



Siebel Composer Developer Preview Guide

Innovation Pack 2016, Rev. A
July 2016

ORACLE®

Copyright © 2005, 2016 Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

Documentation Accessibility

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

Access to Oracle Support

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

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Siebel Composer

About Siebel Composer	7
Benefits of Siebel Composer	8
About the User Interface for Siebel Composer	9
About Compiling and Publishing	10
Siebel Tools Features Included in Siebel Composer	10
Siebel Tools Objects Included in Siebel Composer	11
Siebel Tools Features Excluded From Siebel Composer	11
Siebel Tools Objects Excluded From Siebel Composer	12
Aspects of Responsive Web Design in Siebel Composer	12
About Designating Responsive Web Design for Siebel Business Applications	13
Added Fields in Siebel Composer	14
Added Views in Siebel Composer	17
Planned Enhancements to Siebel Composer	17

Chapter 3: Setting up Siebel Composer

About Setting Up Siebel Composer	19
Process of Setting Up Siebel Composer	20
About the Web Template Migration Process	21
Prerequisite for Web Template Migration Process	22
HTML Tags Migrated to Siebel Composer	22
SWE Tags Migrated to Siebel Composer	23
Manually Running the Migration Script	26
Verifying the Results of Web Template Migration Process	27
Publishing Changes to Siebel Web Templates	27
About the Application Object Manager Component for Siebel Composer	27
Adding the Application Object Manager Component for Siebel Composer	28

- Editing the new_comp_sia.dat File 28
- Running the new_compdef_sia Script 30
- Adding the Virtual Directory for Siebel Composer 31
 - Editing the new_virdirs Script 32
 - Running the new_virdirs Script 33
- Configuring Siebel Composer 34
 - Changing the Timing of Automatic Selection of Changed Field 34
 - Setting the Feature for Custom Symbolic Strings 34
- Enabling the Composer Mode for the Selected and Opened Workspace to Edit Objects 35
 - Composer Toolbar 36
 - Hover Context 37
 - Context Panel 38
 - Execution Bar 39
- Navigating to the Workspace Dashboard in the Siebel Composer Mode Application 39
- Setting up Composer Application Edit 42
 - IIS Reverse Proxy Configuration 43

Chapter 4: Use Cases for Siebel Composer

- Changing Logos in Application Banners 47
- Changing Applet Fields to Read-Only Fields 48
- Changing Labels for Applet Fields 49
- Making Applets Visible in Specific Screen Resolutions 50
- Making Applets Visible to Specific Users 51
- Repositioning Elements of Siebel Composer 52
- Resizing Elements of Siebel Composer 53
- Modifying the List Applet Columns 54
- Publishing Modifications in Siebel Composer 55

Index

1

What's New in This Release

What's New in Oracle's Siebel Composer Developer Preview Guide, Innovation Pack 2016, Rev. A

Table 1 lists the changes in this revision of the documentation to support Siebel Innovation Pack 2016. The new features listed in this table are available only if you install and deploy Innovation Pack 2016 or later.

NOTE: Siebel Innovation Pack 2016 is a continuation of the Siebel 8.1/8.2 release.

Table 1. New Product Features in Siebel Composer Developer Preview Guide, Innovation Pack 2016, Rev. A

Topic	Description
"Enabling the Composer Mode for the Selected and Opened Workspace to Edit Objects" on page 35	Modified topic. References to <i>Enabling the Workspace Feature in the Composer Mode</i> are replaced with references to <i>Navigating to the Workspace Dashboard in the Siebel Composer Mode Application</i> .
"Navigating to the Workspace Dashboard in the Siebel Composer Mode Application" on page 39	Modified topic. It includes the following changes: <ul style="list-style-type: none">■ The topic heading is changed from <i>Enabling the Workspace Feature in the Composer Mode</i> to <i>Navigating to the Workspace Dashboard in the Siebel Composer Mode Application</i>.■ A modified section that discusses how to access and open the Workspace Dashboard view in the Composer mode.■ A new section that discusses how to enable the Workspace feature in the Composer mode.

What's New in Oracle's Siebel Composer Developer Preview Guide, Innovation Pack 2016

Table 2 lists the changes in this revision of the documentation to support Siebel Innovation Pack 2016. The new features listed in this table are available only if you install and deploy Innovation Pack 2016 or later.

NOTE: Siebel Innovation Pack 2016 is a continuation of the Siebel 8.1/8.2 release.

Table 2. New Product Features in Siebel Composer Developer Preview Guide, Innovation Pack 2016

Topic	Description
Enabling the Composer Mode for Workspaces in Siebel Tools on page 35	New topic. It includes information on how to enable the Composer mode for Siebel workspaces and information about the main sections of Siebel Composer.
Enabling the Workspace Feature in the Composer Mode	New topic. It includes information on how to enable the Workspace feature in the Composer mode.
"Setting up Composer Application Edit" on page 42	New topic. It includes information on how to set up Siebel Composer to work with Siebel REST.
"Repositioning Elements of Siebel Composer" on page 52	New topic. It includes information on how to reposition the elements of Siebel Composer.
"Resizing Elements of Siebel Composer" on page 53	New topic. It includes information on how to resize the elements of Siebel Composer.
"Modifying the List Applet Columns" on page 54	New topic. It includes information on how to add and delete the list applet columns.

2

Overview of Siebel Composer

This chapter provides an overview of Siebel Composer. It includes the following topics:

- ["About Siebel Composer"](#)
- ["Benefits of Siebel Composer"](#)
- ["About the User Interface for Siebel Composer"](#)
- ["About Compiling and Publishing"](#)
- ["Siebel Tools Features Included in Siebel Composer"](#)
- ["Siebel Tools Objects Included in Siebel Composer"](#)
- ["Siebel Tools Features Excluded From Siebel Composer"](#)
- ["Siebel Tools Objects Excluded From Siebel Composer"](#)
- ["Aspects of Responsive Web Design in Siebel Composer"](#)
- ["About Designating Responsive Web Design for Siebel Business Applications"](#)
- ["Added Fields in Siebel Composer"](#)
- ["Added Views in Siebel Composer"](#)
- ["Planned Enhancements to Siebel Composer"](#)

About Siebel Composer

Oracle is gradually migrating from Siebel Tools to Siebel Composer to simplify and expedite the process of configuring Siebel Business Applications. In Siebel Composer, you currently have access to some of the functionality in Siebel Tools, so you can use Siebel Composer to perform some of the tasks that in the past you performed in Siebel Tools. Developers can review this guide to understand currently available functionality in this developing product.

Siebel Composer is a Web-enabled application that runs on a server, and you access it in a browser by using a Siebel Open UI client. Siebel Tools is a stand-alone application that runs on Microsoft Windows.

In Siebel Composer, you can add fields to business components or applets, and you can designate objects as inactive or active. For more information about how you can use Siebel Composer, see [Chapter 4, "Use Cases for Siebel Composer"](#).

Continue to use Siebel Tools to configure your Siebel application in your production environment.

Using Siebel Composer, note that:

- Do not implement Siebel Composer in a production environment because this preliminary release of the software is not ready for implementation. Experiment with Siebel Composer in an isolated development environment that will never go to actual production. Do not implement Siebel Composer in the Siebel Enterprise (logical grouping of server components, with one enterprise database and one Siebel Gateway Name Server) for a production environment.
- Ensure that the EAI (Enterprise Application Integration), REST (Representational State Transfer) Proxy Object Manager, and Workspace features are enabled on the environment before you work on Siebel Composer.

For more information about how to setup Siebel Composer Application Edit to work with Siebel REST, see [“Setting up Composer Application Edit”](#). For more information about how to use the Workspace feature in Siebel Tools, see the *Using Siebel Tools* guide.

- Contact business components are the base business components and you cannot deactivate these base components in Siebel Composer. If the Contact business components are inactivated, you won't be able to open Siebel Composer; in this case, you must restore the database to a previous working version before you can open Siebel Composer.
- Siebel Composer does not support the client-base code; therefore, Siebel Composer **does** work with the server and **does not** work with the sample database.
- Siebel Composer has the same client requirements as other Siebel Open UI clients. For more information, see *Deploying Siebel Open UI*.

