethod :-

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 = CancelOperation
  ElseIf MethodName = "Insert" Then
    Pal_Insert Inputs, Outputs
    Service_PreInvokeMethod = CancelOperation
  ElseIf MethodName = "Delete" Then
    'Pal_Delete Inputs              - Not Yet Completed
    Service_PreInvokeMethod = CancelOperation
  Else
    Service_PreInvokeMethod = ContinueOperation
  End If

End Function
```

Modifying the Look and Feel of the Web Client 9

This chapter follows the NREC example to highlight key tasks and information about configuring the look and feel of the user interface. These tasks will lead you through the process of modifying the principal user interface elements based on NREC's requirements. Tasks include modifying graphics, Web templates, and cascading style sheets.

NREC's requirement is to modify the look and feel of the user interface to match their corporate Web site. This includes modifying the shape of tabs and buttons and changing the color scheme. See Figure 27 and Figure 28 on page 200 for the before and after look of the user interface.



Show: My Opportunities

Queries: * Current Opportunities

Opportunity:

Opportunities

1 - 3 of 3

	New	Close Date	Name	Account	Sales Stage	Price	Square Feet	Zip Code
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting	300,00-400,000	1800-2100	94121
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification	250,000 - 275,000	1000	94121
◇		12/12/2001	Condo in East Bay	Mary Caspell	02 - Qualification	350,000 - 400,000	1000	94321

1 of 3

Account: Mike Smith

*Close: 12/12/2001

*Name: Looking for house in Ba

Sales Team: SADMIN

Zip Code: 94121

Price: 300,00-400,000

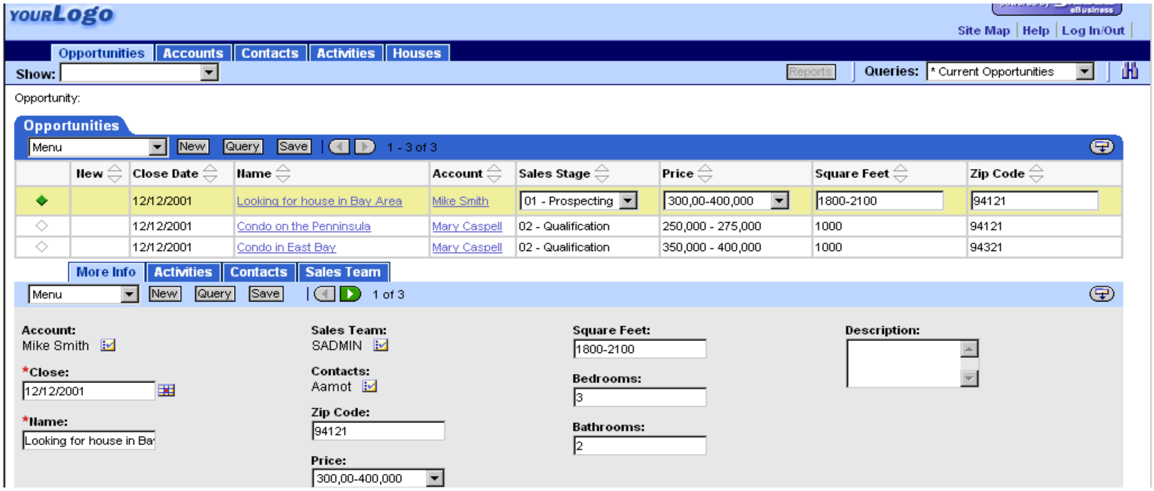
Square Feet: 1800-2100

Bedrooms: 3

Bathrooms: 2

Description:

Figure 27. Standard Look and Feel



	New	Close Date	Name	Account	Sales Stage	Price	Square Feet	Zip Code
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting	300,00-400,000	1800-2100	94121
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification	250,000 - 275,000	1000	94121
◇		12/12/2001	Condo in East Bay	Mary Caspell	02 - Qualification	350,000 - 400,000	1000	94321

Account: Mike Smith

*Close: 12/12/2001

*Name: Looking for house in Ba

Sales Team: SADMIN

Contacts: Aamot

Zip Code: 94121

Price: 300,00-400,000

Square Feet: 1800-2100

Bedrooms: 3

Bathrooms: 2

Description:

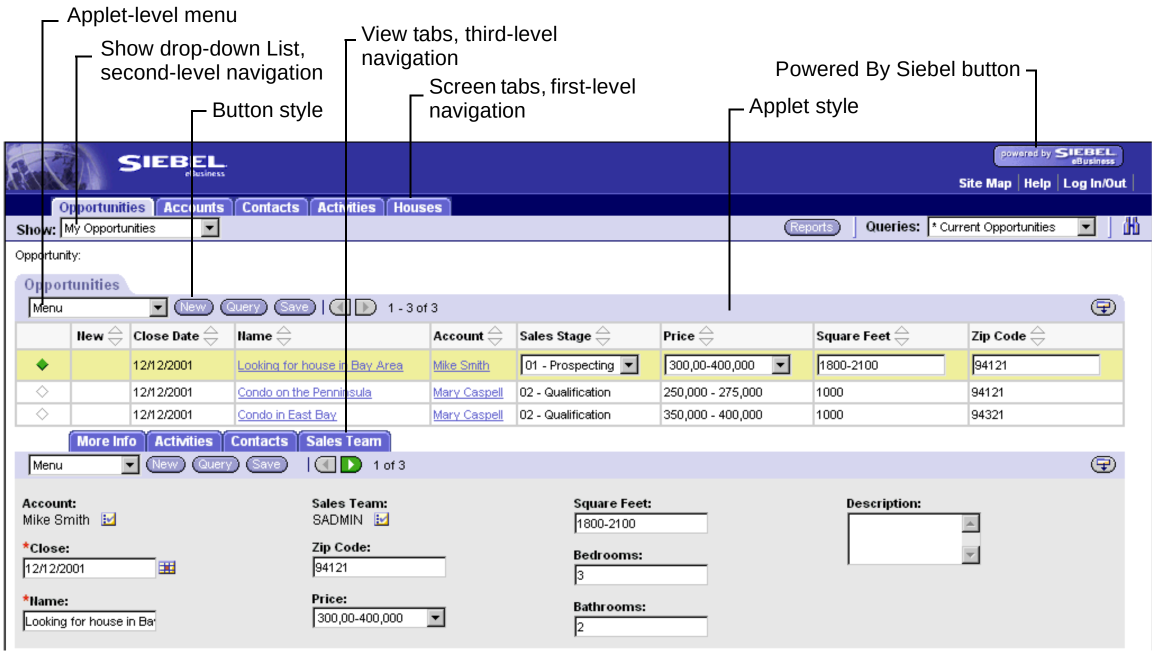
Figure 28. NREC's Modified Look and Feel

The tasks that are required to make the changes are covered in the rest of this chapter.

- "Adding a New Logo to the Banner" on page 206
- "Modifying Tabs" on page 210
- "Modifying Button Styles" on page 214
- "Modifying the Screen Bar and View Bar Colors" on page 216
- "Modifying Applet Colors" on page 218

User Interface Elements

The user interface elements discussed in the this chapter are shown in Figure 29 and Figure 30 on page 202. For reference information about Web templates and other user inference topics, see *Siebel Tools Reference* and *Siebel Tools Online Help*



Show: My Opportunities

Queries: * Current Opportunities

Opportunity:

1 - 3 of 3

	New	Close Date	Name	Account	Sales Stage	Price	Square Feet	Zip Code
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01 - Prospecting	300,00-400,000	1800-2100	94121
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02 - Qualification	250,000 - 275,000	1000	94121
◇		12/12/2001	Condo in East Bay	Mary Caspell	02 - Qualification	350,000 - 400,000	1000	94321

1 of 3

Account: Mike Smith

*Close: 12/12/2001

*Name: Looking for house in Ba

Sales Team: SADMIN

Zip Code: 94121

Price: 300,00-400,000

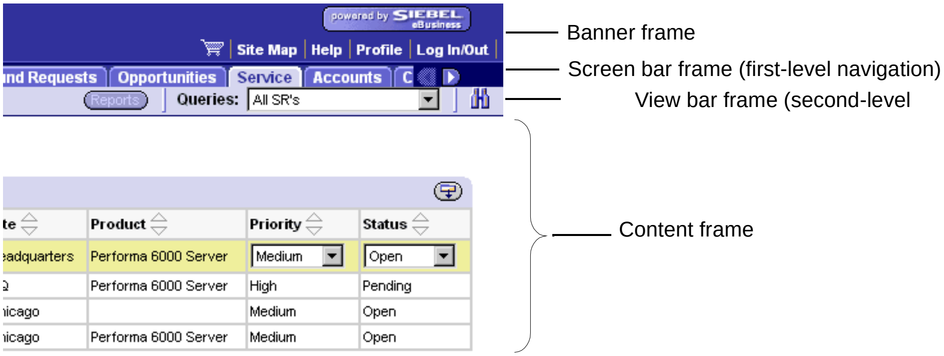
Square Feet: 1800-2100

Bedrooms: 3

Bathrooms: 2

Description:

Figure 29. UI Elements



Queries: All SR's

te	Product	Priority	Status
eadquarters	Performa 6000 Server	Medium	Open
Q	Performa 6000 Server	High	Pending
icago		Medium	Open
icago	Performa 6000 Server	Medium	Open

Banner frame

Screen bar frame (first-level navigation)

View bar frame (second-level

Content frame

Figure 30. Frames

Source Control for Web Template and Related Files

Siebel Web templates and supporting files such as graphics and cascading style sheets are not stored in the Siebel repository. They are stored under the root directory of your Siebel installation. Because these files are outside a Siebel repository they are not controlled by Siebel Tools Check In and Check Out Process. If multiple developers are modifying files outside the repository, it is recommended that you use a third-party source control application to make sure developers do not overwrite each other's work.

See "Location of Siebel Web Templates and Related Files" on page 203 for the locations of Web templates and supporting files.

Location of Siebel Web Templates and Related Files

Siebel Web templates and supporting files are installed with the Siebel Server, Siebel Tools, and the Mobile and Dedicated Web Clients. The locations of the files are listed below.

NOTE: In the directory paths listed below, *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.** The files in the Siebel Tools installation directory are used when objects such as views and applets are rendered using the Preview option in the Edit Web Layout window. The locations of the files in the Siebel Tools installation directory are:

Files	Directory Path
Web templates	*tools_root*\WEBTEMPL
Image files	*tools_root*\PUBLIC\language_code\IMAGES
Style Sheets	*tools_root*\PUBLIC\language_code\FILES

- **Siebel Server**. The files in the Siebel Server installation directory are used by the Siebel Web engine when rendering the user interface in the Web Client. The locations of the files for a Siebel Server installation are:

Files	Directory Path
Web templates	*siebsrvr_root*\WEBTEMPL
Image files	*siebsrvr_root*\webmaster\images\language_code
Style Sheets	*siebsrvr_root*\webmaster\files\language_code

- **Siebel Mobile or Dedicated Web Client.** The files in the Mobile or Dedicated Web Client installation directories are used when rendering the user interface for these clients. The locations of these files are:

Files	Directory Path
Web templates	`client_root\WEBTEMPL`
Image files	`client_root\PUBLIC\language_code\IMAGES`
Style Sheets	`client_root\PUBLIC\language_code\FILES`

The Web Template Development Process

When modifying Siebel Web templates you typically work with the files stored on your local machine. After you have tested 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 depending on whether the changes are simple or complex.

- **Simple Web template modifications**. 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 syntax errors that may occur. Additionally, the Preview functionality allows you to view the Web template and the objects that are mapped to it as they would be rendered in the user interface. For minor changes, it is easier to debug the files this way, rather than having to run the Mobile Web client and navigate to the applet where the changes would appear. After the change is verified using Siebel Tools, developers can manually copy the changed template into the Web template directory of their Siebel Mobile Web Client, launch the application, and do final testing.

NOTE: The modifications presented in this chapter are relatively simple. The tasks are written assuming that you are using this development process.

Complex Web template development. For complex template development, such as creating new Web templates or significantly modifying the layout of existing templates, the process should include creating a prototype in HTML, translating the various parts of the template into SWE syntax, and then debugging as described in the previous paragraph.

For more information about SWE tags and syntax, see *Siebel Tools Online Help*.

Adding a New Logo to the Banner

The banner area is implemented as the top frame in the application frame set. It contains the application banner and the Powered by Siebel Logo, as shown in Figure 31.

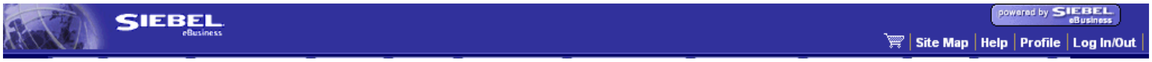


Figure 31. Banner Frame

The banner frame has its own Siebel Web template file that you can modify. For example, you add your company's logo to the left side of the banner, as shown in Figure 32.

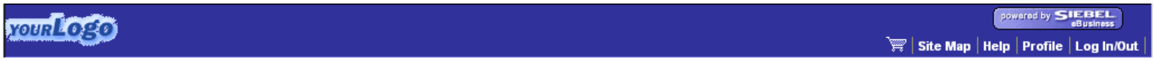


Figure 32. Banner Frame with Your Logo

In this example, assume that the NREC requirement is to add their corporate logo to the banner frame and change the color scheme of the frame to match their corporate Web site. To do this, you need to complete the following two tasks:

- "Modifying the Banner Frame Web Template" on page 207
- "Modifying the Banner Color Scheme" on page 208

Modifying the Banner Frame Web Template

The first step for adding a new logo to the Frame Banner is to add the image file to the Images directory and then modify the Web template to include the new image.

To add a logo to the banner frame

1 Add the image file you will use as your logo to the images directory of your Siebel Tools installation and to the Images directory of your test application.

For example,

- `tools_root\PUBLIC\language_code\IMAGES`
- `client_root\PUBLIC\language_code\IMAGES`

2 In Siebel Tools choose View > Windows > Web Templates Window.

The Web Template Explorer appears.

3 In the Web Template Explorer find dCCFrameBanner.

The content of the template is displayed.

dCCFrameBanner is the banner frame template for customer and partner applications.

4 Right-click in the window that displays the HTML, and then choose Edit Template.

Your default HTML editor opens the dCCFrameBanner.swt file.

NOTE: If you have not set your default HTML Editor, choose View > Options, select the Web Template Editor tab, and then define the path to the editor you want to use.

5 Find the following lines of HTML:

```
<td><img src="images/banner_globe.jpg" border="0"></td>
<td><img src="images/SiebelLogo.gif" border="0"></td>
```

6 Delete the first < td > tag and in the second < td > tag, replace the Siebel Logo image with your logo image.

   Changes appear below:

```
<td><img src="images/your_logo.gif" border="0"></td>
```

   Deleting the first < td > tag removes the image of the globe from the banner. Adding your logo image to the second < td > tag replaces the Siebel logo with your logo.

7 Save the file.

8 Copy the file dCCFrameBanner.swt from the Web Templates directory in your Tools installation directory to the Web Templates directory in your client installation.

   NOTE: Be sure to save a backup copy of the original dCCFrameBanner.swt file.

9 Clear your browser's cache directory and then open the Siebel Customer or Partner application that you want to test.

   For the current example, open the Partner Portal application.

   Changes appear similar to Figure 32 on page 206.

Modifying the Banner Color Scheme

The color schemes and other styles in Siebel Web templates are controlled using a cascading style sheet. To change the color scheme of the banner frame, you must modify the rules defined in the cascading style sheet for your application.

In this example you will work with the banner selector. The banner selector defines display rules for the banner frame. To modify the background color, you will change the value of the Background-color property. For a list of selectors used in cascading style sheets, see *Siebel Tools Online Help*.

To modify the banner background color

1. Navigate to the *client_root*\public\language_code\files directory and then open the file dCCmain.css.

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

   NOTE: The cascading style sheet for Employee applications is main.css.

2. Find the following selector and properties in the Banner Definitions section of the file.