NOTE: The *Siebel Bookshelf* is available on Oracle Technology Network (<http://www.oracle.com/technetwork/documentation/siebel-087898.html>) and Oracle Software Delivery Cloud. It might also be installed locally on your intranet or on a network location.

Benefits of Siebel Composer

Siebel Composer provides the following benefits:

- More elements for composition of the user interface.

For more information, see the description of the Type field in [“Added Fields in Siebel Composer” on page 14](#).

- More efficient rendering of Web pages.

The DOM (Document Object Model) structure in Siebel Business Applications is simplified, and the DOM is assembled using a single HTTP response, not multiple relays between the client and the server. For more information, see [“Aspects of Responsive Web Design in Siebel Composer” on page 12](#).

When you use Siebel Tools, content is trimmed only from the client. Also, post-response JavaScript execution trims content. When you use Siebel Composer, the server evaluates client capabilities, such as resolution and form factors. Content is tailored from the server, and more efficient options are available to manage content.

- Expanded options for generating content.

You can specify standard Siebel expressions for additional elements in the user interface, such as Web templates, Web pages, and views. For more information, see the description of the Expression field in [“Added Fields in Siebel Composer” on page 14](#).

- Expanded options for styling content.

You can designate classes for the CSS (Cascading Style Sheet) in a DOM with valid IDs to configure the styling of a Siebel application. A DOM without valid IDs is removed from the server. For more information, see the description of the HTML Attributes field in [“Added Fields in Siebel Composer” on page 14](#).

Tables related to the layout of the user interface are removed.

- Decreased network traffic.

Only the content that is necessary to currently render the user interface is included in responses. Extraneous content that the client must later discard or ignore is not included in responses. This decrease in content decreases the network traffic.

- No downtime during configuration deployment.

When you use Siebel Tools to configure a Siebel application or a Siebel Web template, you must restart the application server to deploy the updated SRF (Siebel Repository File) into the production environment. When you use Siebel Composer for this configuration, you just log in to the application again to view the deployed configuration. For more information, see [“About Compiling and Publishing” on page 10](#).

- More control in the client.

In addition to existing stubs for the presentation model and physical renderer in applets and views, stubs for the presentation model and physical renderer are available for Web pages. Consequently, the Siebel Configurator can use the presentation model of a Web page to alter its properties and behavior, and the Siebel Configurator can use the physical renderer of a Web page to change its rendering.

- Improved file loading.

The framework files that are needed to start a Siebel application are loaded from the manifest. The files for the presentation model and physical renderer that are needed to render Web pages are also loaded from the manifest.

About the User Interface for Siebel Composer

Siebel Composer implements some existing Siebel Tools functionality in a user interface that is consistent with the user interface for Siebel Business Applications. In the user interface for Siebel Composer, the items in the first level of the Object Explorer in Siebel Tools are available as screens that you select from the Side Menu icon in the top left corner of the screen. Also, the items in the lower levels of the Object Explorer in Siebel Tools are available as views and subviews that apply to those screens.

In Siebel Composer, you navigate between screens and views in the same way that you navigate between screens and views in Siebel Business Applications. For more information about this navigation, see *Siebel Fundamentals for Siebel Open UI*.

In Siebel Composer, you create, edit, and delete records in the same way that you perform these tasks in Siebel Business Applications. For example, to create a new record, you click the plus (+) icon. Likewise, you query for records and review details about records in the same way that you perform these tasks in Siebel Business Applications. For more information about performing these tasks, see *Siebel Fundamentals for Siebel Open UI*.

The functionality that is available from some screens in Siebel Tools is available in the menu that appears when you click the cogwheel icon in Siebel Composer. For example, to lock an object in Siebel Tools, right-click the object record and select Lock Object from the menu that appears. To lock an object in Siebel Composer, you click the cogwheel icon and select Lock Object from the menu that appears.

In Siebel Tools, objects include properties (or attributes). In Siebel Composer, objects include fields. For example, the Name property in Siebel Tools is the Name field in Siebel Composer.

About Compiling and Publishing

In Siebel Tools, you compile modifications to implement your configuration changes in the user interface of the Siebel application. You compile the modifications that you make to objects into the Siebel Repository File (SRF) of the Siebel application.

In Siebel Composer, you publish modifications to implement your configuration changes in the user interface of the Siebel application. You publish the modifications that you make to objects directly into the Siebel application. You do not specify a SRF in the publication process. For more information about publishing, see [“Publishing Modifications in Siebel Composer” on page 55](#).

Siebel Tools Features Included in Siebel Composer

Siebel Composer currently supports the following features in Siebel Tools:

- Locking projects and objects in the local repository.
- Unlocking projects and objects in the local repository.
- Configuring symbolic strings.

Siebel Tools Objects Included in Siebel Composer

Table 3 shows the objects in Siebel Tools that Siebel Composer currently supports.

Table 3. Objects in Siebel Tools That Siebel Composer Supports

Applet	Command	Pick List	View
Application	DLL	Project	Web Page
Business Component	Integration Object	Screen	Web Template
Business Object	Link	Symbolic String	
Business Service	Menu	Table	
Class	Message Category	Type	

Siebel Tools Features Excluded From Siebel Composer

Siebel Composer currently does not support the following features in Siebel Tools:

- Using Object Explorer and Object List Editor.
- Using various windows, wizards, and editors (such as layout editors, designer editors, and the Web template editor) that are only available in Siebel Tools.
- Managing Siebel Repository Files (SRFs) for Siebel Business Applications.
- Tagging objects to manage developer changes.
- Checking out projects and objects from the Server repository
- Checking in projects and objects to the Server repository.
- Validating, examining, and comparing objects.
- Using the Siebel Script Editor, the ST eScript Engine, the Siebel Debugger, and the Script Profiler.
- Configuring locale data and using the Locale Management utility to manage locales and Advanced Compile to localize repositories.
- Compiling modifications to projects and objects.
- Setting debug options.
- Exporting objects to an archive, importing object from an archive, and using Application Deployment Manager (ADM).

Siebel Tools Objects Excluded From Siebel Composer

Table 4 shows the objects in Siebel Tools that Siebel Composer currently does not support.

Table 4. Objects in Siebel Tools that Siebel Composer currently does not support

Assignment Attribute	Icon Map	Server Component Type
Assignment Criteria	Import Object	System Activity Object
Bitmap Category	Pager Object	Task
Content Object	Repository	Task Group
Dock Object	Schema Maintenance Phase	Toolbar
EIM Interface Table	Schema Maintenance Process	Workflow Policy Column
Entity Relationship Diagram	Schema Maintenance Step	Workflow Policy Object
Find	Search Category	Workflow Policy Program
Help Id	Search Engine	Workflow Process
HTML Hierarchy Bitmap	Search Index	

Aspects of Responsive Web Design in Siebel Composer

Siebel Composer allows for a more responsive Web design. Responsive Web design is an approach to Web design aimed at crafting sites to provide an optimal viewing experience (easy reading and navigation with minimal resizing, panning, and scrolling) across a wide range of devices (from desktop computer monitors to mobile devices).

The more responsive Web design feature includes the following aspects:

- A flatter hierarchy structure in the Siebel Web templates and in the DOM (Document Object Model) for more efficient Web page rendering. The DOM is a way to refer to HTML elements as objects.

The Siebel Business Applications for Siebel Tools include a deeper hierarchy structure in Siebel Web templates. The flatter hierarchy structure in Siebel Business Applications for Siebel Composer, with limited element nesting, provides more flexibility to render the screens in an application session and to render application sessions on different devices, such as mobile devices. The application session is rendered more efficiently because multiple requests to the server are not necessary for rendering. Also, only the needed elements are used to render the application session.

The flatter structure facilitates easier update of Siebel Web templates. For example, in the Siebel Business Applications for Siebel Tools, table tags are used to position the fields in a form in the user interface, but in the Siebel Business Applications for Siebel Composer, table tags are not necessary for this positioning.

- A database equivalent for the information in Siebel Web templates.
Instead of accessing and adjusting the .swt (Siebel Web template) file to configure Web templates, you enter information in records and fields in the Web Template Item view of the Web Template screen in Siebel Composer to configure Web templates.
- Additional fields and views to render Web pages, views, and applets. For more information, see [“Added Fields in Siebel Composer” on page 14](#) and [“Added Views in Siebel Composer” on page 17](#).

About Designating Responsive Web Design for Siebel Business Applications

Make sure that the EnableResponsive parameter is set to a value of TRUE for all Siebel Business Applications that use the responsive Web design feature. This parameter enables Siebel Composer. The default value for this parameter is FALSE. You can access this parameter in the Component Parameters list for the component of the Siebel application in the Servers view of the Administration - Server Configuration screen.

Some applets, views, and dialog boxes in those views in Siebel Business Applications are not subject to responsive Web design for this release. Oracle developers for Siebel must enhance these objects in future releases before they can be subject to responsive Web design. The Responsive field of the Applet screen is cleared for applet objects that are currently not subject to responsive Web design.

The Responsive field of the View screen is cleared for view objects that are currently not subject to responsive Web design. If a view is not subject to responsive Web design, then the applets that are included in that view are also not subject to responsive Web design. Only Oracle developers use the Responsive field.

External object interfaces, such as Chat, Dashboard, and Search, are not subject to responsive Web design for this release. These object interfaces are not fully functional with the responsive application shell.

Added Fields in Siebel Composer

All of the properties (attributes) in Siebel Tools are available as fields in Siebel Composer. However, Siebel Composer includes some additional fields. [Table 5](#) describes these added fields. These fields are included in the following views:

- Web Template Item view of the Web Template screen
- Web Page Item view of the Web Page screen
- View Web Template Item view of the View screen
- Applet Web Template Item view of the Applet screen

Table 5. New Fields in Siebel Composer

Field	View	Description
Template Item Holder Name	Web Page Item View Web Template Item Applet Web Template Item	Type a meaningful name for the value in the Item Identifier field, which is a placeholder field. This name allows you to more easily identify an element item that you want to find in the SiebelWeb template.
Item Identifier Name	Web Template Item	Type a meaningful name for the value in the Item Identifier field, which is a placeholder field. This name allows you to more easily identify an element item that you want to find in the SiebelWeb template.

Table 5. New Fields in Siebel Composer

Field	View	Description
Expression	Web Template Item Web Page Item View Web Template Item	<p>Type a standard Siebel expression for the element item to specify content and functionality the user interface. The expression is evaluated at run time to determine if the element qualifies for rendering in the user interface.</p> <p>For the expressions that you enter in the Applet Web Template Item view, you can specify global details, such as profile attributes, system preferences, and user login details. You can also specify information about business components. For the expressions that you enter in other views, you can specify only global details.</p> <p>All expressions must comply with the Query-Based Expression (QBE) syntax for standard Siebel expressions. For more information about expressions, see <i>Siebel Developer's Reference</i>.</p> <p>For examples of expression values, see the following topics for use cases:</p> <ul style="list-style-type: none"> ■ “Making Applets Visible in Specific Screen Resolutions” on page 50 ■ “Making Applets Visible to Specific Users” on page 51

Table 5. New Fields in Siebel Composer

Field	View	Description
Type	Web Template Item View Web Template Item	<p>In the Web Template Item view of Siebel Composer, you can select the following values in the Type field:</p> <ul style="list-style-type: none"> ■ Container. Designates a standard HTML element. For example, the <div> tag identifies a standard HTML element. ■ Content Item. Designates a placeholder to render a Web template item for an applet, a view, or a Web page. For example, the <swe:control> tag renders an applet control. <p>In the View Web Template Item view of Siebel Composer, you can select from among many values for user interface elements in the Type field, including Applet, Button, Text, and Link. View Web Template Item in the Object List Editor in Siebel Tools applies to only applets. (In Siebel Tools, you can add only an applet to a view.)</p> <p>In this release of Siebel Composer, only the user interface elements available in Siebel Tools are supported. In later releases, additional user interface elements will be supported.</p>
HTML Attributes	Web Template Item Web Page Item View Web Template Item	Type the class names that apply to rendering the element item in the user interface.
Sequence	Web Template Item Web Page Item View Web Template Item Applet Web Template Item	Type a number to designate the order in which the element item appears in the user interface relative to the other element items.
Group Name	Web Template Item Web Page Item View Web Template Item Applet Web Template Item	Type the name of the parent element item for the element item to establish a hierarchy between the two items.

Table 5. New Fields in Siebel Composer

Field	View	Description
Web Template Name	Web Page Item View Web Template Item Applet Web Template Item	This field is reserved for future use.
Include Template	Web Template Item	Select another Web template in which to include the item for this Web template.
Repeat Config	Web Template Item	Type a number to indicate the number of times to repeat the element item. For example, you can enter the following text in this field instead of creating ten records for the element item: start=1;count=10.

Added Views in Siebel Composer

Many of the objects in the Object list Editor in Siebel Tools are available as screens and views in Siebel Composer, but Siebel Composer includes the following new views:

- Web Template Item
- Web Template Item Property
- Web Page Item Property
- View Web Template Item Property
- Applet Web Template Item Property

For an Item Property view, the records (or properties) in the view provide contextual meaning to render its parent view. For example, the View Web Template Property view provides contextual meaning to render the View Web Template Item view. In this release of Siebel Composer, complete support for the records in the property views is not yet available.

Planned Enhancements to Siebel Composer

Planned enhancements to this developing product include:

- WYSIWYG (What You See Is What You Get) display and drag-and-drop functionality in the user interface.
- Drop-down lists of values for some fields to replace free form text fields in which you enter data.
- Access to Siebel Composer from within the Siebel application.
- Management of a multi-user development environment for Siebel Composer.

- Publication of repository changes in a development, test, and production environment, where the changes are implemented without having to restart the server.

3

Setting up Siebel Composer

This chapter includes information about the setup tasks that you must perform to deploy Siebel Composer for a new deployment and an existing deployment that you originally installed for an earlier version. It includes the following topics:

- ["About Setting Up Siebel Composer"](#)
- ["Process of Setting Up Siebel Composer"](#)
- ["About the Web Template Migration Process"](#)
- ["About the Application Object Manager Component for Siebel Composer"](#)
- ["Adding the Application Object Manager Component for Siebel Composer"](#)
- ["Adding the Virtual Directory for Siebel Composer"](#)
- ["Configuring Siebel Composer"](#)
- ["Enabling the Composer Mode for the Selected and Opened Workspace to Edit Objects"](#)
- ["Navigating to the Workspace Dashboard in the Siebel Composer Mode Application"](#)
- ["Setting up Composer Application Edit"](#)

About Setting Up Siebel Composer

The setup tasks for you to complete depend on which of the following installation cases applies to you:

- If you install the current release in a new deployment (or as a new installation) and perform the required configuration tasks after installation, then the Application Object Manager component and virtual directory for Siebel Composer are available for the installed and deployed language. In this installation case, a Web template migration process is automatically completed. For more information, see ["About the Web Template Migration Process" on page 21](#).

For more information about performing a new installation, see the *Siebel Installation Guide* for the operating system you are using.

- If you install the current release as a migration installation for an existing Siebel installation of an earlier version, then, to use Siebel Composer, you must create the Application Object Manager component and virtual directory that Siebel Composer uses for the installed and deployed language. In this installation case, a Web template migration process is automatically completed. For more information, see [“About the Web Template Migration Process” on page 21](#).

For the Siebel Enterprise Server and the Siebel Web Server Extension (SWSE), you can install a Siebel Innovation Pack release for the current release as a migration installation. For more information about performing a migration installation, see the *Siebel Installation Guide* for the operating system you are using. For more information about the setup tasks to complete in a migration installation, see [“Process of Setting Up Siebel Composer” on page 20](#).

Install Siebel Composer in an independent development environment, and not a production environment. This environment must be associated with its own Siebel database and its own server modules. You must configure this environment to support Siebel Composer.

Siebel Composer is not supported on mobile devices.

Process of Setting Up Siebel Composer

To set up Siebel Composer

- 1 Determine the installation case that applies to you by reviewing the information in [“About Setting Up Siebel Composer” on page 19](#):
 - If the new installation case applies to you, then do not complete the steps in this process, and see the *Siebel Installation Guide* for the operating system you are using.
 - If the migration installation case applies to you, then complete the steps in this process.

- 2 Install the current release of Siebel as a migration installation for an existing Siebel installation of an earlier version.