```
.banner,
TD.banner TD A,
TD.banner TD A:visited,
TD.banner TD A:hover { background-color: #333399; color: #ffffff;
text-decoration:none; font-weight:bold;}
```

3. Change the value of the background-color property and the color property as shown below:

```
.banner,
TD.banner TD A,
TD.banner TD A:visited,
TD.banner TD A:hover { background-color: #BDD3FF; color: #333399;
text-decoration:none; font-weight:bold;}
```

4. Save the file and then open your application.

   The banner frame appears with the new background color, as shown in Figure 33.

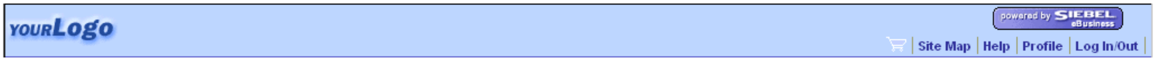


Figure 33. Banner Frame with Modified Color Scheme

Modifying Tabs

Tabs in your Siebel application are used for first-level and third-level navigation. They are broken up into separate elements and then layered on top of each other to create the final look and feel. The text is separated from the graphics, providing greater flexibility when configuring applications. The elements that make up tabs are:

- **Tab background.** Constructed with images.
- **Tab shape**. Constructed with images.
- **Text.** Stored in the Siebel Repository.
- **Text style.** Defined in cascading style sheets.

You can modify tabs to suit your needs. For example, to match tab styles to other pages on their corporate web site, NREC has chosen to create tabs with squared corners and to change the color scheme. The tasks for accomplishing this are:

- "Modifying Tab Images"
- "Modifying Link Styles" on page 212

Modifying Tab Images

Tab background and shape are constructed from three separate images, as shown in Figure 34. NREC's requirement is to modify the default images so that the corners are squared instead of rounded, as shown in Figure 35. This task is typically done by a graphic designer.

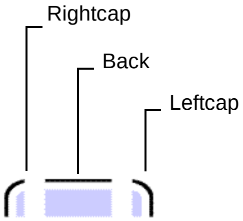


Figure 34. Graphic Elements of Tabs

Tabs support two states: on and off. You must design graphics for both states. There are two sets of tab images to modify. One set of images is for the first-level navigation tabs across the top of a page. The other set of images is for the third-level navigation tabs that appear across the middle of a page. The *on state* of tabs is controlled by one set of images for employee applications and another set of images for customer and partner applications. Because NREC is modifying the Partner Portal application, the images to modify for the *on state* are the ones that begin with "dcctb..." See Table 10.

All of the tab images are located in `root_directory\PUBLIC\`*`language_code`*`\IMAGES`. The *`root_directory`* is the installation directory of your application and *`language_code`* is the three letter code for the language of the installed software.

Table 10. Tab Images to Modify

Navigation	On State (Employee Apps)	On State (Customer and Partner Apps)	Off State (Employee, Customer, and Partner Apps)
First-level navigation tabs	tb1_mid_1.gif	dcctb1_mid_1.gif	tb1_mid_0.gif
	tb1_opn_1_d.gif	dcctb1_opn1_d.gif	tb1_opn_0_d.gif
	tb1_cls_1_d.gif	dcctb1_cls_1_d.gif	tb1_cls_0_d.gif
Second-level navigation tabs	tb2_mid_1.gif	dcctb2_mid_1.gif	tb2_mid_0.gif
	tb2_opn_1_d.gif	dcctb2_opn_1_d.gif	tb2_opn_0_d.gif
	tb2_cls_1_d.gif	dcctb2_cls_1_d.gif	tb2_cls_0_d.gif

NOTE: The _d suffix signifies that an image encodes directionality. If you plan to deploy an application in any of the right-to-left (RTL) languages, such as Hebrew, you will also need to create mirrored copies of these images and place them in the RTL language directories.

To modify default tabs

- Use a graphics software package to modify the images defined in Table 10.

For example, assume that a graphic designer for NREC modified the images that make up tabs, as shown in Figure 36.

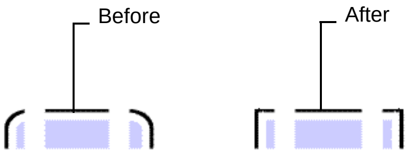


Figure 35. Modified Tab Images

Modifying Link Styles

The text that appears on a tab is actually a link. When a user clicks a tab, the link takes them to another page. To modify the style of the link, you change the appropriate style rules defined in the cascading style sheet. In the NREC example, the requirement is for the style of the link to match the color changes implemented on the banner. Because there are two levels of tabs (first-level navigation and third-level navigation) the properties for two selectors need to be modified. The selectors are:

- **Tier1On.** Defines how the links on the first-level tabs look and behave.
- **Tier3On.** Defines how the links on the third-level tabs look and behave.

To modify link style

1. Navigate to the `*client_root*\PUBLIC\language_code\FILES` directory and then open the file dCCmain.css.

   The file, `dCCmain.css`, is the cascading style sheet for Siebel Employee Applications.

2. Find the following selectors.

   - `Tier1On`
   - `Tier3On`

3 Modify the color and background properties for each selector to match the attributed defined for the banner frame.

The changes are shown below:

```
...

TR.tier1On, TD.tier1On,
TD.tier1on A,
TD.tier1on A:visited,
TD.tier1on A:hover,
{  font-size: 9pt;
   color:#003399;
   background-color: #BDD3FF;
   font-weight:bold;
   text-decoration:none;
}

...

TR.tier3On, TD.tier3On,
TD.tier3on A,
TD.tier3on A:visited,
TD.tier3on A:hover
{  font-size: 9pt;
   color:#003399;
   background-color: #BDD3FF;
   font-weight:bold;
   text-decoration:none;
}
```

4 Save the file.

When you refresh the current page of your application, the changes to the text in the active tabs take effect.

Modifying Button Styles

Most buttons are similar to tabs in that the graphic elements are independent of the text element. Graphics are stored as image files and text is stored in the Siebel repository. This allows you to modify text without having to modify the button, or modify the button without modifying the text.

NREC's corporate Web site contains square buttons rather than buttons with rounded sides. NREC wants to modify the default style of the buttons so that the buttons are consistent with the rest of their Web site. This type of modification is typically done by a graphic designer. Figure 36 shows what the buttons look like before and after the design work.

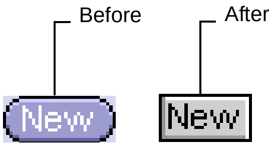


Figure 36. Button Images

To modify button images

1. Navigate to the directory that contains button images.

   `root_directory`\PUBLIC\`language_code`\IMAGES\BTTNS

2. Change the shape of standard buttons by modifying the following files:

   - `dccbtn_opn_d.gif`
   - dccbtn_mid.gif
   - `dccbtn_cls_d.gif`

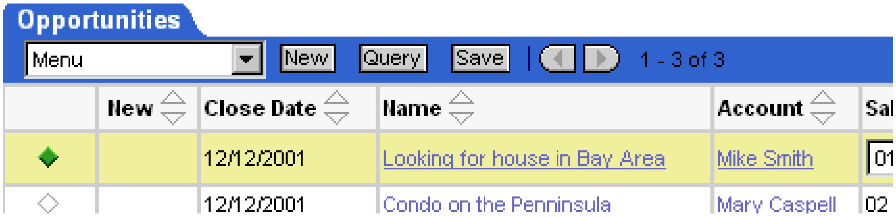
   NOTE: The shape of standard buttons for employee applications are `btn_opn_d.gif` and `btn_cls_d.gif`.

3 Modify the following selectors in the Minibutton section of the cascading style sheet (dCCmain.css) as shown below:

```
.minibutton {}
.minibuttonOn{font-size: 8pt; color: white; text-decoration:none;
}
.minibuttonOn A,
.minibuttonOn A:link,
.minibuttonOn A:visited,
.minibuttonOn A:hover
{  font-size: 8pt; color: #000000; text-decoration:none;}

.minibuttonOff,
.minibuttonOff A,
.minibuttonOff A:visited,
.minibuttonOff A:hover
{  font-size: 8pt; color: #999999;text-decoration:none;}
```

4 Save the files and move them to the appropriate directory on your test machine.

When you open your application, the modified buttons appear with black text as defined in the style sheet.



	New	Close Date	Name	Account	Sal
◆		12/12/2001	Looking for house in Bay Area	Mike Smith	01
◇		12/12/2001	Condo on the Penninsula	Mary Caspell	02

NOTE: There are some buttons that consist of a single image. These buttons have an icon affixed directly to the button face. These buttons do not use the leftcap and rightcap images. Instead, the shape is encoded into the button image and stored as a single image. An example of this type of button is the more button (more.gif).

Modifying the Screen Bar and View Bar Colors

The screenbar and viewbar are the frames that hold the screen tabs and the show drop-down list. See Figure 30 on page 202. These frames contain color elements, which are controlled using the cascading style sheet. Additionally, in the case of customer and partner applications, the background color of the screen bar frame is controlled using a graphic image.

The NREC requirement is to modify the background color for the screenbar and viewbar to match the background of the other user interface elements that were defined for tabs and the banner frame.

The image that controls the screenbar background is:

- **dccscrnbar_back.gif**. Controls the background color of the screen bar frame (first-level navigation tabs.) This image is a tiling background used to make sure that the backgrounds display well in both IE and NS browsers. It is only used for customer and partner applications. Open this image and modify color accordingly.

The selectors that control the viewbar background are:

- **Tier2On.** Controls the background color of the view bar frame (second-level navigation).

- **Tier2Rule.** Controls the background color of the view bar frame (second-level navigation).

To modify screenbar and viewbar style

1. Navigate to the `client_root\PUBLIC\language_code\FILES` directory and then open the file `dCCmain.css`.

   The file, `dCCmain.css`, is the cascading style sheet for Siebel Employee Applications.

2. Locate the tier2 selectors and then modify the value for the background color as shown below:

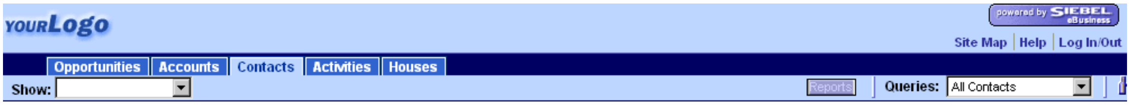
   ```
   .tier2Back {background-color:#BDD3FF;}
   ```

   ```
   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;}
   ```

3. Save the file and then open the application.

   The background color on the screenbar and viewbar reflect your changes.



Modifying Applet Colors

There are three major applet styles in your Siebel application: Parent, Child, and Grandchild. The applet styles define the color of the tab that appears at the top of the applet and the color of the control bar where applet-level menus and buttons appear. In the cascading style sheet, the selectors for each applet style are:

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

Each applet style is defined by sub-elements To modify the appearance of each applet style, you modify the definitions for the following sub-elements:

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

To modify the applet color scheme

1. Open `dCCmain.css`.
2. Locate the following selectors for parent style applets and modify the attributes as shown.

Selector	CSS Attribute	Modified Value
AppletStyle1.AppletButtons	Background-color	#3366CC
AppletStyle1.AppletBorder	Background-color	#3366CC
AppletStyle1.AppletTitle	Background-color	#3366CC
AppletStyle1.AppletTitle	Color	#FFFFFF

3. Locate the following child applet selectors and modify the attributes as shown in the table.

Selector	CSS Attribute	Modified Value
`AppletStyle3 AppletButtons`	`Background-color`	`#BDD3FF`
`AppletStyle3 AppletBorder`	`Background-color`	`#BDD3FF`
`AppletStyle3 AppletTitle`	`Background-color`	`#BDD3FF`

4 Locate the following grandchild applet selectors and modify the values for the subelements as shown.

Selector	CSS Attribute	Modified Value
`AppletStyle7 AppletButtons`	`Background-color`	`#BDD3FF`
`AppletStyle7 AppletBorder`	`Background-color`	`#BDD3FF`
`AppletStyle7 AppletTitle`	`Background-color`	`#BDD3FF`
`AppletStyle7 AppletTitle`	`Color`	`#003399`

5 Save your changes and move the file to your test machine.

The background color scheme on parent and child applets reflects the changes.

Figure 28 on page 200 shows the modified applet color scheme.

Migrating Repository Data to the Test Environment 10

This chapter tells you how to migrate your configuration from the development environment to the test environment. The data to be migrated includes:

- Repository data, such as new or modified applets, views, and business components.
- Database schema changes, such as new or modified columns and tables.
- Modified Web templates and cascading style sheets.

Repository data and database schema changes is migrated using the Repository Migration Utility. Web templates and cascading style sheets are migrated by manually copying the files from the development server to the test server.

Migrating Repository from Development to Test

This section explains how you migrate repository data from a source environment to a target environment. Following the NREC example, the source is the development environment and the target is the test environment.

You use the Repository Migration Utility to migrate data when there has been a database schema change in the source environment, such as adding columns to a base table or creating a new table (see Chapter 6, “Configuring the House and Opportunity Entities”). The Repository Migration Utility exports the repository data from the source environment, updates the physical database schema of the target database, and then imports the object definitions. The result is that the database schema and object definitions of target environment matches source environment.

NOTE: If you only need to migrate user interface and business object definitions, you can use the Export and Import Repository Utilities.

For detailed information about managing repositories, see *Siebel Tools Reference* for more information.

Preparing for the Migration

Migration occurs between one server and another. Therefore, make sure all developers check in their projects from their local databases back to the server before the migration. After migration begins, developers should not make any more changes—all such changes will be lost.

Before migrating the repository you also need to stop all server tasks and disconnect database access until the migration is complete.

NOTE: If testing in the Test environment reveals problems, these problems can be corrected either in the Development environment (and the changes migrated again) or in the Test environment.

Migrating the Repository

The Repository Migration Utility is a wizard that prompts you for the required information as you proceed through a succession of dialog boxes. You launch the Repository Migration Utility from the Siebel Database Configuration Utility.

To migrate a repository

1. Open the Siebel Database Configuration Utility using one of the following methods:

   - From the Windows Start menu, choose Programs > Siebel Servers 7.0 > Configure DB Server.
   - From a command prompt window, navigate to siebsrvr/bin directory and then type the following command:

     ```
     ssincfgw -l ENU -v Y -f SiebSrvr_Root\admin\dbsrvr.scm
     ```

     (where *SiebSrvr_Root* is the root installation directory for your Siebel Server)

   The Siebel Server Configuration Utility appears.

2. Complete the information requested in the following the dialog boxes:

In This Dialog Box...	Enter or Select...
Siebel Enterprise Parameters: Gateway Server Address	Gateway Server Address and Enterprise Server Name.
Installation and Configuration Parameters: Siebel Server Directory	Siebel Server directory
Installation and Configuration Parameters: Siebel Database Server Directory	Database server directory

3. In the Siebel Database Server Options: Siebel Database Operations dialog box, select Migrate Repository from the list of operations and then click Next.

4 Complete the required information as you proceed through the remaining dialog boxes. These dialog boxes will gather information about the source and target environments.

   See the Repositories chapter in *Siebel Tools Reference* for a comprehensive list of dialog boxes and descriptions.

5 After completing the requested information, review the results in the Finish dialog box and then click Finish.

   The Migrate Repository Wizard migrates the repository data from the source database to the target database and updates the physical schema of the target database.

Moving Modified Web Templates and Related Files

Every Siebel Server installation includes a set of Web templates (.swt 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, you must copy the modified files to the target environment. For example, NREC would need to migrate the files modified as described in this chapter from the development environment to the test environment.

To move Web templates and related files from Development to Test

- Copy any new or modified files of the following types from the development server to the test server:

Files	File Location
Web Templates	`siebsrvr_root\WEBTEMPL`
Images	`siebsrvr_root\webmaster\images\language_code`
Cascading Style Sheets	`siebsrvr_root\webmaster\images\language_code`

Using EIM to Load Data into the Test Environment 11

You have already completed many of the tasks to prepare your Test environment. For example, you have moved the application (including your configuration changes) from Development to Test and have performed the setup tasks described in Chapter 12, "Required Application Administration Tasks."

To do further development and testing work, however, you need to have actual corporate data in your Siebel database. Up until now, you have been working only with the sample data available with your Siebel installation.

This chapter tells you how to load data from your legacy system into a Siebel database, using the Siebel Enterprise Integration Manager (EIM).

Generally, EIM is used 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 current goal is to load data into a Siebel database, this chapter deals only with bulk imports. For detailed information about EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

This chapter focuses on one specific import example: loading account data for the NREC. The section "Import Example" on page 229, provides more detailed information about the specific import example presented in this chapter.

Basic EIM Concepts

To begin working with EIM, you need to understand some fundamental concepts, which are described in this section.

Interface Tables

As illustrated in Figure 37, *interface tables* are intermediate database tables between the Siebel database and another database.

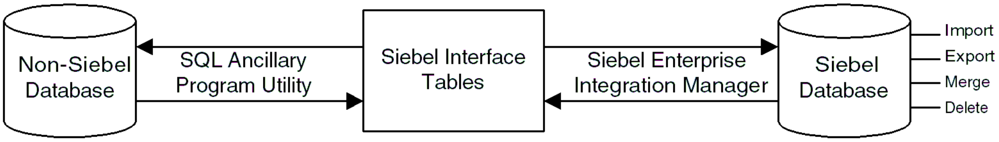


Figure 37. Process Flows Between Siebel Database and Other Databases

Rather than loading legacy data directly into Siebel base tables, you or your database administrator first load it into Siebel interface tables using your database-software-specific utility. Then, you use EIM to load data from the interface tables into the Siebel base tables.

Use the *Interface Tables Reference* to help determine how to map your own tables to the Siebel interface tables.

Process Overview

For any EIM process (whether it be import, export, merge, or delete), you need to complete the following steps:

1. **Load the interface tables.** For more information, see "Using the Interface Tables" on page 230.

2. **Edit the EIM configuration file.** For more information, see "Editing the Configuration File" on page 233.

3. **Run EIM.** For more information, see "Running EIM" on page 238.

4. **Check results.** For more information, see "Checking the Results of the Data Import" on page 241.

This chapter follows these steps to import data.

Import Example

This chapter focuses on how NREC used EIM to import its account data into the Siebel database. To do this, the NREC system administrator needed to load the applicable interface table (EIM_ACCOUNT), prepare the NRECimp.ifb configuration file, and run the EIM import process. This chapter provides the information for this example in the following order:

- Account data that will be loaded into the EIM_ACCOUNT interface table

  See "Using the Interface Tables" on page 230 for information about how to load an interface table, including finding out which columns are mandatory.

- Configuration file

  See "A Sample Configuration File" on page 234 to view the contents of the .ifb file that you will use to customize the behavior of the EIM process. There is also a brief explanation of the parameters that appear in the configuration file.

- EIM process

  See "To start EIM using the GUI" on page 238 for step-by-step instructions on running the EIM process to pull the accounts data from the EIM_ACCOUNT interface table into the specified base tables in the EBUSINESS database.

Using the Interface Tables

See *Interface Tables Reference* for information on using these tables, including:

- 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

Determining Which Columns Are Required

To some extent, you can decide which columns from your source table to load into an interface table. However, for some columns interface tables are required: they cannot be left NULL.

You can find out which columns are required either through Siebel Tools or by referring to *Interface Tables Reference*.

To use Siebel Tools to determine which columns are required

1. In the Object Explorer, expand the EIM Interface Table.

   The EIM Interface Tables window appears in the Object List Editor.

   NOTE: If the EIM Interface Tables object type is not visible, choose View > Options, and then under the Object Explorer tab, move the EIM Interface Tables object type from the Available to the Visible column.

2. In the EIM Tables window, find the interface table you plan to load.

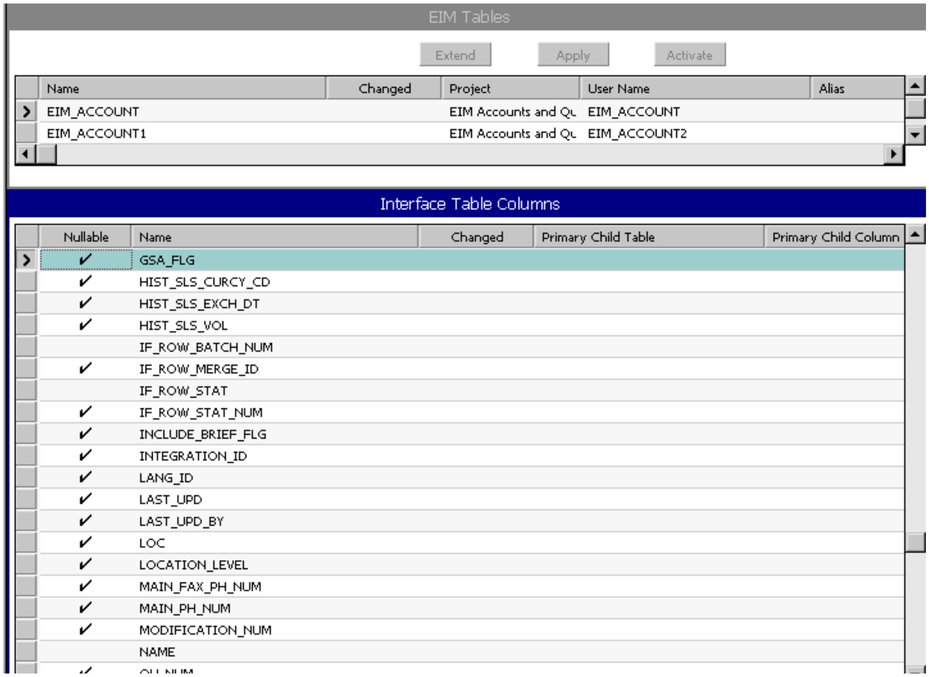
   For the current example, find EIM_ACCOUNT.

3. In the Object Explorer, select EIM Interface Table Column (a child object of EIM Interface Table).

   The EIM Interface Table Columns window appears in which the columns of the current table are displayed.

4 In the Interface Table Columns window, look at which columns have the Nullable property checked.

The columns marked Nullable are *not* required. For example, as shown in the following figure, the IF_ROW_BATCH_NUM, IF_ROW_STAT, and NAME columns are required because they are not marked Nullable.



EIM Tables

Name	Changed	Project	User Name	Alias
EIM_ACCOUNT		EIM Accounts and Qu	EIM_ACCOUNT	
EIM_ACCOUNT1		EIM Accounts and Qu	EIM_ACCOUNT2	

Interface Table Columns

Nullable	Name	Changed	Primary Child Table	Primary Child Column
✓	GSA_FLG			
✓	HIST_SLS_CURCY_CD			
✓	HIST_SLS_EXCH_DT			
✓	HIST_SLS_VOL			
	IF_ROW_BATCH_NUM			
✓	IF_ROW_MERGE_ID			
	IF_ROW_STAT			
✓	IF_ROW_STAT_NUM			
✓	INCLUDE_BRIEF_FLG			
✓	INTEGRATION_ID			
✓	LANG_ID			
✓	LAST_UPD			
✓	LAST_UPD_BY			
✓	LOC			
✓	LOCATION_LEVEL			
✓	MAIN_FAX_PH_NUM			
✓	MAIN_PH_NUM			
✓	MODIFICATION_NUM			
	NAME			
✓	OU_NUM			

NOTE: For your convenience, right-click and choose Columns Displayed to change the column order on this view so that the Nullable property appears next to the Name property, as shown in the preceding figure.

To use the Interface Tables Reference to determine which columns are required

1. In the *Interface Tables Reference*, find the table you plan to load.
2. Look at the value of the Req column.

   If the value is Y, the column is required.

Loading the Interface Tables

To load data into an interface table, use your database-software-specific load utility, such as one of the load utilities available for DB2. For information, see the documentation for your database software.

Make sure you load all the required columns.

Some columns in the interface tables are part of a *user key sequence*. The user key sequence is the set of columns (called user key columns) whose values uniquely identify a row. As EIM loads each row, it checks whether the interface table already contains that row's user key sequence, thus preventing the import of duplicate rows.

To find out which columns of an interface table are part of the user key sequence, check the UK column in the *Interface Tables Reference*.

Editing the Configuration File

EIM reads a special configuration file that specifies the EIM process to perform and the appropriate parameters. This example features the import process.

The EIM configuration file is an ASCII text file of extension type .ifb that resides in the admin subdirectory under the Siebel Server directory. Before initiating an EIM process, you must edit the contents of the EIM configuration file to define the process to be performed.

NOTE: In a Unicode environment, the EIM configuration file must be saved as a Unicode text file. For more information, see *Siebel Enterprise Integration Manager Administration Guide*

In this example, EIM sets the process locale as specified in the configuration file at start-up. The default configuration file, default.ifb, provides the data used in the process if no other configuration file is specified. This example uses the configuration file NRECimp.ifb.

The configuration file begins with a header section used to specify global parameters that apply to all process sections defined later in the file. Following the header section, there must be at least one process section with its associated parameters. Some process section parameters are generic to all EIM processes. Other process section parameters are specific to a particular EIM process. The following parameter sections describe the required header section and generic process section parameters.

See the *Siebel Enterprise Integration Manager Administration Guide* for:

- Other process and header parameters
- Alternative sources from which EIM accepts parameter values
- The process for defining multiple processes in the configuration file

A Sample Configuration File

The sample configuration file in this section shows the entries that are necessary only for the operations in the particular import EIM example for this chapter. See *Siebel Enterprise Integration Manager Administration Guide* for more information about the EIM configuration file.

The text of the sample configuration file, NRECimp.ifb, is shown here:

```
/* 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 Parameter Section

This section describes the header parameters that you will need to specify in the configuration file to properly configure EIM for an import process. See "A Sample Configuration File" on page 234 for the list of the values assigned to each of the header parameters. Table 11 shows only those header parameters featured in the NRECimp.ifb.

Table 11. Required Header Section Parameters

Command	Description
USERNAME	Database/employee login (SADMIN for NREC)
PASSWORD	Database password (SADMIN for NREC)
PROCESS	Initial/main process section to run (Import Accounts for NREC)
[Siebel Interface Manager]	Header section must use this reserved name

Process Parameter Section

A parameter in the process section only applies to the particular process specified and overrides any corresponding value in the header section for the specific process. There are both generic and specific process parameters.

Table 12 describes the parameters that NREC used in the process section of NREC's configuration file, NRECimp.ifb. The process section is generic to all EIM processes. For parameters specific to the import process, see the *Siebel Enterprise Integration Manager Administration Guide*.

Table 12. Process Section Parameters Generic to All EIM Processes

Command	Description
ONLY BASE COLUMNS	Only process these 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 for NREC.
ONLY BASE TABLES	Only process these tables. S_ORG_EXT, S_ADDR_ORG, S_PARTY for NREC.
TABLE	Interface (IF) table for process. EIM_ACCOUNT for NREC.
TYPE	IMPORT,EXPORT,DELETE,MERGE,SHELL. IMPORT for NREC's import example.

Disabling Logging Before Initial Loads

Before performing an initial load through EIM, it is a good idea to disable transaction logging to improve EIM performance. Disabling transaction logging consists of disabling the Docking: Transaction Logging system preference.

Caution: Do not disable Docking Transaction Logging if you have active mobile clients. Otherwise, the server database and mobile client databases will not be synchronized after the import. To synchronize them again, you would need to again extract the database for each mobile client, and run database initialization on each client. See Chapter 16, “Implementing Siebel Remote,” and *Siebel Remote and Replication Manager Administration Guide* for more information on synchronization.

To disable transaction logging

1. Navigate to the System Preferences screen.
2. Select Docking: Transaction Logging.
3. In the System Preference Value field, type FALSE.

   Do not change this value to TRUE until after you import all the initial data.
4. Click outside the row to save changes.

You can also change the transaction logging preference by changing the LOG TRANSACTIONS parameter in the EIM configuration file.

Running EIM

This section describes how to start an import process from the Siebel Client. For information about starting the import process from the command line, see the *Siebel Enterprise Integration Manager Administration Guide*.

To start EIM using the GUI

1. Navigate to the Enterprise Operations screen.
2. Click the Component Requests view tab.
3. In the Component Requests form, click the menu button and then New Record.
4. In the Component/Job field, click the select button.

   The Component/Jobs dialog box appears.

5. In the Component/Jobs dialog box, select the Enterprise Integration Mgr component and click OK.

If you want to use a component job based on EIM for your component request, you must first define the component job. For information on defining component jobs, see *Siebel Server Administration Guide*.

1. Complete the rest of the fields and click Save.
2. In the Component Request Parameters list, add or change any component parameters for the EIM process and click Save.

   Figure 38 on page 240 shows an example of setting parameters in the Component Request Parameters list.

   a. In the Component Request Parameters list, click the menu button and then New Record.
   b. In the Name field, click the select button.

      The Job Parameters dialog box appears.

   c. To select the EIM configuration file to use, select Configuration file and click OK.

d In the Value field, type the name of the EIM configuration file and click Save. The default value is default.ifb.

NOTE: You can use the Uniform Naming Convention (UNC) filename when specifying the EIM configuration file if you have read permission to the path.

e To set the batch number for the EIM process, repeat Step a through Step b on page 238. Select Batch Number and click OK. In the Value field, type the batch number and click Save. The default value is 0.

f To select the EIM process from the configuration file, repeat Step a through Step b on page 238. Select Process and click OK. In the Value field, type the process name and click Save.

g Optionally, to activate error flags, repeat Step a through Step b on page 238. Select Error Flags and click OK. In the Value field, type 1 and click Save. The default value is 0.

For information on error flags, see *Siebel Enterprise Integration Manager Administration Guide*.

h Optionally, to activate SQL trace flags, repeat Step a through Step b on page 238. Select SQL Trace Flags and click OK. In the Value field, type 8 and click Save. The default value is 0.

For information on SQL trace flags, see *Siebel Enterprise Integration Manager Administration Guide*.

i Optionally, to activate trace flags, repeat Step a through Step b on page 238. Select Trace Flags and click OK. In the Value field, type in the appropriate value and click Save.

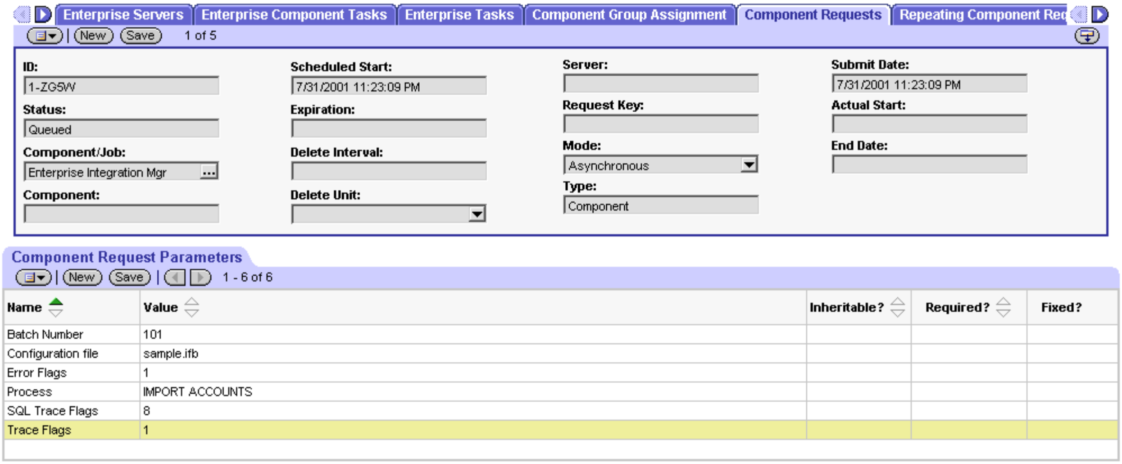
For information on trace files, see *Siebel Enterprise Integration Manager Administration Guide*.

NOTE: You will need to identify at least a batch number, process name, and configuration file for the task. If the batch number component parameter is set to 0, the batch number in the EIM configuration file (if any) will be used.

3 In the Component Requests form, click the menu button, and then click Submit request.

Caution: EIM is a multistep process. After the EIM process is running, do not stop or pause the task. Otherwise, some steps may not roll back correctly.

Figure 38 shows an example of running an EIM process as described.



ID: 1-ZG5W

Status: Queued

Component/Job: Enterprise Integration Mgr

Component:

Scheduled Start: 7/31/2001 11:23:09 PM

Expiration:

Delete Interval:

Delete Unit:

Server:

Request Key:

Mode: Asynchronous

Type: Component

Submit Date: 7/31/2001 11:23:09 PM

Actual Start:

End Date:

Name	Value	Inheritable?	Required?	Fixed?
Batch Number	101			
Configuration file	sample.ifb			
Error Flags	1			
Process	IMPORT ACCOUNTS			
SQL Trace Flags	8			
Trace Flags	1			

Figure 38. Running an EIM Process

To reactivate Docking Transaction Logging

1 Navigate to the System Preferences screen.

2 Select Docking: Transaction Logging.

3 In the System Preference Value field, type TRUE.

4 Click outside the row to save changes.

Checking the Results of the Data Import

Your EIM log file contains comprehensive status and diagnostic information about the import process. NREC's EIM log files are located in the *siebel_srvr_root*\log\server directory, the default.

To review your import process

- View the EIM log file for the corresponding task number.

  The log file name is Eim_*task#*.trc, where *task#* is the Task Number for the EIM process that you ran.

In addition, you can check the status of each row EIM tried to import by using the IF_ROW_STAT column. For more information, see *Siebel Enterprise Integration Manager Administration Guide*.

Required Application Administration Tasks 12

This chapter describes how NREC completed the general setup tasks in the Test environment. This chapter assumes a successful installation of Siebel products in the Test environment. For an overview of NREC's installation process, see Chapter 2, "Installing Siebel Applications."

All tasks in this chapter are performed in the administration views of the Siebel Web client or dedicated Web client. Assume NREC accessed the views by logging on to the Partner Manger application. The same views are accessible through other employee applications, such as Call Center. Although the tasks in this chapter are written from the perspective of an NREC employee setting up the application in the test environment, you can follow along using an application logged on to the Sample database.

NOTE: Generally, you should always perform system administration tasks against the server database. Although you can perform these tasks against your local database and synchronize, there is the potential for error, including data conflicts, performance problems resulting from a large local database, and large numbers of transactions to route.

The tasks in this chapter do not cover Access Groups and Catalogs, which allow you to control access to master data. Because NREC did not use Access Groups or Catalogs in their implementation, the tasks for setting them up are not covered in this chapter. For more information about setting up Access Groups and Catalogs, see *Applications Administration Guide* and *Security Guide for Siebel eBusiness Applications*.

Logging On as the Siebel Administrator

To perform the tasks in this chapter, you must log on to your Siebel application using the *administrator* user name and password assigned by your database administrator. The default values are SADMIN/SADMIN. You can use SADMIN/SADMIN to log in to the Siebel Sample database to walk through the tasks in this chapter.

To log on as an administrator to a server datasource

1. Open your Web browser.
2. Go to the URL for your application.

   The login screen appears.
3. Enter SADMIN in the User ID and Password fields.
4. Click OK.

   Logging on as SADMIN allows you to see all the screens and views mentioned in this guide.

To log on as an administrator to the sample datasource

1. Open the Mobile Web Client on your local machine.

   For the NREC example, log on to Siebel Partner Manager.
2. Enter SADMIN/SADMIN as the user name and password.
3. Select Sample as the datasource.

   Siebel Partner Manager opens, displaying data from the Sample database.

NOTE: If you modified tabs as described in Chapter 9, “Modifying the Look and Feel of the Web Client,” the images for the off state may appear in your employee application. You need to replace the original images to return the user interface to the original look and feel. For more information about tab images, see “Modifying Tabs” on page 210.

Defining Views

Often developers will add views to the system during the configuration stage of the implementation process. You need to register new views in the application before they will appear in the user interface. You did this during the development cycle, in order to unit test the new views (see "Testing Changes" on page 63). However, you registered the views in your local development environment; now you need to register them in the Test environment.

Use the View Administration view to register new the views in your system. You must do this before you can associate views with responsibilities.

NOTE: You should turn off any of the forecast views that you won't be using based on your organization's forecasting configuration decision.

To register views

1. From the application-level menu, choose View > Site Map > Application Administration > Views.
2. Add records for the new views.