NOTE: The overall installation process includes many specific tasks and requirements that are not mentioned in this topic. For more information about installing Siebel, see the *Siebel Installation Guide* for the operating system you are using.

- 3 Run Incremental Repository Merge for the Siebel database.

NOTE: You must run Incremental Repository Merge after you install the current release of Siebel as a migration installation for an existing installation of an earlier version. For more information about running Incremental Repository Merge, see *Siebel Database Upgrade Guide*.

- 4 Add the Application Object Manager component for Siebel Composer.

For more information, see [“Adding the Application Object Manager Component for Siebel Composer” on page 28](#).

- 5 Add the virtual directory on the Web server for Siebel Composer.

For more information, see [“Adding the Virtual Directory for Siebel Composer” on page 31](#).

About the Web Template Migration Process

A Web template migration process is automatically completed when:

- You install the current release in a new deployment.
- You run Incremental Repository Merge after you install the current release as a migration installation for an existing Siebel installation of an earlier version.

The time needed for Incremental Repository Merge is longer than the time in earlier releases because of the Web template migration process.

- You upgrade the Siebel database from a release earlier than the Siebel 8.1.1/8.2.2 release.

In this process, a script is run to convert file-based content for Siebel Web templates into the table-based content for Siebel Web templates. This table-based content appears in fields in Siebel Composer. For more information about the newly added fields in Siebel Composer, see [“Added Fields in Siebel Composer” on page 14](#).

The semantics of the logical Web template are retained in this conversion process. [Table 6](#) shows how the information in Siebel Web template files is mapped to the fields in Siebel Composer.

Table 6. Information in Siebel Web Template Files Mapped to Fields in Siebel Composer

Information in Siebel Web Template Files	Fields in Siebel Composer
HTML (Hypertext Markup Language) tag and SWE (Siebel Web Engine) tag For more information, see “HTML Tags Migrated to Siebel Composer” on page 22 and “SWE Tags Migrated to Siebel Composer” on page 23 .	Name
Name of nested content when tags are nested under another tag	Group Name
Order in which a tag appears in the SiebelWeb template	Sequence
Class names for an HTML tag or an SWE tag HTML	Attributes

New Web templates are created for the Siebel Web template files that are included in another Siebel Web template file.

All Siebel Web template files in the appropriate directories are included in the migration process. For more information about the appropriate directories, see [“Prerequisite for Web Template Migration Process” on page 22](#).

The migration script is supported in all environments and database platforms. For more information about installing a new Siebel database, see *Siebel Installation Guide* for the operating system you are using. For more information about upgrading the Siebel database and about using Incremental Repository Merge, see *Siebel Database Upgrade Guide*.

Prerequisite for Web Template Migration Process

In the Web template migration process, files associated with Siebel Web templates are migrated to table-based content from the appropriate directories. This process looks for a Siebel Web template, and then looks for its associated file in the *siebsrvr_root\webtemp\ouiwebtemp\custom* directory. If this file exists in this directory, then the process parses the file, and looks for the next Siebel Web template. If this file does not exist in this directory, then the process looks for the file in the *siebsrvr_root\webtemp\custom* directory. If this file exists in this directory, then the process parses the file, and looks for the next Siebel Web template. If this file does not exist in this directory, then the process includes the file in the List of Missing Files section of the *swt_mig.log* file. For more information, see [“Verifying the Results of Web Template Migration Process” on page 27](#).

Files associated with Siebel Web templates are not migrated from the *siebsrvr_root\webtemp\ouiwebtemp* directory and the *siebsrvr_root\webtemp* directory. Consequently, you must copy files associated with Siebel Web templates from these directories into the corresponding *custom* directories if the files do not already exist in the *custom* directories.

HTML Tags Migrated to Siebel Composer

[Table 7](#) shows the HTML (Hypertext Markup Language) tags that are migrated to fields in Siebel Composer in the Web template migration process.

Table 7. HTML Tags Migrated to Siebel Composer

Tag Name	General Tag Purpose
div	Generates a container record for a Web template item and for its properties.
li	Generates a container record for a Web template item and for its properties.
span	Generates a container record for a Web template item and for its properties.
ul	Generates a container record for a Web template item and for its properties.

SWE Tags Migrated to Siebel Composer

Table 8 shows the SWE (Siebel Web Engine) tags that are migrated to fields in Siebel Composer in the Web template migration process.

Table 8. SWE Tags Migrated to Siebel Composer

Tag Name	General Tag Purpose	Migration Restrictions
swe:applet	Generates a content item record for a Web template item, and updates the corresponding holder name of the template item in the mapped record of the view Web template item.	None.
swe:applet-assoc-mvg	Generates a content item record for a Web template item.	None.
swe:applet-tree-list	Generates a content item record for a Web template item, and updates the corresponding holder name of the template item in the mapped record of the view Web template item.	None.
swe:case	Generates a container record for a Web template item.	None.
swe:control	Generates a container record for a Web template item, and updates the corresponding holder name of the template item in the mapped record of the applet Web template item.	None.
swe:current-view	Generates a content item record for a Web template item, and updates the corresponding holder name of the template item in the mapped record of the Web page item.	None.
swe:default	Generates a container record for a Web template item.	None.
swe:form-applet-layout	Generates a container record for a Web template item.	None.
swe:frame	Generates a container record for a Web template item.	Only the following types are included in the migration: <ul style="list-style-type: none"> ■ AltView ■ AppMenu ■ ThreadBar
swe:if	Generates a container record for a Web template item.	None.

Table 8. SWE Tags Migrated to Siebel Composer

Tag Name	General Tag Purpose	Migration Restrictions
swe:if-var	Generates a container record for a Web template item.	<p>Only the following names are included in the migration:</p> <ul style="list-style-type: none"> ■ AppletFilter-No ■ AppletFilter-Yes ■ Child ■ ChildWithPointer ■ Grandchild ■ Parent ■ ParentWithPointer
swe:include	Populates the included template for a Web template item.	None.
swe:list	Generates a content item record for a Web template item, and generates a new control (ListControl) record for the applet Web template item in a mapped applet.	None.
swe:menu	Generates a content item record for a Web template item, and generates a new control (TreeControl) record for the applet Web template item in a mapped applet.	<p>Only the following types are included in the migration:</p> <ul style="list-style-type: none"> ■ Button ■ Default

Table 8. SWE Tags Migrated to Siebel Composer

Tag Name	General Tag Purpose	Migration Restrictions
swe:nav-control	Generates a content item record for a Web template item and for its properties.	<p>Only the following types are included in the migration:</p> <ul style="list-style-type: none"> ■ Category View ■ Detail Category ■ Detail Category View ■ Detail Category With View ■ Screen ■ Screen With Category <p>Only the following styling tabs are included in the migration:</p> <ul style="list-style-type: none"> ■ HB ■ Tab
swe:node	Generates a content item record for a Web template item.	None.
swe:pageitem	Generates a content item record for a Web template item, and updates the corresponding holder name of the template item in the mapped record of the Web page item.	None.
swe:pdqbar	Generates a content item record for a Web template item.	None.
swe:screenbar	Generates a content item record for a Web template item.	None.
swe:screenlink	Generates a content item record for a Web template item.	None.

Table 8. SWE Tags Migrated to Siebel Composer

Tag Name	General Tag Purpose	Migration Restrictions
swe:this	<p>When this tag appears under a swe:applet tag, swe:control tag, or swe:pageitem tag, generates a property for a Web template item under its parent tag. This property has a type of Control Property.</p> <p>When the property for this tag is RowCounter or Title, generates a record for a Web template item, and generates a new control (RowCounter or Title) record for the applet Web template item in a mapped applet.</p>	<p>Only the following properties are included in the migration:</p> <ul style="list-style-type: none"> ■ DisplayName ■ FormattedHtml ■ RequiredIndicator ■ RowCounter ■ Title
swe:togglebar	Generates a content item record for a Web template item, and generates a new control (StaticToggle) record for the applet Web template item in a mapped applet.	None.
swe:toolbar	Generates a content item record for a Web template item and for its properties.	None.
swe:view	Generates a content item record for a Web template item.	None
swe:viewlink	Generates a content item record for a Web template item.	None.