   For example, new views for the NREC example, are:

   - House Detail View
   - House Detail - Appraisals
   - House Detail - Renovations

   The view name must match the view name as it is defined in the Siebel repository. You can copy the view name from Siebel Tools and paste it into the View list to avoid mistakes.

NOTE: You generally do not need to modify or delete records for registered views. A view record would require modification only if its name was changed. The only time a view may need to be deleted would be if it no longer existed or if you didn't want anyone to have access to it. Even then, a preferable approach would be to simply not assign it to any responsibilities.

Defining Company Structure

The first step in setting up your Siebel application is to define your company's structure. This includes defining organizations, divisions, positions, and responsibilities.

NOTE: Changing your company structure—such as positions and divisions—can cause routers to re-evaluate visibility for all objects related to the objects that have changed. This action can result in a performance impact. For more information, see *Siebel Remote and Replication Manager Administration Guide.*

Understanding Company Structure

Defining your company's structure involves setting up the following entities:

- **Organizations and organization skills.** Organizations represent the broad divisions of your company. They can represent internal or external groups of users—for example, internal divisions or partners. Organizations are used to control access to data.
- **Divisions.** Divisions belong to organizations but do not affect visibility. Divisions are used to group positions, record addresses, and maintain default currencies.
- **Positions and position skills.** Positions represent specific job slots. Positions determine which record users can access.
- **Responsibilities.** Responsibilities control the screens and views a user can access.

For more information about setting up company structures, see *Applications Administration Guide*.

Understanding Access to Data

How you define your company's structure in your Siebel application affects the records and views to which users have visibility, or access.

There are several relationships between organizations, responsibilities, positions, and people.

There is a many-to-many relationship between a person and that person's positions. For example, an individual could hold one or more positions—Sales Representative for the X territory definition and Sales Manager for the XYZ region. Conversely, more than one person may be assigned to a single position, such as when two people share a single job.

In addition to having one or more positions, each user is then assigned one or more responsibilities. Responsibilities determine which views users see; positions determine which records users have access to.

In a situation in which a user is associated with one responsibility and multiple positions, that user would have the same set of views regardless of which position that user logged on with. However, if the user changed position, that user would see different data. A user who is logged on as a sales representative sees only data (accounts, opportunities, contacts, and so on) that pertains to this position. Even though the user still has the My Team's Accounts view, the position the user is logged on as does not have any other positions reporting to it so the user does not see other data.

NOTE: A user can change positions while logged in by choosing View > User Preferences > Change Position and selecting a different position in the list.

If the user changes to the Sales Manager position, the My Team's Accounts view will now show the user information for all of the positions reporting to that user if the user is the primary on the account. Again, the views have not changed, but the data in the views has.

Having organizations adds another level of visibility. Some data records will be unavailable depending on what organization a user's position belongs to. (Users can also be given more visibility within their organization by receiving access to certain views.)

For more information on organizations and access to data, see *Applications Administration Guide* and *Security Guide for Siebel eBusiness Applications*.

Setting Up Organizations and Organization Skills

Setting up organizations is an optional step in an implementation. If you don’t set up additional organizations, you will automatically have everything assigned to a default organization. Having one organization is like having no organization. There is no impact on visibility and data access.

Because NREC is using Partner Portal, it will set up one parent organization called NREC and then a child organization for each of its partner real estate agencies. This structure will allow NREC to logically group these different user groups and partition data accordingly.

Initially NREC creates the NREC organization. Then after the rest of the company structure is set up—positions, responsibilities, and so on—NREC will use Siebel Partner manager to set up partners and promote them to organizations with NREC as the parent organization. See “Registering Partners and Creating Partner Organizations” on page 260.

For more information on organizations and company structure, see *Applications Administration Guide*.

To set up an organization

1. From the application-level menu, choose View > Site Map > Group Administration > Organizations.

   The Organizations view appears.

2. Add a new record.

   Some fields are listed in the following table.

Field	Value	Description
Name	NREC Organization	Unique name for the organization. It is recommended that you include the word *organization* in the name.
Partner Manager Position	SADMIN	Used for Partner Portal to represent the person in the organization who manages the relationship with partners. For more information, see *Siebel Partner Relationship Management Administration Guide*.
Partner Flag	FALSE	Indicates whether the organization represents an external enterprise.

NOTE: Organizations cannot be deleted.

Setting Up Divisions

Divisions belong to organizations and are used to record addresses and to maintain default currencies. User reporting structures are defined by their parent position, but their country of operation and currency are defined by their division. You must have at least one division set up in order to implement Siebel eBusiness Applications.

A division is automatically created when you create an organization. For example, the NREC division was created when you created the NREC Organization.

To review the division created for your organization

1. From the application-level menu, choose View > Site Map > Group Administration > Divisions.

   The Divisions list appears.

2. In the Divisions list, query for the Organization you created in "Setting Up Organizations and Organization Skills" on page 248.

   For example, query for NREC.

3. Review the values defined for the division.

Setting Up Positions and Position Skills

An employee has to have a position in order to log on to the Siebel application. Positions determine which records users with a particular position can access. Positions represent a job slot in your organization. As you enter your company's positions, refer to your company's organization chart to determine reporting relationships (positions and parent positions). There is always one position that does not have a parent position. For instance, the CEO position probably does not report to a higher level. Define positions in each level of your company's division hierarchy.

Because you choose parent positions as you create new positions, you should start at the top of the organization chart and work your way down.

To create positions

1. From the application-level menu, choose View > Site Map > Group Administration > Positions.

   The Positions view appears.

2 In the Positions list, add a new record for each of the positions you want to define.

Some fields are described in the following table.

Most fields in the Position list are filled in automatically from the Employee record of the active employee. If you have not set up employees, you can associate them with positions later.

Field	Comments
Billing Product	Used for Professional Services. For more information, see *Applications Administration Guide*.
Compensable	Used for incentive compensation. For more information, see the *Siebel Incentive Compensation Guide*.
End Date	Last day for the currently associated employee to be associated with this position. After this date, the employee will no longer have visibility to the records associated with this position.
Organization	Select an organization for the position. A position can only have one organization. If you want a user to have visibility to some organizations, but not all organizations, then you must create a position for each organization and assign the employee to each position. The employee can then see one organization’s data at a time, by choosing View > User Preferences > Change Position.
Position	The name of the position. Required.
Territory	Allows a position to be associated to a territory for use by the Assignment Manager module. For more information, see *Siebel Assignment Manager Administration Guide*.
Position Type	Informational field that indicates the type of position. It has no impact on visibility.

The following table lists sample positions for the NREC example. Later, as described in the section “Setting Up an Employee in Your Siebel Application” on page 257, you will add the employees listed in the last column.

Position	Parent Position	Employee
CEO	N/A	Ben Warren
Sales Manager	CEO	Joe Smith
Information Technology Manager	CEO	Kelly Brown
Administrator	Information Technology Manager	Lisa Bolivar
Real Estate Agent 1	Sales Manager	Leslie Castro
Real Estate Agent 2	Sales Manager	Michael Stevens
Marketing Associate	Sales Manager	Pablo Williams
Partner Manager	Sales Manager	Pam Frederick

Most fields in the Position applet get filled in automatically from the Employee record of the active employee. For example, the Start Date field shows the start date of the employee marked Active for the position.

To set up position skills

1. From the application-level menu, choose View > Site Map > Group Administration > Positions.

   The Positions view appears.

2. In the Positions list, select the position to which you want to add skills.
3. Click the Position Skills tab.
4. In the Position Skills list, add a new record.
5. Select a skill in the Item field.

   Skills must exist before you can add them. Skills are added using Siebel Tools.

6. Add comments, if desired.
7. In the Position Skill Item list, add a new record.

Defining Responsibilities

Responsibilities determine which views users can access. For example, the System Administrator responsibility allows access to all views. Defining responsibilities lets you limit user access to views and, therefore, to your Siebel application's information and functions.

You should define responsibilities that correspond to the major job functions in your organization. For example, you might create responsibilities for NREC CEO, sales manager, partner manager, field sales representatives, and so on.

- In the NREC example The Real Estate Agent responsibility might have a select set of views in Siebel Sales, such as Opportunities, Contacts, Accounts, Activities, and so on.
- The sales manager responsibility might have access to the same screens as the sales representative, plus additional views.
- The CEO responsibility might have access to all the views except applications administration.

To define a responsibility, you must specify which views are available to that responsibility. You may prefer to copy the sample responsibilities that ship with your Siebel application and then customize them for your purposes.

NOTE: You cannot edit sample responsibilities.

You should grant access to the System Preferences view to only a select group of administrators. System preferences control server logic and processing. Therefore, end users should not be given access to the System Preferences view.

You should not add Applications Administration screens to responsibilities associated with end users. Likewise, you should limit access to the Employees, Master Forecasts, Mobile Clients, Responsibilities, Views, and Territories screens. The work performed using these screens has far-reaching implications for the entire application.

You may want to hide the license key view to discourage unauthorized users from attempting to change license keys. You can do this using the License Keys view in the Applications Administration screen. To hide the license key view, disable the Siebel License Key view in the user's responsibility.

NREC created new views during the development cycle and then registered the views in the application. Now NREC must add these views to the responsibilities that need to have access to them.

NOTE: You should add all new views to the Siebel Administrator responsibility, because the people with this responsibility typically help troubleshoot the Siebel application.

To copy an existing responsibility

1. From the application-level menu, choose View > Site Map > Application Administration > Responsibilities.

   The Responsibility Administration view appears.

2. Select the responsibility you would like to copy.

   NOTE: You cannot edit the sample responsibilities. You must copy them before you can modify them.

3. Click the menu button and then choose Copy Record.

   The views for the existing responsibility are copied, but the users are not.

4. Add any other desired views to the responsibility.

   For the NREC example, add the following views:

   - House Detail View
   - House Detail - Appraisals
   - House Detail - Renovations

5. Enter a new name for the responsibility, such as NREC Partner Real Estate Agent.

6 Enter a new description for the responsibility so that you can later identify its function.

7 Select an organization for the responsibility.

   By default, the Responsibility Administration view shows all responsibilities, regardless of organization. However, you may want to configure new views in Siebel Tools that restrict the visibility to responsibilities as a way to delegate administration of responsibilities. For more information on configuring views, see *Siebel Tools Reference*.

Now you must set up your users, so they can be assigned responsibilities. Without responsibilities, a user cannot use the Siebel application.

Setting Up Users

Before users can access Siebel data, you must define users in your database server, your Siebel application, and your authentication environment. Your database administrator creates database accounts and optionally encrypts their passwords. Depending on your user authentication environment, each user may have an individual database account, or accounts may be shared by multiple users. If you implement an external authentication system, you must configure user authentication including associating users with database accounts.

For the current example, assume NREC is setting up each user with an individual database account and is not using an external authentication system.

For more information about setting up user authentication, see *Security Guide for Siebel eBusiness Applications*.

For more information about setting up users, see *Applications Administration Guide*.

Setting Up Database Users

Before you can set up someone as a Siebel application user, that person must have a database account user name.

Add the database accounts on the appropriate database and add these accounts to the group SSE_ROLE.

The exact steps for adding users and placing them in this role group depend on the database software your organization is using. Work with your database administrator to set up developers as database users in the SSE_ROLE group.

Types of Users

Siebel eBusiness Applications recognize several types of users:

- **Employees.** Individual internal users. Employees typically have login privileges.
- **Partners.** Individual users in partner companies to whom your company may grant access to some of its data. A partner needs a position in order to access views. See "Registering Partners and Creating Partner Organizations" on page 260.
- **Users**. Internal and external users with login privileges. Customers for customer applications to whom your company may grant some data access. A user does not need a position to access views.
- **Persons**. Includes employees, partners, users, and contacts. Persons may or may not have login privileges.

Setting Up an Employee in Your Siebel Application

After users have a database account, you can set up these database users as Siebel application users.

As the Siebel application administrator, you are able to:

- Add or delete a user
- Add multiple positions, responsibilities, organizations, and territories to a user
- Remove all positions connected to a user

This task can be done by an administrator in your organization or an administrator at a partner organization whose responsibility allows them access to the view.

Initially, you will want to set up employee records. However, you will also need to set up partners, users, and persons. The procedures for setting up other types of users are similar to the procedures for setting up employees.

To set up an employee

1. From the application-level menu, choose View > Site Map > User Administration > Employees.

   The Employees view appears.

2. Add a new record.

3. Select at least one responsibility. You can also add organizations and positions.

4. Complete the additional fields.

   Some fields are described in the following table.

Field	Comments
User ID	Enter the user ID for this employee in uppercase letters for compatibility across database systems.
Password	Applicable only when using LDAP or Microsoft Active Directory for login authentication. Otherwise, it is not editable. For more information about system-level parameter setup to make the field editable when using LDAP or Microsoft Active Directory, see *Security Guide for Siebel eBusiness Applications*.
Position	The user's position. Positions and their employees are defined for NREC in the section "Setting Up Positions and Position Skills" on page 250.
Responsibility	The user's responsibilities. Associating responsibilities with users is described in the section "Associating Responsibilities with a User."

5. Repeat these steps for each employee you need to add.

Associating Responsibilities with a User

When you are first implementing your Siebel application, you define responsibilities, as described in "Defining Responsibilities" on page 253. After you have defined responsibilities, you assign them to users as described in "Setting Up Users" on page 256. This process allows you to assign user responsibilities at the same time as you enter other user information.

After your initial implementation, you may work directly from the Responsibility Administration view, or you may continue to work from the Employee Administration view to change or add associations between users and their responsibilities.

To associate a user with a responsibility

1. From the application-level menu, choose View > Site Map > Application Administration > Responsibilities.

   The Responsibility Administration view appears.

2. Select a responsibility that you wish to assign to a user.
3. Select the User list applet.
4. Click the menu button and then choose Edit > Add New Record.
5. Select one or more employees from the Add Employees dialog box, and then click Add.

   The added employees now have access to the views listed in the Views applet.

Registering Partners and Creating Partner Organizations

This section covers how NREC set up partners in their test environment to resemble its business model. NREC will create an organization for each of its partner real estate agencies. Additionally, NREC will set up the company structure for each partner, including positions, responsibilities, and employees. This section uses one partner, San Francisco Real Estate, as an example.

For detailed information on managing partners, see *Siebel Partner Relationship Management Administration Guide*.

Registering Partners and Promoting Them to Organizations

NREC will register each partner and then convert it to an organization within the NREC company structure.

To register a partner and then promote it to an organization

1. From the application-level menu, choose View > Site Map > Partner Administration > Approved Partners.

   The Approved Partners view appears.

2. In the Approved Partners list, click New to enter a new record representing the partner. For example, enter San Francisco Real Estate.

   A Partner form appears, allowing you to enter the record.

3. Save the record.

   The record is displayed in the Partners list.

4. Click the Register button.

   A form appears allowing you to select the Organization flag, which will automatically promote the partner to an organization.

5. Select the Organization flag and enter NREC as the parent organization.

   The partner is moved to the Registered partners list and an organization is created for the partner as a child of the parent organization—NREC.

Creating Positions, Responsibilities, and User Assignments

Now that you have created a registered partner, you can add the positions, responsibilities, and user assignments for the partner.

To create positions for a registered partner

1. From the Show drop-down list, choose Registered Partners.
2. In the Partners form, query for the partner you created in the previous procedure.

   For example, San Francisco Real Estate.
3. Enter new records using the following views:
   - **Positions.** Enter positions for the partner organization.
   - **Responsibilities.** Enter responsibilities for the partner organization.
   - **User Assignments**. Enter employee records for the partner organization.

Assignment Manager 13

In previous chapters, you used Siebel Tools to configure the business logic of NREC's Partner Portal application. In addition to Siebel Tools, there are other mechanisms that allow you to configure business logic as well. These include modules such as Siebel Business Process Designer, Siebel Personalization, and Assignment Manager. These modules are administered at run time using views in the Siebel Web Client. They allow you to modify business logic without having to recompile the Siebel .srf file. This chapter describes how NREC used Assignment Manager to implement business rules. Subsequent chapters cover Business Process Designer and Personalization.

Assignment Manager allows you to create business rules that automatically assign entities such as opportunities, service requests, or activities to the appropriate individuals. For example, NREC wants to automatically assign opportunities to employees at partner real estate agencies. When an opportunity is entered into the system, it contains the ZIP Code where the customer is interested in buying property. Assignment Manager will use this attribute to automatically assign the opportunity to a real estate agent located within that ZIP Code. After assignment rules are defined and made active, they will be automatically triggered whenever a new opportunity is created.

Configuring Territory Assignment Components

NREC’s assignment rules will use the ZIP Code attribute on the opportunity to determine how to assign the opportunity. This is a custom attribute that you added to the opportunity business component in “Adding Fields to the Opportunities Business Component” on page 146.

Because this is a custom attribute, NREC would need to add a new assignment criteria using Siebel Tools before being able to create assignment rules. This section summarizes the high-level tasks necessary to create the assignment criteria. For detailed information about configuring assignment attributes, see *Siebel Assignment Manager Administration Guide*.

The tasks to create the Opportunity ZIP Code assignment criteria are:

- Create a new Workflow Policy Column
- Define a new Workflow Policy Component Column as a child record of the existing Opportunity Workflow Policy Component
- Create a new Assignment Attribute and Assignment Attribute Column
- Create a new Assignment Criteria and Assignment Criteria Attributes

NOTE: Defining assignment criteria is only necessary when the predefined assignment criteria do not meet your organization’s business needs. The tasks are summarized here to give you an idea of the underlying configuration that supports the subsequent sections of this chapter.

Creating Assignment Rules Based on Territories

The first step is to create assignment rules based on territories. For example, three of NREC's territories are San Francisco, Peninsula, and East Bay. Each of these territories is served by a partner real estate agency. Each real estate agency is associated with a set of ZIP Codes, which you will define as criteria values in the next procedure.

- San Francisco: 94101, 94102, 94103
- Peninsula: 94301, 94401, 94010
- East Bay: 94701, 94602, 94580

For the NREC example, you would need to create an assignment rule for each territory listed above. The following procedures use San Francisco as an example.

To create assignment rules based on territories

1. From the application-level menu, choose View > Site Map > Assignment Administration > Assignment Rules.
2. In the Assignment Rules list, click the Menu button and then choose New Record.
3. Enter a new record using the following values:

Field	Example Value	Description
Name	San Francisco	Unique name for the rule
Object	Opportunity	The object to be assigned
Minimum Score	0	Although NREC is not using scoring, you must enter a value here.
Assignees from Rule	All Above Minimum	Use all assignees with an assignment score greater than or equal to the assignment rule's minimum score. For a description of other methods see *Siebel Assignment Manager Administration Guide*.

Specifying Assignment Criteria and Values for Rules

After defining assignment rules for each territory, you need to define the assignment criteria and criteria values.

- **Assignment criteria.** Criteria used to determine which candidates qualify to receive the assignment. In the NREC example, the criteria will compare the attributes of an opportunity and the attributes of a partner employee. Partner employees that have appropriate attributes will qualify for the assignment.
- **Criteria values**. Specific values or range of values for a given criterion. Criteria values are used for comparison. In the NREC example, the values for the assignment criteria are the ZIP Codes that fall within a particular territory.

To specify territory assignment criteria and values

1. Navigate to the Assignment Rule View and select the assignment rule for San Francisco that you created in the previous section.
2. Click the Criteria view tab.

   The Assignment Rule is displayed in the top form applet and Assignment Criteria list is displayed below it.

3 Add a new record in the Assignment Criteria list.

Field	Value for NREC Example	Description
Criteria	Opportunity ZIP Code	Defines the type of criteria to be used. In this case, the NREC developer needed to create the assignment criteria, because the attribute being used is a custom attribute defined by NREC. For information about defining assignment criteria, see *Siebel Assignment Manager Administration Guide*.
Comparison Method	Compare to object	Method used to match objects to candidates. For example, Compare to object compares criteria values to object attributes.
Score	0	Candidates that satisfy this criteria have this score added to their total score.
Inclusion	Include	Method used to match criteria values and candidates.
Required	Always	Determines whether the criteria is required to qualify.

4 In the Values List applet (located below the Criteria List applet), add new records for each of the ZIP Codes included in this assignment rule.

For example, for the San Francisco territory assignment rule, you would enter records for:

- 94101
- 94102
- 94103
- And so on

Adding Positions for Each Assignment Rule

Just as each rule has criteria that specify when the rule takes effect, each rule also has a list of positions, employees, or organizations that specifies the candidates for the assignment. In the NREC example, the assignment is based on positions. Each assignment rule will be associated with the positions at the respective partner real estate agency. For example, the positions at the San Francisco Real Estate Agency will be associated to the assignment rule for the San Francisco territory.

To add positions to an assignment rule

1. Navigate to the Assignment Rule view, and then select the assignment rule for San Francisco.

2. Click the Positions view tab.

3. Click the menu button and then choose New Record.

   The Add Positions dialog box appears.

4. Select the Positions for the partner real estate agency that you want to associate with this rule.

   In the current example, these will be the positions for the San Francisco Real Estate Agency.

   - SF Real Estate Sales Manager
   - SF Real Estate Agent 1
   - SF Real Estate Agent 2

Releasing the Assignment Rules

After you have created and defined assignment rules, you must release them to instruct Assignment Manager to use these rules.

NOTE: This procedure releases all assignment rules. Do not release assignment rules while associated server tasks are running.

To release assignment rules

1. Navigate to the Assignment Rules view, and then select the rule for San Francisco.
2. Click the Release button.

   Assignment rules are released to Assignment Manager for use.

Activating the Rules

To activate Assignment Manager rules, you must start several Siebel Server components.

However, it is best to start the components just once—after you have created all your rules in both Assignment Manager and Business Process Designer. For this reason, instructions on starting the components appear only at the end of the Business Process Designer chapter, in "Activating the Rules" on page 284.

For more information about Assignment Manager, see the *Siebel Assignment Manager Administration Guide*.

Siebel Business Process Designer 14

There are many ways to customize a Siebel application. In the last chapter, you saw one way—writing assignment rules using Assignment Manager. This chapter introduces another way to customize your application—using Business Process Designer to automate business processes.

For example, NREC plans to use Siebel Business Process Designer to automatically send an email notification to partner real estate agents when an opportunity has been assigned to them. The email will be based on a simple template, and will inform them that they have a new opportunity in their queue. A *workflow process* will define the steps necessary to send the email. A *workflow policy* will trigger the process when it detects that an opportunity has been assigned. This chapter describes the tasks required to set up this business rule, including:

- "Creating an Email Template" on page 273
- "Creating a Workflow Process" on page 275
- "Creating a Workflow Policy" on page 281
- "Activating the Rules" on page 284

All the work in this chapter is done using the administration screens in NREC's test environment.

Configuring Siebel Communications Server

To send email using Siebel Business Process Designer, you must successfully configure your email communications system and configure Siebel Communications Server. This includes tasks such as:

- Setting up your Siebel Server machine to interface to different messaging systems that Siebel Communications Server supports
- Configuring and administer the Siebel Communications Server components you need, including the Communications Outbound Manager server component
- Configuring Siebel Communications Server communications drivers and profiles
- Configuring additional recipient groups for your communications, based on Siebel business objects and business components

For detailed instructions on performing these configuration tasks, see *Siebel Communications Server Administration Guide*.

Creating an Email Template

Before you can set up a workflow process to send an email message, you must first configure an email template. An email message sent from the Siebel application uses an email template, similar to how a form letter uses a template. Some template elements are replaced in the final message text. For example, Siebel field names are substituted with field values from database records. Email templates are created using the Communications Administration screen.

There are two kinds of email templates: simple and advanced. Simple templates are used for commands such as Send Email, Send Fax, Send Wireless Message, and Send Page, and for replies in Siebel eMail Response. Advanced templates are used for outbound communication requests, such as the request triggered by a workflow process. The template discussed in this section is an advanced template.

To create an email template

1. From the application-level menu, choose View > Site Map > Communications Administration > All Templates.

   The All Templates view appears.

2. In the Templates list, enter a new record using the following values:

Field	Value
Name	NREC Email template
Channel Type	Email

3. Click the Advanced view tab.

4 In the Advanced form, enter a new record using the following values:

Field	**Value**	**Comments**
Delivery Profile	Default SMTP Profile	Created for the Internet SMTP/ POP3 Servers communications driver, for an email template. The profile must exist before you create the template. In this profile, the communications sender should be defined using the From Address parameter override.
Recipient Group	Opportunity Sales Team	The recipient group you specify here only determines which substitution fields are available for the template text. The actual recipients are determined in the context in which the template is used.
Subject	New Opportunity in Queue	Default subject line for the email. The subject line can also include substitution fields.
Public	TRUE	Determines whether the template will be available to all users.
Available Substitutions	N/A	Fields available for substitution. You can copy and paste these fields into the body of the template, or in the subject line. Note that fields must be enclosed with brackets ([ ]) in the template text.
Template Text	Dear [Full Name], A new opportunity has been assigned to you. Thank you, NREC	Text to appear in the body of the template. The items in brackets are are substitutions.

For detailed information about email templates and fields, and about communications drivers and profiles, see *Siebel Communications Server Administration Guide*.

The email template is now available to use in the workflow process.

Creating a Workflow Process

A workflow process defines the series of actions you want to occur in the workflow. After the workflow process is triggered, it performs the specified actions. In general, a workflow process consists of one or more process steps, which can be Start steps, decisions, invocations of business services methods, subprocesses, or other types of steps. In the current example, the steps are simple:

- **Start.** Each workflow process must begin with a start step.
- **Send Email business service.** Predefined business services are available to use for performing actions such as sending email.
- **End.** Each workflow process must end with an End step.

You use the Workflow Processes view (part of the Siebel Business Process Administration screen) to create the workflow process.

Using the Business Process Designer

Figure 39 shows the Business Process Designer with these steps for the current example.

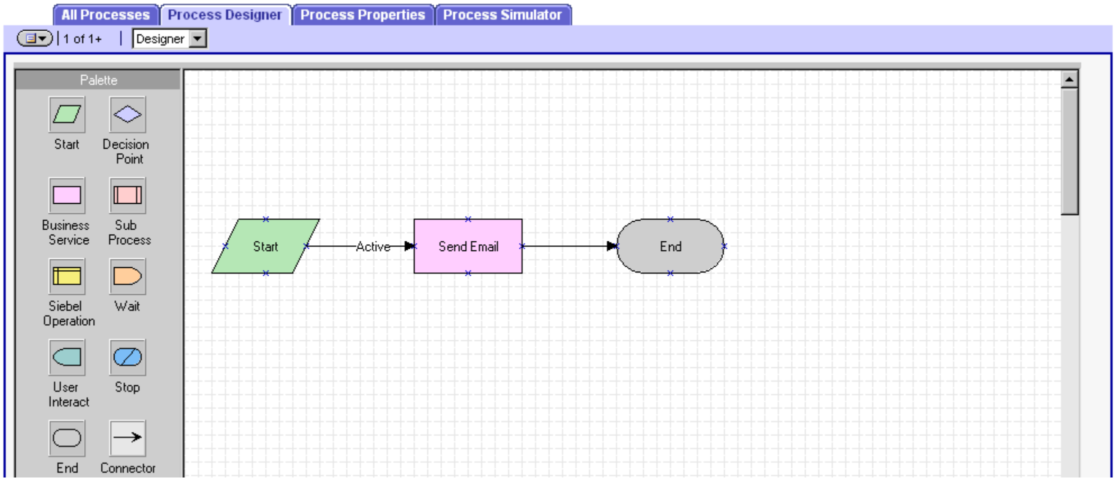


Figure 39. Business Process Designer View with Flowchart of Current Example

The flowchart in Figure 39 on page 275 shows the result of work in this and other Siebel Workflow views.

To create a workflow process

1 From the application-level menu, choose View > Site Map > Siebel Business Process Administration > Workflow Processes.

2 In the Workflow Processes list, enter a new record with the following values.

Field	Value	Comments
Name	NREC New Opportunity Notification	Unique name that identifies the process.
Business object	Opportunity	Business object is Opportunity because the goal is to trigger an action based on changes to the Opportunity record. The pick list contains only those business objects that have a primary business component defined.
Group	Partner Portal	Groups are logical collections of workflow processes. You can select an existing group or enter a new one in the list of values table. See *Applications Administration Guide* for information on how to enter list of values.

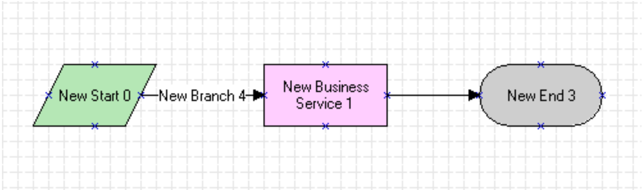
3 Open the Business Process Designer by choosing the Process Designer view tab.

The Business Process Designer view appears.

4 Use the Designer applet to drag appropriate icons from the Palette to the flowcharting workspace.

Notice that different shapes and colors are used to represent different types of steps. In the current example, a pink rectangle represents a business service. For information about the different types of steps, see *Siebel Business Process Designer Administration Guide.*

5. Drag Connectors from the Palette to the flowcharting workspace to connect the steps of the process.



6. Double-click each step icon in your flowchart to drill down to a view and define the specific values for that step.

   For example, double-clicking the Send Contact Email icon takes you to the Business Service Arguments view, in which you define this particular business service. For details about which values NREC supplied for the current example, see "Details About the Steps in the Sample Process" on page 278.

7. After defining the details of each workflow, navigate back to the Workflow Processes list, make sure the workflow process is selected, and then click the Activate button.

Details About the Steps in the Sample Process

This section explains the NREC-specified values for each of the four steps in the process described in "Creating a Workflow Process" on page 275.

Start

Double-clicking the Start step takes you to the Start form. This section tells you what values NREC specified for this step.

In the Start form applet, the only value NREC changed is the value of Name (setting it to Start, from its default value of New Start 0).

In the Next Steps list applet, NREC modified the existing record with the field values shown in Table 13.

Table 13. Field Values for the Next Steps Applet

Field	Value	Comments
Name	Active	This value specifies the name of the connector from the Start step to the next step.
Type	Default	This value indicates that the next step should be the one pointed to by the connector from Start.

NOTE: Because the Type field is set to Default, the value for the Next Step field will be automatically populated after you define the next step.

Send Contact Email (a Business Service Step)

Double-clicking the Business Service step icon takes you to the Business Services view.

In the Business Service form applet, NREC specified the field values shown in Table 14.

Table 14. Field Values for the Business Service Applet

Field	Value	Comments
Name	Send email	Unique name.
Business Service	Outbound Communications Manager	Specify this value because you want to send an email using the Siebel Communications Server.
Method	Create and Submit Request	Choose this method because you want to send the email template that you created earlier. For information about business services and their methods, see *Siebel Communications Server Administration Guide* or *Siebel Business Process Designer Administration Guide.*

In the Input Arguments list applet, NREC created four records with the field values shown in Table 15. For a detailed description of the arguments required for predefined business services, see *Siebel Business Process Designer Administration Guide.*

Table 15. Field Values for Input Arguments Applet

Input Argument	Type	Value	Property Name	Property Data Type
Request Name	Literal	NREC Request		
Recipient Group	Literal	Opportunity Sales Team		
Comm Template Name List	Literal	NREC Email Message		
Source Id List	Process Property		Object Id	String

An Outbound Communications Manager business service does not take any output arguments.

End

Double-clicking the End step takes you to the End Arguments view. The only value NREC changed in this view is the value of Name (setting it to End, from its default value of New End 0).

After you finish creating the process by defining all the steps, you must create a policy that calls this process. A Workflow policy consists of a condition and an action. The action in this example is to invoke the Account Status Change process. For instructions on defining the action and conditions, see the following section, "Creating a Workflow Policy."

Creating a Workflow Policy

Workflow policies define the conditions under which a workflow process should be invoked. In this example, the condition is when an opportunity is assigned by Assignment Manager. When this occurs the workflow policy will invoke the workflow process defined in "Creating a Workflow Process" on page 275.

Creating a Policy Action

Workflow policy actions define the events that you want to occur when the conditions of your Workflow policy are met. In the NREC example, the action is to run a workflow process.

To create a policy action

1. From the Show drop-down list, choose Workflow Policy Actions.
2. In the Actions list, add a new record to specify what actions the policy should trigger.

   In the current example, the required fields and their values are:

Field	Value	Comment
Name	NREC Opportunity Notification	The name of the action can be any name meaningful to you.
Program	Run Workflow Process	This value specifies that the policy will trigger a workflow process.
Workflow Object	Opportunity	Name of the object associated with the workflow policy.

3. In the Arguments list, add records for the necessary arguments.

   In the current example, the required fields and their values are:

Argument	Value	Comment
Argument	ProcessName	Since you previously specified Run Workflow Process, ProcessName is the only item in the pick list.
Value	NREC Opportunity Notification	This value is the name of the process you created as described in "Creating a Workflow Process" on page 275. The workflow process must be active for it to appear in the drop-down list for the Value field.

Creating a Policy Condition

Now that you have defined the event that you want to occur as a Policy Action, you can define the conditions under which the action will occur. You do this by defining a Policy Condition.

To create a policy condition

1. From the Show drop-down list, choose Workflow Policies.
2. In the Policies list, enter a new record.

   For the current example, complete the fields described in the following table.

Field	Value	Comment
Name	NREC Opportunity Notification	This name can be any name that is meaningful to you.

Field	Value	Comment
Workflow Object	Opportunity	Choose this value because the action is based on changes to an Account field.
Group	NREC	Since this group does not currently exist, you must create it: **1** Click the Select button in the Group field. **2** In the Workflow Groups dialog box, click New. **3** Specify NREC in the Name field. **4** Click Save. **5** Click OK.

3 In the Conditions list, enter the following conditions.

Condition Field	Operation
Position ID	Is Added

4 In the Actions form, enter the following record.

Field	Value	Comments
Action	NREC Opportunity Notification	This is the name of the Workflow Policy Action you created in "Creating a Policy Action" on page 281.
Sequence	10	Sequence of the Actions.

Activating the Rules

To activate rules—whether they are Assignment Manager rules or Workflow rules—you must run the following Siebel Server components:

- Generate Triggers
- Workflow Monitor

Running Generate Triggers

The Generate Trigger (GenTrig) component on the Siebel Server allows you to create database triggers. The Workflow Policies module of Siebel Business Process Designer uses these database triggers to identify which records may match policy conditions. The Generate Triggers component needs to be rerun whenever new policies are created or deleted.

You can run the Generate Triggers component from either the Server Manager graphical user interface, or command line mode. The following procedure is for the GUI. For information about running Server Manager using the command line interface see *Siebel Server Administration Guide*.