Manually Running the Migration Script

You can manually run the migration script for the Web template migration process by entering the following command in a command-line interface:

```
java -jar $DbsrvrRoot\common\SWTMigration.jar /s $SiebelRoot /c "$ODBCDataSource" /t $TableOwner /u $TableOwner /p $TablePassword /o $SiebelLogDir /d $DatabasePlatform /r "Siebel Repository" /l $SiebelLogDir/swt_migration.log
```

where:

- *\$DbsrvrRoot* is the path of the database server installation.
- *\$SiebelRoot* is the path of the Siebel Server installation.
- *\$SiebelLogDir* is the path to the log directory.
- *\$DatabasePlatform* is the database platform, such as Oracle, DB2UDB, MSSQL, and DB2390.

You can manually run the migration script as many times as needed.

CAUTION: Each time that you run the script, all of the existing table-based content for Siebel Web templates is deleted, and then the new table-based content is created.

Verifying the Results of Web Template Migration Process

After the migration script runs, you can access the swt_mig.log file to review the results of the Web template migration process. This file is available when you manually run the migration script and when the migration script automatically runs. Near the end of this file, the following sections appear:

- **List of Missing Files.**

This section includes the files that are referenced in the repository but that do not exist in the Siebel Web template file directories.

- **List of Files which cannot be parsed.**

This section includes the files that are not converted to fields in Siebel Composer because of an incorrect structure in the DOM (Document Object Model).

Publishing Changes to Siebel Web Templates

The Web template migration process generates the table-based content for Siebel Web templates. If the migration installation case applies to you, then you must publish the content in these tables so that the content is implemented in the Siebel application. If the new installation case applies to you, then publication of this content is automatically completed for you.

To publish the table-based content for Siebel Web templates, enter the following command in a command window on the computer where you installed Siebel Tools:

```
siebdev /c tools.s.cfg /u $SIEBUSER /p $SIEBPWD /d "$ODBCDataSource" /l enu /
fullpublish
```

About the Application Object Manager Component for Siebel Composer

The Application Object Manager component for Siebel Composer is Siebel Web Tools Object Manager (ENU). This component is currently available for only the U.S. English language. The alias for this component is SWToolsObjMgr_enu. This component is defined in the Siebel Web Tools component group (alias of SiebelWebTools). The application name for this component is Siebel Web Tools. This component uses the swtools.cfg configuration file.

Perform the tasks to create and configure this server component for existing installations for which this component is not currently defined. Then you perform the task to create the corresponding virtual directory on the Web server to support this component. For more information, see [“About Setting Up Siebel Composer” on page 19](#).

Adding the Application Object Manager Component for Siebel Composer

To add the Application Object Manager component, perform the following tasks:

- [“Editing the new_comp_sia.dat File” on page 28](#)

Edit the new_comp_sia.dat file to include a definition for the Siebel Web Tools Object Manager component.

- [“Running the new_compdef_sia Script” on page 30](#)

Perform these tasks on each Siebel Server on which you run this component. For more information, see [“About the Application Object Manager Component for Siebel Composer” on page 27](#).

NOTE: Perform these tasks only if they are required for your installation case. For more information, see [“About Setting Up Siebel Composer” on page 19](#).

This task is a step in [“Process of Setting Up Siebel Composer” on page 20](#).

Editing the new_comp_sia.dat File

You edit the new_comp_sia.dat file to define the Siebel Web Tools Object Manager (ENG) component. You can edit this file and then run the new_compdef_sia script on additional Siebel Servers, as necessary.

To edit the new_comp_sia.dat file

- 1 Stop the Siebel Server.
- 2 For the Siebel Gateway Name Server, complete the following steps:
 - a Stop the Siebel Gateway Name Server.
 - b Back up the siebns.dat file on the Siebel Gateway Name Server.
 - c Restart the Siebel Gateway Name Server.
- 3 On the Siebel Server computer, navigate to *SIEBSRVR_HOME/bin/lang*.

In this directory, *SIEBSRVR_HOME* is the Siebel Server installation location, and *lang* is the three-letter code for the language for which you edit the *new_comp_sia.dat* file. (Use the backslash for Microsoft Windows; use the forward slash for UNIX or Linux.)
- 4 Back up the existing new_comp_sia.dat file.
- 5 Using an appropriate text editor, such as Notepad on Microsoft Windows, open the new_comp_sia.dat file.

NOTE: When you edit new_comp_sia.dat, do not enter any special characters, such as tabs.
- 6 If the file does not contain a section for the Siebel Web Tools component group, then add the following information in the component group section of the file:
 - a As needed, append an entry for the SiebelWebTools component group, defined as follows:

```
[Component Group]
```

```
SiebelWebTools = Siebel Web Tools
```

- b** As needed, create a section for the SiebelWebTools component group, defined as follows:

```
[SiebelWebTools]
```

```
FullName = Siebel Web Tools
```

```
Description = Siebel Web Tools Components
```

```
CreateDisabled = TRUE
```

- 7** Add the following information in the component section of the file:

- a** Add a new entry for the Siebel Web Tools Object Manager component in the existing components section, defined as follows:

```
[Components]
```

```
SWToolsObjMgr_enu = Siebel Web Tools Object Manager (ENU)
```

- b** Add a new section for the Siebel Web Tools Object Manager component, defined as follows:

```
[SWToolsObjMgr_enu]
```

```
FullName = Siebel Web Tools Object Manager (ENU)
```

```
Alias = SWToolsObjMgr_enu
```

```
Description = Siebel Web Tools Object Manager
```

```
CompGrp = SiebelWebTools
```

```
CompType = AppObjMgr
```

```
FixedParam: DataSource = ServerDataSrc
```

```
FixedParam: Lang = ENU
```

```
Param: CfgRepositoryFile = siebel_si_a_srf
```

```
Param: LocalCode = ENU
```

```
Param: MaxTasks = 20
```

```
FixedParam: ConfigFile = swtools.cfg
```

```
FixedParam: NamedDataSource = ServerDataSrc, GatewayDataSrc, DataMart
```

```
Param: CFGApplicationName = Siebel Web Tools
```

```
Param: CFGApplicationTitle = Siebel Web Tools
```

```
Param: CFGApplicationSplashText = Siebel Web Tools
```

```
Param: SearchDefName = WebTools Definition
```

```
Param: ProxyEmployee = PROXYE
```

Param: LoginDomain = EXTERNAL
Param: AllowAnonUsers = TRUE
Param: EnableCDA = FALSE
Param: EnableEmailClientAutomation = FALSE
Param: ListRowStyle = Siebel List
Param: LoginView = Login View (WebTools)
Param: SystemSWFname = CHTMLType.swf
Param: UserAgentFile = ua.ini
Param: AppletSelectStyle = Applet Select
Param: HighInteractivity = TRUE
Param: EnableOpenUI = TRUE
RunMode = interactive

NOTE: The information that you specify includes several required parameter settings for the component. Some of the parameters enable Siebel Open user interface, which this application requires. You can use Siebel Server Manager to configure several of the parameter settings later, for example, to tune application performance.

- 8 Save the new_comp_sia.dat file.

Next, run the new_compdef_sia script. For more information, see [“Running the new_compdef_sia Script” on page 30](#).

Running the new_compdef_sia Script

You run the new_compdef_sia script to add the server component definition that you defined in the new_comp_sia.dat file. The script adds the Application Object Manager component that Siebel Composer (the Siebel Web Tools application) uses. Perform this task after you perform the task to edit the new_comp_sia.dat file. For more information, see [“Editing the new_comp_sia.dat File” on page 28](#).

When you run the script, you must specify the same language that you specify within the edited sections of the new_comp_sia.dat file. This language is U.S. English.

To run the new_compdef_sia script

- 1 Log in as a Siebel administrator user on the Siebel Server computer.
- 2 Set the SIEBEL_HOME environment variable to *SES_HOME/siebsrvr*.
(Use the backslash for Microsoft Windows; use the forward slash for UNIX or Linux.)
- 3 Open a command window or shell window and run a command as follows:

■ On Microsoft Windows, run a command like the following command:

```
new_compdef_sia.bat siebelgateway:port siebelenterprise username password language
```

■ On UNIX or Linux operating systems, run a command like the following command:

```
./new_compdef_sia.ksh siebelgateway:port siebelenterprise username password language
```

In these commands:

- *siebelgateway* is the name of the Siebel Gateway Name Server.
- *port* is the port number that it uses.
- *siebelenterprise* is the name of the Siebel Enterprise.
- *username* is the name of the Siebel user with certain privileges who runs the script (for example, SADMIN).
- *password* is the password for this user, and *language* is the language for which you create the Application Object Manager component.
- You must use a language of U.S. English (enu).
- Specifying the port number and the preceding colon are optional if the Siebel Gateway Name Server uses the default port number, 2320.