To generate triggers using the GUI

1. In the Siebel Client, select View > Site Map > Server Administration >Enterprise Operation > Component Request.
2. Click New.
3. Select Generate Triggers from the Component Job list.

   This creates a new line entry but does not start the task.

4. Click New in the Component Request Parameter list and define the following parameters:

Name	Value	Comment
Trigger File Name	TRIGGER.SQL	This filename identifies the SQL script that needs to be run. The file is created in the *siebel_srvr_root* directory during installation, and its contents get created as a result of generating triggers.
EXEC	TRUE	Setting this parameter to TRUE causes the SQL script to be run automatically.
Table Owner Password	Your Password	Password for the table owner. This is required for Oracle and IBM DB2.

5. Click the Component Request form applet menu, and then select Submit Request.

6. To view changes to the state, refresh the screen by clicking Run Query from the applet menu.

   Upon completion, the Status field contains either Success or Error.

7. View log details by doing the following:

   a. Click the Screen Enterprise Operation.

   b. In the Show field, select Tasks.

   c. Click the view tab called Task Info Log.

   For more information about administering server component parameters, see *Siebel Server Administration Guide*.

Starting Workflow Monitor Agent

You must start two Workflow Monitor Agent tasks: one for Assignment Manager and the other for Workflow policies.

To create a Workflow Monitor Agent component definition

1. From the application-level menu, choose View > Site Map > Server Administration > Enterprise Configuration.

   The Enterprise Configuration view appears.

2. Click the Component Definitions tab.

   Two Component Definitions lists appear.

3. From the upper Component Definitions list menu, choose New Record.

   A new record appears.

4. Complete the following fields:

Field	Value	Description
Name	NREC’s WorkMon	Name of the component.
Component Type	WorkMon	Worfkow Monitor Agent component type.
Component Group	NREC Workflow Component Group	Select an existing component group.
Alias	NRECWorkMon	Alias for the component. The alias can not contain blank spaces.

5. From the upper Component Definitions list menu, choose Save Record.

   The component definition is saved. To view the definition, you must perform a query.

To set parameters and activate a Workflow Monitor Agent component definition

1. In the upper Component Definitions list, perform a query for the component definition.
2. In the lower Component Definitions list, select the Group Name parameter. Enter the name of the Workflow Policy Group for the requests the component will process.
3. In the lower Component Definitions list, select the Default Tasks parameter. Set the Value to 1.

   This sets the component to start when the Siebel server starts, and stop when the Siebel server shuts down.
4. Optional. You may make additional changes to the component parameters. For a description of Workflow Monitor Agent parameters, see *Siebel Business Process Designer Administration Guide*.
5. From the upper Component Definitions list menu, choose Enable Component Definition.

   The definition state changes from “Creating” to “Active.”
6. Restart the Siebel server.

   Your changes take effect.

Personalization 15

This chapter walks you through the steps for using Siebel Personalization to create a simple business rule that filters NREC's list of houses. The filter will be based on the ZIP Code value of the house record and the user's Postal Code profile attribute. When a partner real estate agent logs in to NREC's application, the agent will only see the houses in the House Detail view that are in the same ZIP Code as is defined on the user's profile.

NOTE: ZIP Code is a field that you added to the Internal Products business component in "Configuring the House Detail View" on page 114.

The tasks in this chapter are all performed using administration views available in the Siebel Web Client. Like Siebel Workflow and Siebel Assignment Manager, Siebel Personalization provides you with another way to customize the business logic of your application without having to work in Siebel Tools and recompile an .srf file.

Creating Rule Sets

Rule sets are collections of the business rules that define how content is displayed. You will create one rule set to contain the business rules for the NREC example.

To create a rule set

1 In a Siebel employee application, navigate to Personalization Administration > Rule Sets.

The Rule Sets view appears.

2 Add a new record to the Rule Sets list using the following values.

Field	Value
Name	NREC Houses Filter
Active	TRUE

Associating Rule Sets with Applets

Rule sets are associated with one or more applets. These are the applets to which the business rules for the rule set apply. NREC is using Personalization to filter the list of houses for sale based on a user's ZIP Code. When users view the list of houses for sale, they will only see the houses for sale in their area. In this case, the applets to which the Personalization rule applies are the Product Form Applet and the Product List Applet.

To associate a rule set with an applet

1. In a Siebel employee application, navigate to Personalization Administration > Applets.

   The Applets view appears.

2. In the Applets list, enter a new record for the Product List Applet.

3. In the Rule Sets list, add a new record for the Rule Set you created as described in "Creating Rule Sets" on page 290.

Field	**Value**	**Description**
Sequence	1	Choose the order in which the rule sets will be evaluated.
Start Date	Leave blank	The date when the conditional expression starts to be evaluated. If Start Date is blank, the conditional expression is evaluated continuously or until the End Date is reached.
End Date	Leave blank	The date after which the conditional expression is not evaluated. If End Date is blank, the conditional expression is evaluated continuously after the Start Date. If both Start and End Date are blank, the conditional expression is always evaluated.

4. Repeat Step 1 through Step 3 to associate the Product Form Applet with the NREC Product Filter.

Creating Rules

Personalization rules control how content is targeted to users. There are three types of business rules: Expression, Business Service, or Invoke Method. For the NREC example, you will write an Expression personalization rule.

Expression rules use Siebel Query Language to set the parameters that control the content displayed to users. In this case, the expression will specify that when displaying records from the Internal Product business component only show those records that have a ZIP Code field that matches the user’s Postal Code attribute. The logic expressed in Siebel Query Language is:

```
[Zip Code] = GetProfileAttr(‘Me.Org.Postal Code’)
```

NOTE: ZIP Code is a field that you added to the Internal Products business component in “Configuring the Internal Product Business Component” on page 121.

For more information about other types of personalization rules and Siebel Query Language, see *Personalization Administration Guide*.

To create a rule

1. In a Siebel employee application, navigate to Personalization Administration > Rule Sets.
2. In the Rule Sets list, select the NREC Houses Filter rule set that you created as described in “Creating Rule Sets” on page 290.

3 In the Rules list, enter a new record using the following values:

Field	Value	Description
Name	NREC Houses Filter	Choose a name that suggests the purpose of the rule.
Sequence	1	Required. Enter numbers in this field to set the order in which the rules should be evaluated.
Rule Type	Expressions	Evaluates inclusion and exclusion expressions directly.
Active	TRUE	Check the box to use the rule.
Include Expression	`[ZIP Code] = GetProfileAttr('Me.Org.Postal Code')`	Used with the Expressions rule type. An expression that sets parameters to include content.

NOTE: You can type the expression directly into the Include Expression field or you can click the Select button to invoke the Expression Designer. The Expression Designer provides you with a drag-and-drop interface for writing complex expressions. For instructions on using the Expression designer to enter the Include Expression for this example, see "Using the Expression Designer" on page 294.

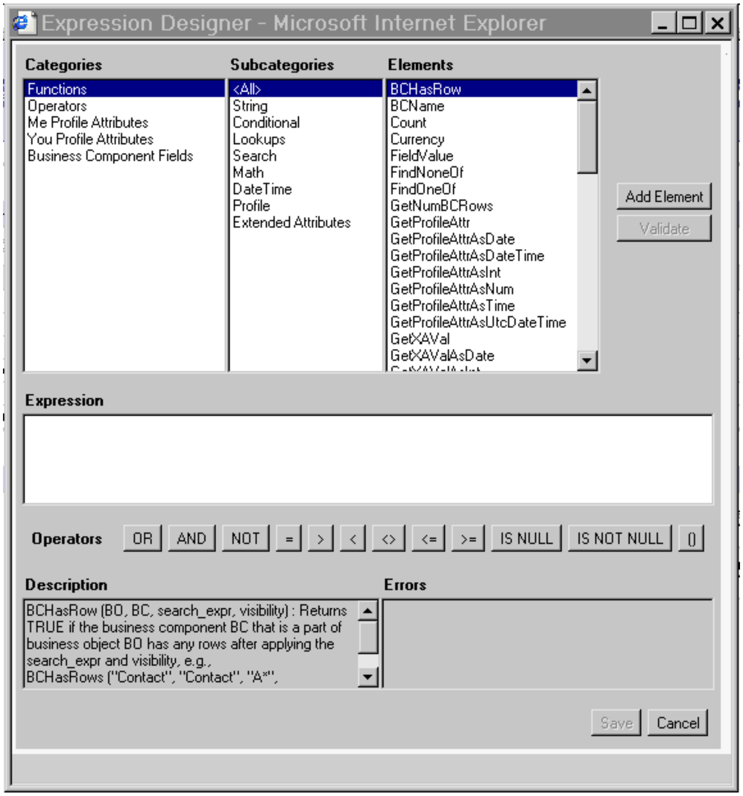
Using the Expression Designer

The Expression Designer allows you to write personalization rules without having to learn Siebel Query Language. It provides you with a drag-and-drop interface and helps you validate syntax of the expression. This procedure walks through the steps for creating the Include Expression defined in Step 3 on page 293. This procedure is an alternative to typing the expression directly into the Include Expression field.

To write a personalization rule using the Personalization Business Rules Designer

1 Click the select button in the Include Expression field.

The Personalization Business Rules Designer appears.



BCHasRow (BO, BC, search_expr, visibility) : Returns TRUE if the business component BC that is a part of business object BO has any rows after applying the search_expr and visibility, e.g., BCHasRows ("Contact", "Contact", "A*",

2. Choose Business Component Fields from Categories, <All> from Subcategories, and Zip Code from Elements, and then click Add Element.

   The Expression box displays `[ZIP Code]`.

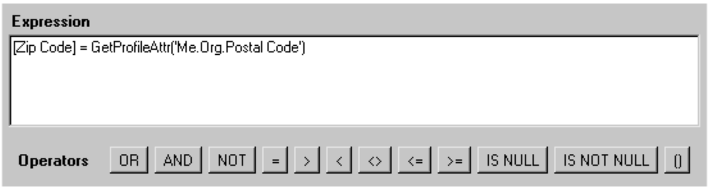
3. Place the cursor after `[ZIP Code]` in the Expression box.
4. Click = in the Quick bar.

   The Expression box displays `[ZIP Code]=`.

5. Place the cursor after =.
6. Choose Me Profile Attributes from Categories, Show as String from Subcategories, Me.Org.Postal.Code from Elements, and then click Add Element.

   The Expression box displays the completed rule:

   ```
   [ZIP Code] = GetProfileAttr(‘Me.Org.Postal Code’)
   ```



Expression

[Zip Code] = GetProfileAttr('Me.Org.Postal Code')

7. Click Save to validate the rule.

   The Expression Designer closes and the expression is displayed in the Include Expression field of the Applet Rules More Info form.

Testing

Personalization rules can be tested in a staging environment before being used in the production environment. This is done in the Test view under Personalization Administration.

NOTE: For the current example, use the House Detail view in your configured application to enter a few house records. Be sure to enter records with several different ZIP Codes. You will define one of these values as a profile attribute when you test. Only records with that ZIP Code should appear. For information about the House Detail View, see "Configuring the House Detail View" on page 114.

To test personalization rules

1. In a Siebel employee application, navigate to Personalization Administration > Test.

   The Test view appears.

2. Enter a Primary User Login and Primary User Password.

   This is the user name and password of the user you want to use to test the personalization rules.

3 For example, to test NREC’s personalization rule, use SADMIN/SADMIN. In the Test Application field, enter the command string for the Mobile Web client.

NOTE: You can right-click your program icon, choose Properties, and then copy and paste the command string from the Target field into the Test Application field. You may have to add the data source using the /d switch.

```
D:\<Siebel_install_dir>\BIN\siebel.exe
/c “D:\<Siebel_install_dir>\bin\ENU\ config_file.cfg”
/d data_source
```

where:

- *<Siebel_install_dir>* is the full path to the client installation directory
- config_file is the application configuration file, for example, uagent.cfg for Siebel Call Center and siebel for Siebel Sales
- data_source is the database to which to connect: Local, Sample, or ServerDataSrc

For example:

```
D:\sea703\client\BIN\siebel.exe
/c "D:\sea703\client\bin\ENU\scw.cfg"
/d Sample
```

NOTE: You can also enter the URL for a Siebel Web Client. For example, http://<*machine_name*>/eChannel. When testing using the Siebel Web Client, the Test Mode dialog box will appear after you complete Step 6 with instructions about how to launch a new instance of the application. For more information, see *Personalization Administration Guide*.

4. Click Load.

   The persistent user profile attributes of the primary user are loaded in the Primary User Attributes list with a `Me.` prefix.

   Person-related attributes have the value Person in the Source field. Organization-related attributes have the value Organization.

5. Select the Me.Org.Postal Code primary user attribute and enter a value.

   For example, enter 94121.

6. Click the Test button.

   A new instance of the specified application opens. For the current example, a new instance of Siebel Partner Portal opens.

7. Navigate to the House Detail view and test the results.

   Only records that have a ZIP Code value of 94121 appear.

8. Click the Save button in the Test view to save the test setup.

Implementing Siebel Remote 16

This chapter begins by explaining how National Real Estate Clearinghouse (NREC) uses Siebel Remote, illustrating the hardware architecture, and outlining the tasks to set up the Siebel Remote server. Next, the chapter outlines and briefly describes the tasks to set up a new Siebel Remote User. Then the chapter describes the process for synchronizing a mobile Web client.

NREC is using Siebel Remote so that its employees can connect to a Siebel Server with their laptop computers and exchange updated data and files. This process is known as *synchronization*. This chapter uses Kate Strong, a real estate agent with NREC, to detail the Siebel Remote scenario. Kate Strong is a mobile user.

Siebel Remote supports mobile computing by allowing field personnel to share current information with virtual teams of other mobile and connected users across the organization.

Typically, the mobile Web client is a laptop computer used by a field sales or service representative. Siebel eBusiness Applications allow a mobile Web client to download a portion of the Siebel database and the Siebel File System to a laptop. This allows users to access their data locally, without being connected directly to the Database Server, Siebel Server, or File System.

As Kate Strong enters and updates information in the local database on her laptop, Siebel Remote Client software tracks the changes as synchronization transactions. Subsequently, when Kate connects to the Siebel Remote server through her dial-up networking connection, these transactions are uploaded from her mobile Web client to the server during synchronization. In a similar manner, transactions occurring on the server that are applicable to Kate will be tracked. During synchronization, these transactions are downloaded from the server to her laptop.

Figure 40 illustrates the main elements of the Siebel Remote architecture.

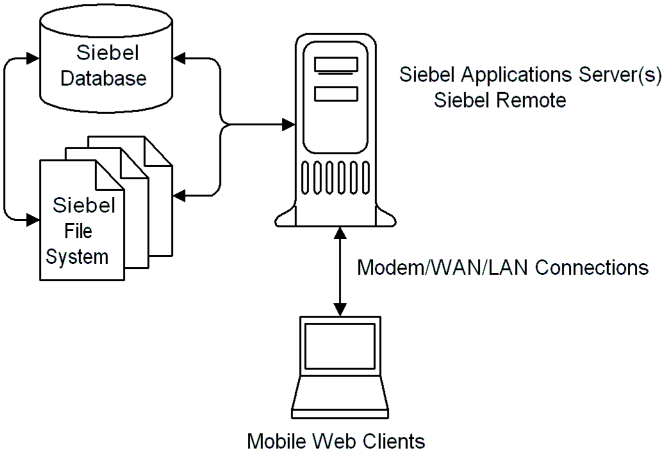


Figure 40. Siebel Remote Hardware Architecture

Setting Up the Siebel Remote Server

The Siebel Remote server runs the Siebel Remote components (such as txnproc, txnroute, and others) and manages synchronization sessions with mobile Web clients. For the NREC implementation, the Siebel Remote server runs on the same machine as the Siebel Server.

The Siebel Remote server provides an interim storage area for data required to synchronize mobile databases with the Siebel database server.

Setting up the Siebel Remote server in preparation for setting up Siebel Remote users includes the following tasks:

- Setting Siebel Remote System Preferences
- Disabling Local Access to All Views
- Starting Siebel Remote Server Components
- Enabling Windows NT Rights on Siebel Remote Server
- Generating a New Database Template

For detailed information on these and other Siebel Remote administration tasks, see the *Siebel Remote and Replication Manager Administration Guide*.

Setting Up a New Siebel Remote User

Setting up a mobile Web client involves certain tasks for both the Siebel Remote server and the mobile Web client. The system administrator must repeat each of these steps for each mobile client.

A system administrator at NREC completed the entire process of setting up each mobile client. NREC decided not to rely on end users to complete the configuration because the system administrator could be more thorough and accurate in completing each task.

Setting up a new Siebel Remote user entails the following tasks:

1. **Create a mobile Web client user account and privileges.** See "Creating a Mobile Web Client User Account and Privileges" on page 303 for details.
2. **Set up mobile Web client hardware and software.** See "Setting Up Mobile Client Hardware and Software" on page 303 for details.
3. **Enable network connectivity.** See "Enabling Network Connectivity" on page 303 for details.
4. **Establish autodial preferences.** See "Establishing Autodial Preferences" on page 303 for details.
5. **Set synchronization preferences.** See "Setting Siebel Remote Preferences" on page 304 for details.
6. **Register a mobile Web client.** See "Registering a Mobile Client" on page 305 for details.
7. **Run database extract for a mobile Web client.** See "Running Database Extract for a Mobile Web Client" on page 307 for details.
8. **Initialize a mobile Web client local database.** See "Initializing a Mobile Web Client Local Database" on page 309 for details.