For example, on Microsoft Windows, you might run a command like the following command:

```
new_compdef_sia.bat siebgwty1 myenterprise sadmin password ENU
```

- 4 When prompted, verify that the parameters are correct.
- 5 Restart the Siebel Server.

The new component is added for the U.S. English language.

Next, create the virtual directory. For more information, see [“Adding the Virtual Directory for Siebel Composer” on page 31](#).

Adding the Virtual Directory for Siebel Composer

To add the virtual directory, perform the following tasks:

- 1 [“Editing the new_virdirs Script” on page 32](#)

Edit the `new_virdirs` script to include the virtual directory name required for Siebel Composer (the Siebel Web Tools application).

- 2 [“Running the new_virdirs Script” on page 33](#)

Run the `new_virdirs` script and specify the language for which you want to create a virtual directory.

Perform these tasks on each Web server where you install Siebel Web Server Extension (SWSE) and on which you require the virtual directory. These tasks also update the `eapps.cfg` file for the SWSE to add language-specific application sections. Perform these tasks only once so that multiple sections are not added to the file.

NOTE: Perform these tasks only if it is required for your installation case. For more information, see “About Setting Up Siebel Composer” on page 19.

This task is a step in “Process of Setting Up Siebel Composer” on page 20.

Editing the `new_virdirs` Script

You edit the `new_virdirs` script to specify the virtual directory name that Siebel Composer (the Siebel Web Tools application) requires. The virtual directory name is also used for the application section in the `eapps.cfg` file. You can run an edited script on additional Web servers, as necessary.

To edit the `new_virdirs` script

- 1 Stop the Web server.
- 2 Back up the `eapps.cfg` file, the `eapps_sia.cfg` file, and the Web server configuration file, such as `obj.conf`, `http.conf`, or `magnus.conf`, as appropriate for your browser.
- 3 Navigate to `SWSE_HOME/config`.

In this directory, `SWSE_HOME` is the SWSE installation location.

(Use the backslash for Microsoft Windows; use the forward slash for UNIX or Linux.)

- 4 Back up the existing `new_virdirs` script as follows:
 - On Microsoft Windows, the name of the script is `new_virdirs.bat`.
 - On UNIX or Linux, the name of the script is `new_virdirs.sh`.
- 5 Using an appropriate text editor, such as Notepad on Microsoft Windows, open `new_virdirs.bat` or `new_virdirs.sh`.

NOTE: When you edit the script, do not enter any special characters, such as tabs.

- 6 Add the elements for the Siebel Web Tools application to an existing entry in this file as follows so the file creates a new virtual directory and new `eapps.cfg` file section for the application:

```
%JAVA_HOME%\j ava -j ar %ORACLE_HOME%\confi g\new_vi rdi rs. j ar
```

```
{webtool s} {SWTool sObj Mgr} %LANG
```

In this entry:

- `webtools` is the name of the virtual directory
- `SWToolsObjMgr` is the name of the corresponding Application Object Manager without the corresponding language code. (The language-specific items are created when you run the script.)

- 7 Save the `new_virdirs` script.

Next, run the `new_virdirs` script to add the virtual directory for the specified language. For more information, see [“Running the new_virdirs Script” on page 33](#).

Running the new_virdirs Script

You run the `new_virdirs` script to add the virtual directory for the specified language that Siebel Composer (the Siebel Web Tools application) uses. Perform this task after you perform the task to edit the `new_virdirs` script. For more information, see [“Editing the new_virdirs Script” on page 32](#).

NOTE: If running the `new_virdirs` script does not create the required virtual directory, then you must restore the files on the Web server from the backup files that you created in Step 2.

To run the new_virdirs script

- 1 Navigate to `SWSE_HOME/config`.

In this directory, `SWSE_HOME` is the SWSE installation location. (Use the backslash for Microsoft Windows; use the forward slash for UNIX or Linux.)

- 2 Open a command window or shell window and run a command as follows:

- On Microsoft Windows, run a command like the following command:

```
new_virdirs.bat ENU
```

- On UNIX or Linux operating systems, run a command like the following command:

```
./new_virdirs.sh ENU webserverlocation
```

In this command, `webserverlocation` is the installation location of the web server instance.

- 3 Specify the anonymous user for Siebel Composer by adding parameters like the following parameters to the `eapps.cfg` file:

```
AnonUserName = guestusername
```

```
AnonPassword = guestuserpassword
```

Specify the anonymous user name and password in each language-specific application section.

NOTE: Add values similar to the values defined for other applications. In most cases, the `AnonPassword` value represents an encrypted value. To specify a different password, first encrypt the password by using the `encryptstring` utility. For more information, see *Siebel Security Guide*.

- 4 Restart the Web server.
- 5 Start a browser and start the Siebel Web Tools application by entering an appropriate URL, such as the following URL:

```
http://server_name/webtools_enu
```

In this command, `server_name` is the name of your server.

Configuring Siebel Composer

To configure Siebel Composer, perform the following tasks:

- 1 “Changing the Timing of Automatic Selection of Changed Field” on page 34
- 2 “Setting the Feature for Custom Symbolic Strings” on page 34

Changing the Timing of Automatic Selection of Changed Field

When you change a record in Siebel Composer, the Changed field of the record is automatically selected. The timing of this automatic selection is determined by the date and time value that you select in a field in the Composer Options view of the User Preferences screen.

The date and time value in this field automatically defaults to the date and time that you log in to Siebel Composer for the first time. Each user can change this value as needed. If you are not interested in reviewing records that were changed before a certain date and time, then select that date and time in the field in this view. Although records might be changed before this date and time, the Changed field is not selected for these changed records. If you clear this field so that no date and time appears in it, then the field value defaults to the current date and time.

To change the timing of automatic selection of Changed field

- 1 Navigate to the User Preferences screen, then the Composer Options view.
- 2 Select the appropriate date and time value in the following field: Mark as changed all records that have been updated since.

Setting the Feature for Custom Symbolic Strings

You can change the EnableToolsConstrain parameter in the configuration file for Siebel Composer to allow or to not allow users to create custom symbolic strings. By default, users are allowed to create custom symbolic strings. You can temporarily change this parameter to not allow users to create custom symbolic strings, and then later set this parameter back to its default value.

To set the feature for custom symbolic strings

- 1 On the Siebel Server computer, navigate to `SIEBSRVR_HOME/bin/eng`.
In this directory, `SIEBSRVR_HOME` is the Siebel Server installation location. (Use the backslash for Microsoft Windows; use the forward slash for UNIX or Linux.)
- 2 Using an appropriate text editor, such as Notepad on Microsoft Windows, open the `swtools.cfg` configuration file.
NOTE: When you edit `swtools.cfg`, do not enter any special characters, such as tabs.
- 3 Change the value in the EnableToolsConstrain parameter as follows:

- a Set the parameter to FALSE to allow users to create custom symbolic strings.
 - b Set the parameter to TRUE to not allow users to create custom symbolic strings.
- 4 Save the swtools.cfg file.

Enabling the Composer Mode for the Selected and Opened Workspace to Edit Objects

Users with the Siebel Responsibility *Composer Administrator* can use the Application Edit (hammer) icon on the top-right corner of the Siebel application window to enable the Composer mode for Siebel workspaces. They can also use this same Application Edit icon to revert the Siebel application from the Composer mode to the runtime-only mode when necessary.

NOTE: The Application Edit icon (hammer) is available only when a workspace is selected, enabled, and available for editing, and it applies only to users who have the Siebel Responsibility *Composer Administrator*. For more information, see “Navigating to the Workspace Dashboard in the Siebel Composer Mode Application” on page 39.

Figure 1 illustrates the Siebel application window after the Composer mode is enabled. The Application Edit and the Workspace icons are available at the top right corner of the window. Four main sections of Siebel Composer for Workspace are also displayed.

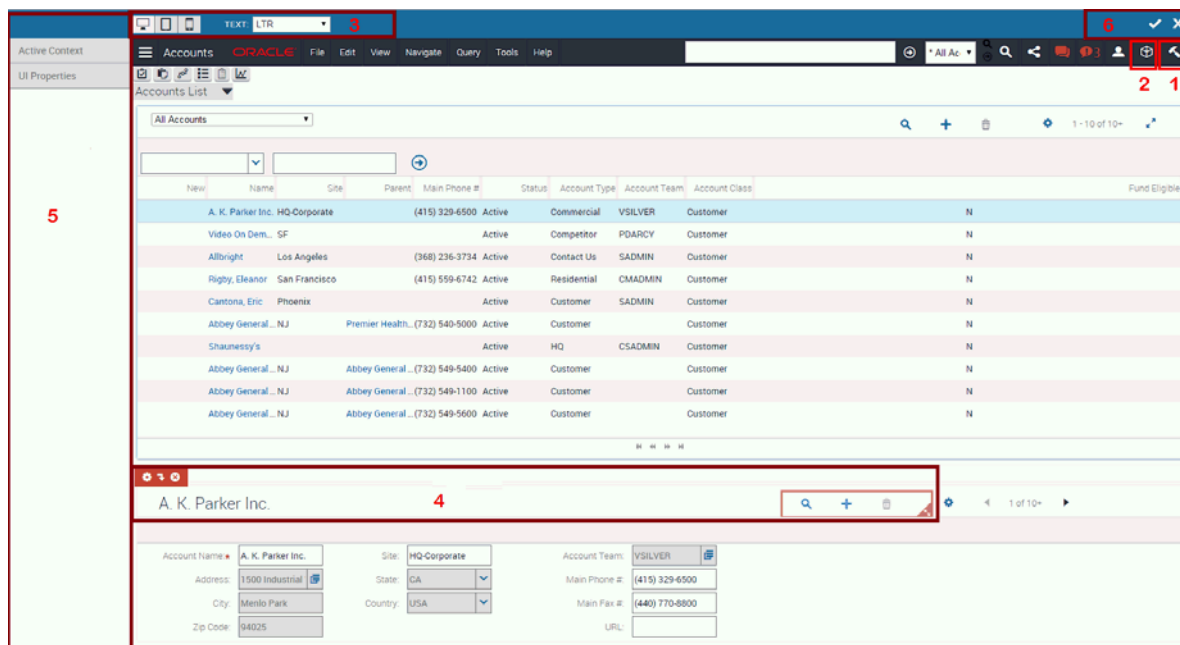


Figure 1. Example of Siebel Composer for workspaces

Explanation of Callouts

Figure 1 includes the following information:

1 Application Edit icon

Use the Application Edit (hammer) icon to switch Siebel applications between the Composer mode and the Runtime-only mode.

2 Workspace icon

Use the Workspace (cube) icon to navigate to the Workspace Dashboard view. For more information, see [“Navigating to the Workspace Dashboard in the Siebel Composer Mode Application” on page 39](#).

3 Composer toolbar

The Composer toolbar governs the overall context of the composer configuration. For more information about the Composer toolbar, see [“Composer Toolbar” on page 36](#).

4 Hover context

When you move the mouse across various elements on the Siebel application, the application dynamically changes the hover context and surrounds the current context by the orange border. For more information about the hover context, see [“Hover Context” on page 37](#).

5 Context panel

The context panel on the left-hand section of the Siebel Composer is a logical representation of the UI context that is currently being worked on. For more information about the context panel, see [“Context Panel” on page 38](#).

6 Execution bar

The Execution bar appears on the top-right corner of the Siebel Composer application and consists of three buttons. For more information about how to use the buttons on the Execution bar, see [“Execution Bar” on page 39](#).

Composer Toolbar

The Composer Toolbar currently consists of these two sections: Platform Simulator and Text Direction Charger.

- The Platform Simulator section enables users to configure the Siebel application for the different viewports (modes). When any of these viewports are used, the application is reloaded to simulate the user experience in that particular view. Note that the layout changes made to the active context are saved against the selected viewport only.

These three viewports are available in the Platform Simulator section:

- Desktop view

Select the Desktop icon to configure the application in desktop mode (or a full-screen size).

- Tablet view

Select the Tablet icon to configure the application in tablet mode (or a medium-screen size).

■ Phone view

Select the Phone icon to configure the application in phone mode (or a compact-screen size).

- The Text Direction Changer section displays the Text drop-down list that is used to configure the Siebel application for the different text directions. Note that the layout changes made to the active context are saved against the selected text direction only.

The Text drop-down list includes these two text directions: LTR (left-to-right) and RTL (right-to-left).

When the toggle is changed between the two text directions, the application is reloaded to simulate the user experience in that particular text direction.

NOTE: The Platform Simulator and the Text Direction Changer sections can be used in conjunction to configure a combination of modes. In addition, the Composer Toolbar may have further global options in future releases of Siebel Composer.

Hover Context

Moving the mouse across various elements on the Siebel application dynamically changes the hover context as represented by the changing orange border surrounding the current context. Various UI artifacts may activate on the screen depending on the operations of the particular context.

Figure 2 illustrates the Siebel application displaying the hover context, which includes context menu, container border, and resize indicators.

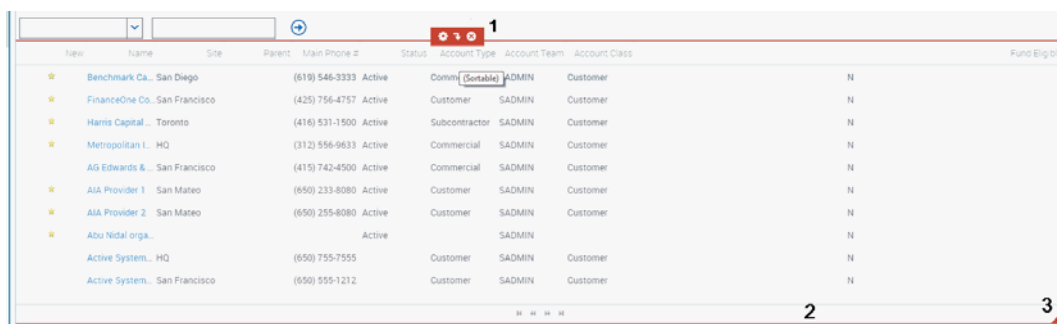


Figure 2. Example of the Siebel application displaying the hover context

Explanation of Callouts

Figure 2 includes the following information:

1 Context menu

The context menu is a small popup that appears on the top left of the element that is currently being worked on, depending on whether it has a configurable repository representation.

The following buttons are available in the context menu:

- Properties button

This button is not in use in the current Siebel Tools version.

- Drilldown button

Use this button to change the context in the Context Pane to the current element.

- Delete button

Use this button to delete the current element.

2 Container border

The container border indicates the element in the Siebel application UI that is currently being operated on. This element informs the user that any changes that are being performed at this point will be applicable to all of the elements inside this border.

3 Resize indicator

The resize indicator is a small icon that appears on the bottom right corner of the element that is being currently worked on, depending on whether it is a resizable element.

For more information about how to resize the indicator, see [“Resizing Elements of Siebel Composer” on page 53](#).

Context Panel

The context panel reflects the context that will be altered when users perform operations on the system. Also, certain operations may be available only in certain contexts.

The context panel consists of the following two sections, which you click to expand:

- Active Context

On expansion, the Active Context section shows a tree representation of the current context on the Siebel application. The tree includes the children elements that are available for configuration. The tree may also include elements that are not visible in the UI, but the ones that are configured in the repository.

By default, the current Siebel view is the active context. The active context can be changed using the Context Menu on the Hover Context. For more information about Context Menu, see [“Hover Context” on page 37](#).

The bottom of the Active Context list includes the buttons used to add elements to the active context.

NOTE: In this version of Siebel Composer, you can only add List Columns to the valid Siebel List Applet when the list columns are listed in the Active Context section. For more information, see [“Modifying the List Applet Columns” on page 54](#).

- UI Properties

The UI Properties section on expansion shows a box model of the hovered element in the Siebel application. This section reflects the pixel sizes of the element itself, and all paddings and offsets that may be applicable on it.