NOTE: For information about advanced topics, such as authentication or conflict detection and resolution, see *Security Guide for Siebel eBusiness Applications* and *Siebel Remote and Replication Manager Administration Guide.*

Creating a Mobile Web Client User Account and Privileges

NREC chose not to use authentication for mobile Web clients. Therefore, NREC did not need to create any accounts or passwords for these. Because it is the easiest and most popular configuration, it is also the default configuration.

If you want to authenticate mobile Web clients, you need to create accounts for each client depending on the authentication method. For more information about this topic, see *Siebel Remote and Replication Manager Administration Guide*.

Setting Up Mobile Client Hardware and Software

Install the necessary hardware and software on the mobile Web client. This step may include:

- Configuring users.
- Installing disk drives, memory cards, and operating system software.

For more information, see *Siebel Web Client Administration Guide*.

Enabling Network Connectivity

Install the necessary hardware and software on the mobile Web client to allow the mobile Web client to exchange files with the Siebel Remote server. This step may include choosing communication settings and installing networking cards, modems, and software.

For more information about network connectivity, see *Siebel Web Client Administration Guide*.

Establishing Autodial Preferences

A mobile client user using Microsoft dial-up networking can configure Siebel Remote to automatically to establish a connection with the Siebel Remote server when the user initiates a synchronization session.

To establish autodial preferences

1. Choose File > Synchronize.
2. In the Siebel Remote Synchronize dialog box, click Setup.
3. From the Siebel Remote Preferences dialog box, click the Connection pick list and choose the appropriate connection.
4. Click OK, and then Synchronize or close the dialog box.

NOTE: You must first define your phone book entries using Microsoft Dial-Up Networking for Windows 2000, NT, and 98/95 platforms. There is a special situation with the Windows 98/95 setup. When setting up the dialup connection on Windows 98/95, you must establish the connection before using Siebel Remote to synchronize.

Setting Siebel Remote Preferences

NREC is using the default settings for all Siebel Remote parameters established during installation in the client configuration file. Thus, there are no tasks for setting remote preferences.

Setting Synchronization Options

The Siebel Remote client reads configuration parameters in the Siebel configuration file (default is siebel.cfg) to specify the location of the Siebel Server directories, the Siebel File Server directories, and the Siebel Database installation. Before using Siebel Remote, you must set the values for the configuration parameters. The Siebel installation utility creates a siebel.cfg in the client bin directory with default values for each configuration parameter.

When Kate Strong and other NREC mobile users perform synchronization within the application, that is, using File > Synchronize > Database, configuration information is read from the CFG file of that particular application. For example, if Siebel Call Center is used, then configuration information is read from the uagent.cfg.

For more details about synchronization parameters and enabling encryption for synchronization networking, see *Siebel Remote and Replication Manager Administration Guide*.

Registering a Mobile Client

This section describes how to register a mobile Web client. It assumes the Siebel Administrator has previously set up Kate Strong as an employee in the Siebel application.

NOTE: It is important to make sure mobile users have the Client Status view in their responsibilities. This view helps mobile Web clients resolve data merge conflicts on their local databases by showing conflict information after synchronization. See *Applications Administration Guide* for additional detail regarding the setting up of employees.

To register a new mobile Web client on a parent node

1 From the application-level menu, choose View > Site Map > Remote Administration > Mobile Clients.

2 In the Parent Server list, select the appropriate parent node.

If your deployment does not use Replication Manager, then the appropriate node is HQ.

3 In the Mobile Clients list, click the menu button and select New Record.

4 In the New Mobile Client form, enter Kate Strong in the Mobile Client field.

Caution: The mobile Web client name must be entered in uppercase letters and be eight characters or less. It is recommended to use the mobile Web client User ID (see next step) as the mobile Web client name. It *can* contain only Roman, alphanumeric, and the _ or - characters. It *cannot* include spaces, periods, or other invalid characters (/ \ : * ? " < > |) as in the DOS file naming schema. Siebel Remote uses the mobile Web client name to create inbox and outbox directories on the Siebel Server.

5 In the User ID field, click the select button and choose the User ID for Kate Strong and click OK.

The User ID is used to access Kate Strong's local database during initialization and synchronization.

6 In the Routing Model field, click the select button and choose the Sales Representative routing model and click Pick.

NOTE: Each mobile user will be associated with one Routing Model. A Routing Model includes a set of routing rules. Union of the routing rules determines whether a record will be routed to a mobile Web client. For more information on data routing models, see *Siebel Remote and Replication Manager Administration Guide* and the release notes documentation for your application.

7 In the Language(s) field, click the select button, and choose English as the preferred language for Kate Strong.

A language preference allows the mobile user to download data in a preferred language, or languages, for the following dock objects that contain Translation Tables: LOV, Product, Literature, Catalog, Catalog Category. This helps optimize the size of the local databases.

For more information about language preferences, see *Siebel Remote and Replication Manager Administration Guide*

8 Complete the remaining fields as appropriate.

The Sync Password field is used by the Synchronization Manager if the authentication method in the Siebel Server Component Parameters is set to Siebel. Set the password in this field and give it to the mobile user.

The App Server Name field is not populated until the Database Extract is run for the mobile user. At the time the mobile Web client record is created (S_NODE) the APP_SERVER_NAME field is NULL.

NOTE: If you use EIM to load mobile user records, records without a HQ node as the parent node do not appear in the Mobile Clients view. The parent node is stored in the following two columns: EIM_NODE.par_name and EIM_NODE.par_node_type_cd. Although these columns are not required for EIM, they are required for Siebel Remote. When you enter mobile users using the Mobile Clients view, these columns are populated by default.

9 Navigate to Applications Administration > Responsibilities > Responsibilities list, and select the corresponding Responsibility with the Routing Model suffix.

   This is required because Kate Strong's data routing model, Sales Representative, is one of the following:

   - Sales Representative
   - Minimal Data
   - Analyst
   - Sales Manager

   The corresponding Responsibility with the Routing Model suffix relates to the data routing model assigned in Step 6 above. For more information about corresponding routing models and how these help optimize the size of local databases, see *Siebel Remote and Replication Manager Administration Guide*.

   a In the Users list, add a new record.

   b In the Add Users Selection dialog box, select the mobile user and click OK.

Running Database Extract for a Mobile Web Client

The database extract process retrieves data visible to Kate Strong from the server database. It retrieves data according to routing rules that define the level of access to information for her. It creates compressed files that contain data to be loaded into her local database when she initializes her laptop.

Before running a database extract for Kate Strong, you must make sure that your organization's reporting hierarchies are updated. Use the Position Administration view from the Application Administration screen to verify that she has a valid position in your organization's hierarchy. The resulting information is used by the system's routing rules, and may affect the outcome of the database extract. For more information on positions, see *Applications Administration Guide*.

To run a database extract for a mobile Web client

1 From the application-level menu, choose View > Site Map > Server Administration > Enterprise Operations.

2 Click the Component Requests tab, and click New.

3 In the Component/Job field, select Database Extract from the pick list.

4 In the Component Request Parameters list, click New and add the necessary parameters.

The required parameter for Database Extract is Client Name.

The value for the Client Name parameter is KSTRONG, the name of Kate Strong's mobile Web client.

5 From the Component Requests menu, select Submit request.

The mobile client database is extracted. This may take a few minutes.

Server Directory Tree After Running Database Extract

The Database Extract program creates the appropriate directory and its subdirectories for Kate Strong on the Remote server.

NOTE: The installation program also places a directory named txnproc in the docking subdirectory within the Siebel server root directory. Do *not* modify the contents of this directory under any circumstances.

The following example shows a portion of the server directory tree after you run Database Extract for Kate Strong:

```
siebel
   docking
      kstrong
         inbox
         outbox
      txnproc
```

For more information about this topic, see "Sample Directory Tree After Running Database Extract" on page 85.

Initializing a Mobile Web Client Local Database

The volume of information that must be downloaded from the Siebel Remote server to initialize a mobile Web client's database is usually substantial. You should establish a LAN (rather than a modem or WAN) connection between the server and the mobile Web client for this process.

Alternatively, the local database can be initialized from a CD-ROM or other media—if compressed files have been copied into the folder specified as FileSystem parameter. For more information about extracting to a CD, see *Siebel Remote and Replication Manager Administration Guide*.

NOTE: To initialize a mobile Web client database, the TableOwner parameter in the CFG file must be set to Siebel (the default).

To initialize the mobile Web client database

1. Establish a connection between the Siebel Remote server and Kate Strong's laptop.
2. In the mobile Web client's Siebel program group, click the Siebel Remote icon.

   NOTE: Verify that the icon is pointing to the appropriate CFG file. The default is siebel.cfg.

3. In the Siebel Remote Parameters dialog box, enter the following information in the appropriate fields:

Field	Entry
Client Name	KSTRONG (registered Siebel client name).
User Name	KSTRONG (login name).
Password	KSTRONG (If an authenticator password was assigned, enter that instead of KSTRONG).

4. Click Continue.
5. Monitor the process for errors.

To initialize the mobile client database during login

- Log in to the local database when starting the application. When Siebel eBusiness Applications cannot find a local database, it will attempt to initialize the local database. Follow the prompts.

After the initialization completes, Kate Strong's laptop is ready for use in the field. Kate Strong will need to resynchronize with the Siebel Remote Server on a frequent basis. Instructions for this process are included in the next section, "Synchronizing a Mobile Web Client" on page 311. There is an auto synchronization option to help maintain the frequency of synchronization.

Synchronizing a Mobile Web Client

This section describes the processes for synchronizing a mobile Web client.

Routing and Merging

On the Siebel Remote server, the Transaction Router and Transaction Merger components continuously route and apply transactions for mobile clients. Figure 41 illustrates the processes that occur when a mobile client is synchronized.

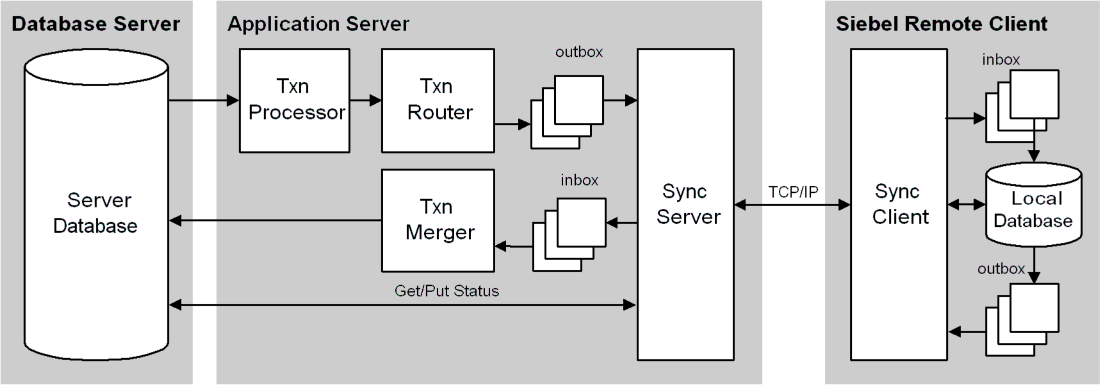


Figure 41. Synchronizing a Mobile Web Client (Routing and Merging)

For more detailed information about routing and merging, see *Siebel Remote and Replication Manager Administration Guide*.

Synchronization Session

To synchronize an existing mobile Web client, the mobile user launches Siebel Remote, either from within the Siebel mobile Web client or in stand-alone mode. Siebel Remote executes the following steps:

- Connects
- Validates mobile Web client
- Check for 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
- Cleans up

For more detailed information about synchronization steps, see *Siebel Remote and Replication Manager Administration Guide*.

How Changes Are Propagated to and from a Mobile Web Client

Between synchronization sessions, the Siebel Remote server prepares transactions applied to the database server by both mobile and dedicated users. Siebel Server components write the transactions to a separate directory for each mobile user, such as Kate Strong. These transactions, combined with items from the File System, are downloaded to Kate during her next synchronization session. Items from the File System include updated, published, or requested marketing literature, correspondence templates, and other types of file attachments.

A similar process occurs on Kate's laptop as well, although without the server component.

Process Flow for Changes by Connected Users

This section describes the process flow for downloading changes on the server database to local databases, such as Kate's. The flow begins from the time a connected user creates a new opportunity until it appears in Kate's local database.

This specific process flow includes an example of a telesales representative in a Call Center. The telesales person talks to potential customers responding to a new NREC advertising campaign.

The telesales representative decides to create a new opportunity record for one of the more promising responses. The figure below illustrates this flow and includes the following groups of steps:

1. The telesales person creates a new opportunity record—a transaction saved in the opportunities table on the server database. A copy is saved to the master transaction log.
2. A mobile user invokes a synchronization session from the laptop.

   The mobile user can use the Siebel client while the Siebel Remote client applies the changes to the local database, as shown in Figure 42.

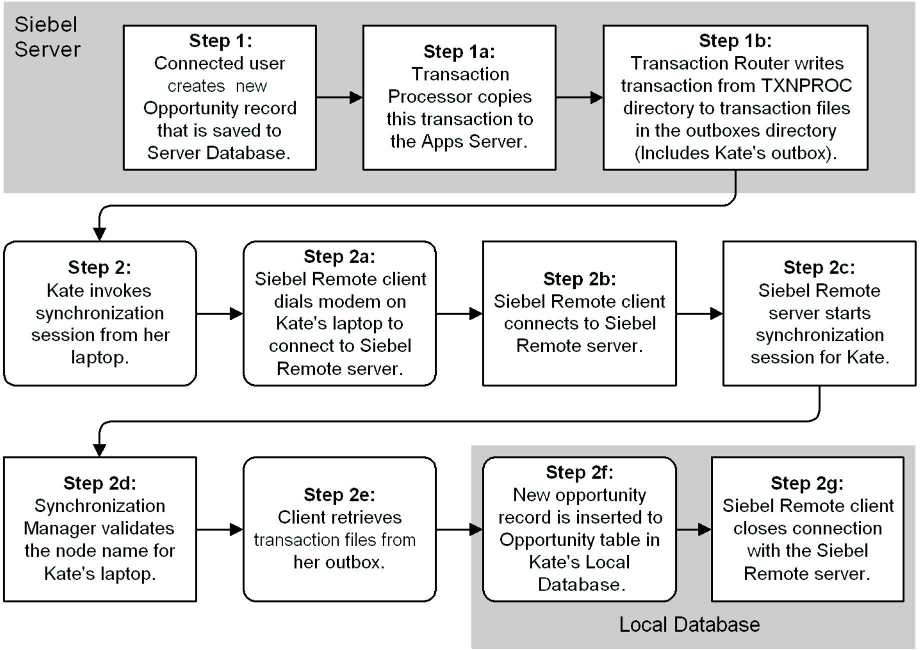

Siebel Server
Step 1: Connected user creates new Opportunity record that is saved to Server Database.
Step 1a: Transaction Processor copies this transaction to the Apps Server.
Step 1b: Transaction Router writes transaction from TXNPROC directory to transaction files in the outboxes directory (Includes Kate's outbox).
Step 2: Kate invokes synchronization session from her laptop.
Step 2a: Siebel Remote client dials modem on Kate's laptop to connect to Siebel Remote server.
Step 2b: Siebel Remote client connects to Siebel Remote server.
Step 2c: Siebel Remote server starts synchronization session for Kate.
Step 2d: Synchronization Manager validates the node name for Kate's laptop.
Step 2e: Client retrieves transaction files from her outbox.
Step 2f: New opportunity record is inserted to Opportunity table in Kate's Local Database.
Step 2g: Siebel Remote client closes connection with the Siebel Remote server.
Local Database

Figure 42. Process Flow for Changes by Connected Users

Caution: Users should never directly modify the local transaction log. The Siebel Remote synchronization client automatically purges the local transaction log table.

For more information about process flow for changes by connected users, see *Siebel Remote and Replication Manager Administration Guide*.

Process Flow for Changes Made by Mobile Users

Mobile Web clients, such as Kate, use a local database to store data for user access. The local database contains Siebel eBusiness Applications tables that store user data. The local database also contains a local transaction log to store transactions created by the mobile user. Siebel Remote forwards these transactions to the Siebel Remote server when the client synchronizes.

This section provides a description of each phase of the process flow, from the time when Kate modifies the new opportunity until the time when the modifications appear in the Server database. For this example, assume that Kate is meeting with the potential new client, represented by the opportunity record entered by the telesales representative in the previous section.

The figure below illustrates this flow and includes the following groups of steps:

1. As a result of the meeting, Kate makes changes to the Opportunity record in her local database. She enters these changes immediately after the meeting while working offline.
2. The modified opportunity record is saved to the Opportunities table in Kate’s local database.
3. The transaction record is saved to the Local Transaction log.

4 Kate synchronizes the laptop. The Siebel Remote client extracts pending transactions from the Local Transaction log into transaction files (.dx). The client then places these DX files in the outbox directory on Kate's laptop. Siebel Remote connects to the server and the DX file is sent to the server and inserted in the database. as shown in Figure 43.

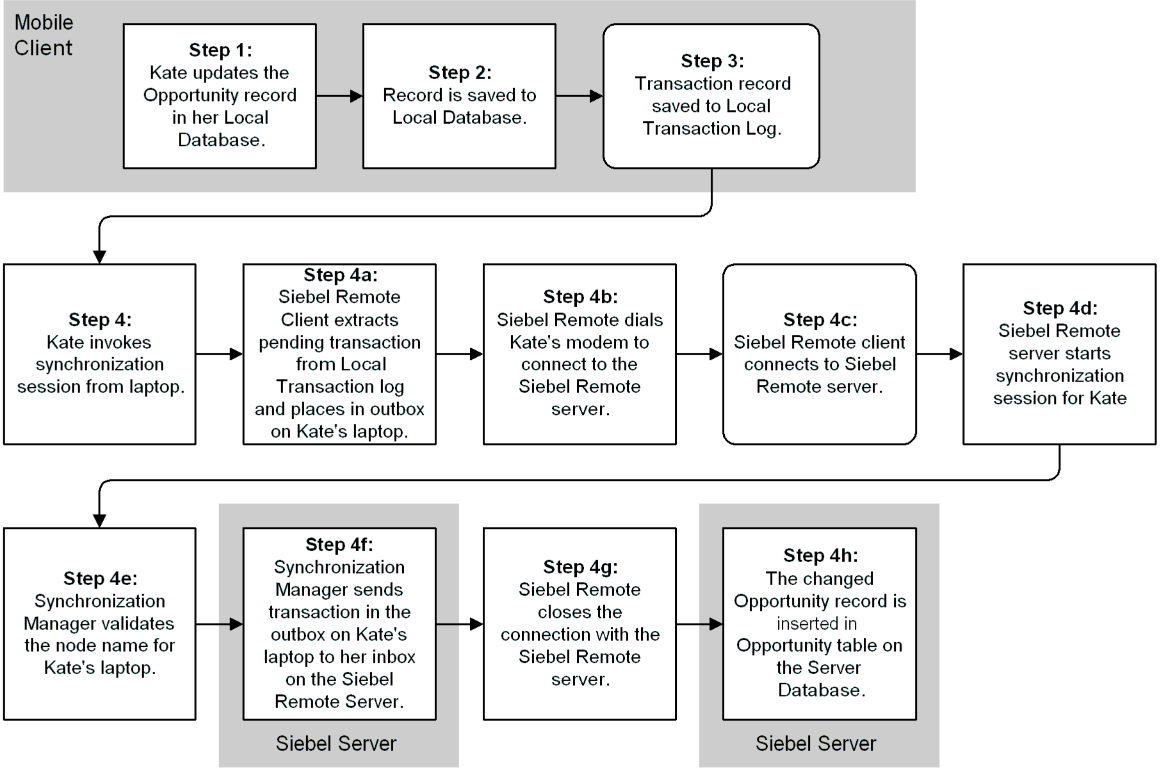

Mobile Client
Step 1: Kate updates the Opportunity record in her Local Database.
Step 2: Record is saved to Local Database.
Step 3: Transaction record saved to Local Transaction Log.
Step 4: Kate invokes synchronization session from laptop.
Step 4a: Siebel Remote Client extracts pending transaction from Local Transaction log and places in outbox on Kate's laptop.
Step 4b: Siebel Remote dials Kate's modem to connect to the Siebel Remote server.
Step 4c: Siebel Remote client connects to Siebel Remote server.
Step 4d: Siebel Remote server starts synchronization session for Kate
Step 4e: Synchronization Manager validates the node name for Kate's laptop.
Step 4f: Synchronization Manager sends transaction in the outbox on Kate's laptop to her inbox on the Siebel Remote Server.
Siebel Server
Step 4g: Siebel Remote closes the connection with the Siebel Remote server.
Step 4h: The changed Opportunity record is inserted in Opportunity table on the Server Database.
Siebel Server

Figure 43. Process Flow for Changes by Mobile Users

For more information about process flow for changes by mobile users, see *Siebel Remote and Replication Manager Administration Guide*.

Synchronizing a Mobile Web Client Machine

Kate must synchronize frequently to obtain and view possible updates in the server database. Also, there may be updates to the store of documentation, marketing literature, and sales brochures in the file system.

Siebel 7.5 includes an auto synchronization option to help maintain the frequency of synchronization. Frequent synchronization by mobile users can improve the performance of the system. After initializing their local database, mobile users enable or disable auto-synchronization from User Preferences > DB Synchronization. For more information see, *Siebel Remote and Replication Manager Administration Guide*.

To synchronize a mobile Web client

1. Start a Siebel application on the Mobile Web client machine.
2. Select File > Synchronize Database.
3. Choose the synchronization options.

   See *Siebel Web Client Administration Guide* for options.
4. Click Synchronize.

Deploying the Application 17

This chapter covers the steps that NREC took to migrate its data and application from the test to production environments. Many of the tasks in this section were covered in previous chapters.

Migrating Data from the Test Environment to Production

After completing your testing, you are ready to move data from the Test environment to the Production environment.

The data you must move includes the following:

- **Tools configurations, including schema changes.** Use the Repository Migration Utility to move the latest Tools configuration from your Test environment to your Production environment. The procedure is the same as the one you followed in moving configurations from Development to Test. Refer to the instructions in Chapter 10, "Migrating Repository Data to the Test Environment."

- **Modified files, such as .srf file, Web templates, image files, and cascading style sheets.** You must copy any changes you made to Web templates and related files, as described in Chapter 10, "Migrating Repository Data to the Test Environment."

- **Transactional data, such as accounts, contacts, opportunities, and so on.** You have a snapshot of this data in your Test environment, after having completed an EIM import. However, this snapshot may by now be out-of-date, as updates may have continued to be made to the data in your legacy system. Therefore, rather than copying this user data from Test to Production, do another EIM import from your legacy system—but this time directly into your Production environment. For more information about EIM imports, see Chapter 11, "Using EIM to Load Data into the Test Environment" and *Siebel Enterprise Integration Manager Administration Guide*.

- **Setup data, such as employees, positions, responsibilities, and so on.** You must copy this information from either your Test environment or your legacy system to your Production environment.

- **Program data, specifically Assignment Manager rules and Workflow processes and policies, and Personalization rules and expressions.** You must copy this information from your Test environment to your Production environment.

This chapter gives instructions for the last two items—migrating setup data and program data from Test to Production.

Moving Setup Data

You must move setup data into your Production environment. Setup data includes information about employees, positions, responsibilities, and so on. Because the setup data in your Test environment may be obsolete by the time you are ready to move into Production, it is a good idea to move the data into the Production server directly from your legacy system, using EIM for the import. For example, use the EIM_EMPLOYEE interface table to import employee data from your legacy system into the Production server.

For information about using EIM, see *Siebel Enterprise Integration Manager Administration Guide*. To determine which interface tables to use, see *Siebel Interface Tables Reference*.

Moving Program Data

You must move program data, specifically Assignment Manager, Workflow and Personalization data from the Test environment (where you created it) to the Production environment, as described in this section.

Moving Assignment Manager Data

You can use EIM to move all your Assignment Manager data.

To move Assignment Manager data

- Use EIM, using the following interface tables:
  - EIM_ASGN_GRP
  - EIM_ASGN_WL
  - EIM_ASG_GRPDTL

For more information about EIM, see the *Siebel Enterprise Integration Manager Administration Guide*.

Moving Workflow Data

Moving Workflow data consists of moving data about workflow processes and moving data about workflow policies and groups. Moving each of the two types of data requires you to follow a different procedure.

To export a process

1. Log on to the application in the test environment.
2. Navigate to Siebel Workflow Administration > Workflow Processes > All Processes.
3. Select the process or processes you want to export. To select more than one process, press and hold the CTRL key while selecting the processes.
4. From the Processes list menu, select Export Workflow.

   The XML workflow process definition appears.
5. From the XML dialog box menu, select File > Save As.
6. Enter the file path, file name and the .xml file name extension, and then click Save.

   The process or processes are exported. If you selected more than one process to export, all the processes are saved to one XML file.

NOTE: When exporting a process containing sub processes, you must also export the subprocesses. Subprocesses are not exported automatically.

To import a process definition

1. Log on to the application in the production environment.
2. Navigate to Siebel Workflow Administration > Workflow Processes > All Processes.
3. From the applet menu, select Import Workflow.
4. Select a path and file name of the process to import.
5. Click Open.

   The process is imported with a status of In Progress.

NOTE: If a process definition of the same name exists in the target environment, the newly imported process definition’s version number will be incremenated by one.

To move workflow policies and groups

- Use EIM, using the following interface tables:
  - EIM_WFM_GROUP
  - EIM_WFM_RULE
  - EIM_WFM_ACTION

For more information about EIM, see *Siebel Enterprise Integration Manager Administration Guide*.

Moving Personalization Data

Personalization rules, events, and actions can be exported as an XML file for later importation into another Siebel environment.

Exporting Personalization Data as an XML File

All personalization data rules, events, and actions are exported in one XML file.

To export personalization data as an XML file

1. Navigate to any Personalization Administration view, for example, Views.
2. Click the menu button, and then choose XML Export.

   The File Download window appears.
3. Select the Save this file to disk radio button, and then click OK.

   A dialog box appears prompting you specify a file name and storage location.

   The default file name is personalization.ooo.xml.
4. Save the XML file.

Importing Personalization Data

Personalization rules, events, and actions can be imported from an XML file generated by a previous export. This process can take several minutes.

To import personalization data from an XML file

1. Navigate to any Personalization Administration view, for example, Views.
2. Click the menu button, and then choose XML Import.

   A dialog box appears.
3. Click Browse.

   A dialog box appears prompting you for a file name.
4. Choose a file, and then click OK.
5. Click Submit.

   The personalization data is imported. When the process is finished, a message displays how many records had conflicts and were inserted, updated, and skipped.

Rolling Out to End Users

As described in "Migrating Data from the Test Environment to Production" on page 318, migrating data provides the production server with the latest data. The next logical step in the deployment is to roll out the applications to users. This section follows the NREC example to show how NREC rolled out the application to end users.

Rolling Out to Siebel Web Client Users

With the exception of the Web browser, the applications for Web Client users reside on the server. Therefore, as long as all the client machines have a Web browser, you need only tell users the URLs to use to log on to for each application.

Rolling Out to Mobile and Dedicated Web Client Users

The Packager Utility allows you to create custom software installation packages for distribution to end users. This installer contains the Siebel Mobile and Dedicated Client executables and your custom configuration. Using the installer allows end users to create a Siebel environment on their machines that duplicates the environment on the administrator's client machine.

After the Packager Utility assembles the software into a single, self-extracting file, you can distribute this installer to your users. For example, NREC chose network distribution, as described in "Making Your Customized Installer Available to End Users" on page 329. However, Siebel Packager packages can also be distributed in the following ways:

- **Siebel Anywhere.** You can use Siebel Anywhere to distribute and execute the installation automatically for both dedicated and mobile users. However, because receiving Siebel Anywhere kits requires the Siebel client to be already installed on the user's machine, you cannot use Siebel Anywhere for an initial rollout. You can use it only for upgrades. For more information, see *Siebel Anywhere Administration Guide*.

- **CD-ROM.** You can copy the customized software package onto CD-ROMs for distribution to end users.

- **Other methods.** You can distribute the program through any other file distribution mechanism, such as email or FTP.

Preparing to Use Siebel Packager

You must complete the procedures in this section before using the Packager Utility.

To prepare to use the Packager Utility

1 Make sure the Packager Utility has been installed on a client machine in your Production environment. This initial installation will serve as a model for other installations, which may be performed by running Packager against the initial installation.

- Select Custom during client installation and be sure to select the Packager Utility option.
- The rest of this chapter refers to the root-level directory of the client installation as *`siebel_clnt_root`*.
- The Packager will use the files from this client installation (or another client installation, as specified when running the Packager) for the installer that it creates.

NOTE: Be sure to customize this model Siebel client installation so that it is identical to how you intend to package it. When creating the custom installer, the Packager Utility will reproduce this model installation exactly.

2 Copy any changed files to the appropriate subdirectories under *`siebel_clnt_root`*, or under the root directory of another installation that you will use to create the custom installation package.

Such files may include custom repository (`.srf`) or configuration (`.cfg`) files, and Web template files.

3 Make sure that you have sufficient free disk space on the client on which the Packager Utility is installed.

During the packaging process, the Packager Utility temporarily requires three times the amount of disk space required by the Siebel client software you are packaging, plus two times the disk space required by the third-party software you are packaging.

4 Create a custom siebel.ini file.

The siebel.ini file controls the following behavior of the installer that the Packager will create:

- Whether or not the Siebel client installation program checks that the appropriate versions of third-party software have been installed on the client machine
- Whether or not the data sources that the Siebel application relies on are installed and how they are configured
- Which installation dialog boxes users are presented at run time and which installation parameters they can specify

The siebel.ini file used by the standard Siebel client installer prompts the user for parameters—not usually the desired behavior for an end-user installation. It is highly recommended that you create a siebel.ini file customized to your environment.

Make your changes to the siebel.ini file located in the \siebel_client\packager\temp\package_name directory using a standard text editor. This file is largely self-documented to help you in customizing it. If you need more information, see *Siebel Web Client Administration Guide*.

Preparing Siebel Components for Packaging

You can use the Packager Utility to package the Siebel client installation software. Of the different installable components that make up a Siebel application, NREC chose to package the ones described in Table 16.

Table 16. Software Installed in the Test Environment

Component	Description
BIN	Siebel Executable Files (Binaries) located in the `\siebel_clnt_root\bin` directory, including the required .dll files, configuration files, and the Siebel executable. You should include all files in this directory in your self-extracting installer, except the user preferences file, user.prf, and the session file, siebel.ses. You may want to replace the siebel.cfg file with your customized configuration file.
FONTS	Contains font files, located in the `siebel_clnt_root\fonts` directory.
LOCAL	Location of the local database, located in the `siebel_clnt_root\local` directory. Local databases are unique to individual users and should not be packaged.
LOCALE	Language-specific files, located in the `siebel_clnt_root\locale`. directory. Do not omit this module when creating a package.
MSGTEMPL	Message files used by the client, located in the `siebel_clnt_root\msgtempl` directory.
Objects	Object Configuration Template Files (Configured Objects) located in the `siebel_clnt_root\objects` directory— the precompiled *.srf file to distribute to end users. The \objects directory must contain at least one .srf file before you start the Packager Utility.
PUBLIC	Contains HTML, JavaScript and image files for Siebel Web Client, located in the `siebel_clnt_root\public` directory.
SQLTEMPL	Contains SQL scripts used by the Siebel Web Client, located in the `siebel_clnt_root\sqltempl` directory.
WEBTEMPL	Contains Web templates, located in the `siebel_clnt_root\webtempl directory`.

For information about other types of files that can be packaged, see *Siebel Web Client Administration Guide*.

Running Siebel Packager

This section describes how to run Siebel Packager. The Siebel Packager wizard walks you through four windows to help you create the custom client installer.

NREC is deploying a single-language version of their application. This requires two separate installers—one for BASE and one for ENU (U.S. English). Both packages must use the same package name. You need a separate installer for each language component, including BASE. For example, if NREC were planning a multilingual roll out, it would require an installation of BASE as well as each additional language pack.

To create the BASE package

1. From the Windows Start menu, select Programs > Siebel Client 7.0 > Packager.
2. Click OK at the Choose Setup Language screen.

   The Siebel Client Packager, Step 1 of 4 - Directory Definition dialog box appears.
3. Create the BASE package:
   - **a** Enter a Package Name. For example, NREC Package.
   - **b** Leave the default values in the remaining fields.
   - **c** Click Next.
4. In the Module Definition dialog box, select Packager from the Modules list and click Remove and then Click Next.

   You do not want to include Packager itself as a module.
5. In the Packaging window, click Start.

   The utility displays progress information while it executes.
6. When the utility is complete, click OK and then Next.
7. In the Self-extracting Archive dialog box, click Exit to finish the packager.

To create the language package

1. Go to Start > Programs > Siebel Client 7.0 > Packager, and click OK at the Choose Setup Language dialog box.
2. In the Directory Definition screen:

   a. Choose <LANGUAGE> in the Language Packs list (for example, choose ENU for U.S. English).

   b. Enter a Package Name (for example, NREC Package).

   c. Leave the default values in the remaining fields.

   d. Click Next.

   The Module Definition dialog box appears.
3. Choose Packager from the Modules list, click on Remove to remove it, and then click Next.

   The Packaging dialog box appears.
4. Click Start to begin creating the package.
5. When the Packager is complete, click OK, and then click Next to continue.
6. In the Self-extracting Archive dialog box, click Exit to finish the packager.

NOTE: The parameters used in the siebel.ini file need to be modified so that the installation on the end-user machines is silent. The siebel.ini file can be edited by clicking on the Edit siebel.ini file in the Packager. For more information about creating Base and Language Packages, and about editing the siebel.ini file, see *Siebel Web Client Administration Guide*.

Making Your Customized Installer Available to End Users

After you have tested your customizations and are satisfied with the client installer you have created, make your customized installer available to end users. As described earlier, you can distribute your customized installation program to end users in a number of ways. Because NREC chose to distribute the installer using network distribution, this section describes that distribution method. For instructions about other distribution methods, see *Siebel Web Client Administration Guide*.

To distribute a self-extracting installer over a LAN (if you created a self-extracting archive)

1. Put the self-extracting installer (packager_name.exe) in a network-accessible directory. Make sure that all users have access to this directory.
2. Send an email to users explaining how to copy and extract the package from this location. Consider telling users how to ftp the self-extracting archive on their own machines and install it from there.

To distribute an installer over a LAN (if you did not create a self-extracting archive)

1. Put the package directory in a network-accessible location to which all users have access.
2. Send an email to users to tell them how to install the package from this location.

After or during this rollout, you are ready to provide user training on the applications.

Index