Execution Bar

The Siebel Composer application accumulates the user's changes at the Siebel client as operations are done using the Composer UI. The Siebel application uses the following methods to commit these changes to the server:

- When users explicitly commit changes using the Execution Bar.
- Through an option that is available to users when a navigation occurs.
- Implicitly when certain operations are performed by users on the Siebel Composer application.

The Execution Bar appears in the top right corner of the Siebel Composer application and consists of these buttons:

- The Pending Commit Counter (number) button.

This button indicates the number of changes that were made on the Composer user interface and accumulated at the Composer client, which are not yet committed to the Siebel server.

- The Commit Changes (check mark) button.

Use this button to explicitly commit any pending Composer changes at the client to the Siebel server.

- The Discard Changes (X) button.

Use this button to discard any pending Composer changes at the client to the Siebel server.

Note that sometimes the Siebel Composer application displays a modal dialog box asking users to either commit or discard changes that have accumulated. An example of this case is when a user makes certain changes in the Composer UI and then navigates to a different view without committing or discarding the changes. At other times, the Siebel Composer application may intelligently choose to commit the changes to the server; for example, when a user adds or deletes a list column using the Siebel Composer application, any changes that are pending at the client at this point are automatically committed to the server.

At all points, the Pending Commit Counter reflects the changes at the client which are yet to be committed to the server.

Navigating to the Workspace Dashboard in the Siebel Composer Mode Application

You can use the Workspace (cube) icon in any Siebel application that is set in the Composer mode to access and open the Workspace Dashboard view, and then create or edit workspaces

NOTE: The Application Edit (hammer) icon is only available after you create a new workspace or open an existing one using the Workspace dashboard, and then click the Close (X) button to close the Workspace dashboard.

To access and open the Workspace Dashboard view in the Composer mode

- 1 On the Siebel application window, click the Workspace Dashboard (cube) icon at the top-right corner of the window.

The Workspace dashboard appears.

- 2 Either create a new workspace or open an existing one.
- 3 Close the Workspace dashboard by clicking the Close (X) button at the top-right corner of the Workspace dashboard.

The Siebel application reappears displaying the Application Edit (hammer) icon at the top-right corner of the window.

To enable the Workspace feature in the Composer mode

- 1 Ensure that Siebel Tools Client for release IP2016 is installed successfully.

For more information on how to install Siebel Tools, see the *Siebel Installation Guide*.

- 2 Open the Command Prompt window from your computer by clicking the Start button and then selecting the Run option.

The Run window appears.

- 3 Enter the value *CMD* in the Open field.

The Command Prompt window appears.

- 4 Change the directory in the Command Prompt window to the *<TOOLS_HOME>/BIN* folder using this command:

```
cd <TOOLS_HOME>/BIN
```

The Command Prompt window displays the following arguments and parameters, which you can use to run the EnableWorkspace.exe utility:

- -s Siebsrvr/Tools Installation path specified (required)
 - -t Siebel Table Owner (required)
 - -u Username (required)
 - -p Password (required)
 - -o ODBC Data Source (required)
 - -d DB Platform Name (Oracle, MSSQL, DB2UDB or DB2390)
 - -l Log File Name (default: EnableWorkspace.log)
 - -w Workspace Owner Username (required)
- 5 Use the listed arguments and parameters to run the EnableWorkspace.exe utility.

The following example shows the arguments and parameters that are used to run the EnableWorkspace.exe utility in a MSSQL database. In this example, *dbo* is the table owner of the MSSQL database and *MSZXM187* is the database instance that is used in this example environment.


```
EnableWorkspace -s C:\Tools23057 -t dbo -u MSZXM187 -p MSSQL -o SSD -d MSSQL -l  
EnableWorkspace.log -w SADMIN
```

The following example shows the arguments and parameters that are used to run the EnableWorkspace.exe utility in an Oracle database. In this example, *orakwj115* is the table owner and the user ID of the Oracle database, and these parameters are used to execute the SQL statements that exist in the EnableWorkspace.exe utility.

```
EnableWorkspace -s C:\Tools4 -t orakwj115 -u orakwj115 -p orakwj115 -o "SSD" -d  
oracle -l EnableWorkspace.log -w SADMIN
```

6 View the output stages, representing the progress of the tasks performed by the utility.

The Command Prompt window lists the following output stages after you run the EnableWorkspace.exe utility:

- Stage 1 of 7: Setting the Repository ID
- Stage 2 of 7: Validate the Workspace Data
- Stage 3 of 7: Generate the Workspace Data
- Stage 4 of 7: Update the Workspace Data in all Repository Tables
- Stage 5 of 7: Generate the indexes
- Stage 6 of 7: Update the statistics for all Workspace enabled Repository Tables
- Stage 7 of 7: Post Run Cleanup

The Workspace feature is now enabled in the Composer mode.

To create a new workspace in the Workspace dashboard

- 1** Click the Create button on the upper-left corner of the Workspaces dashboard.
- 2** Enter a workspace name and click the OK button.

The new workspace with the name that you entered is created and automatically opened.

To open an existing workspace in the Workspace dashboard

- 1** If necessary, expand the Main workspace (parent workspace) to display a list of the children workspaces.
- 2** Select an existing workspace from the list of the children workspaces under the Main workspace.
- 3** Click the Open button on the Workspace toolbar.

The selected workspace is opened.

Setting up Composer Application Edit

To set up Siebel Composer Application Edit to work with Siebel REST

1 Set up a reverse proxy to front the Siebel REST Java Web Container and the Siebel Web Server. For more information about setting up a reverse proxy, see ["IIS Reverse Proxy Configuration" on page 43](#).

2 Install or upgrade the Siebel server to Innovation Pack 2016.

For more information about installing and upgrading Siebel server, see the *Siebel Installation Guide* for the operation you are using.

3 Log into Server Manager and perform the following steps:

a Enable EAI (Enterprise Application Integration) Comp group using this command:

```
enable compgrp EAI
```

b To enable the access to the Siebel's repository data using REST, create a named subsystem by name JAVA as in this example:

```
create named subsystem JAVA for subsystem jvmsubsys
```

```
change param DLL= <location of JVM DLL>jvm.dll for named subsystem JAVA
```

```
change param CLASSPATH= $SIEBEL_ROOT\classes\Siebel.jar; $SIEBEL_ROOT\classes\SiebelJI_enu.jar for named subsystem JAVA
```

```
change param VMOPTIONS="-Xdebug -Xrunjwp:transport=dt_socket,address=8000,server=y,suspend=n" for named subsystem JAVA
```

NOTE: While setting up Siebel Composer to work with Siebel REST, you must replace *<location of JVM DLL>* and *\$SIEBEL_ROOT* with the actual paths that configures your environment.

For more information on how to configure Siebel server, see *Siebel System Administration Guide*.

4 While configuring server using Siebel Configuration Wizard, specify users for anonymous access using these parameters:

- Anonymous User Login Name
- Anonymous User Password
- Anonymous User Password (confirm)

Depending on your requirement, this anonymous user may be either a different anonymous user that is set up for the dedicated REST channel usage or the same anonymous user that you specified in the previous step as the contact user in the SWSE logical profile (used for anonymous access for Siebel Open UI Applications).

If the Siebel server is already configured, the anonymous login name and password can be specified or changed using the following server manager commands:

```
create named subsystem JavaContainerPropSubSys for subsystem
JavaContainerPropSubSys
```

```
change param javacontainerpropprosubsystype=JavaContainerPropSubSys for comp
SiebelRestComplntMgr
```

```
change param AnonUser=<User Name> for named subsystem JavaContainerPropSubSys
```

```
change param AnonPwd=<Encrypted User Password> for named subsystem
JavaContainerPropSubSys
```

NOTE: The password in the previous command must be encrypted using the *encryptstring.exe* utility located in *siebsrvr\bin*.

- 5 Restart the Siebel server.
- 6 After you restart the Siebel server, ensure that:
 - EAI comp is running
 - *Javacontainer1* folder is created under *\$SIEBEL_ROOT\javacontainer*
 - *Javacontainer* (tomcat) is running at port 9001
- 7 Access *http://<server name>:<port if applicable>\siebel-rest\v1.0\heartbeat* and ensure this location is either available or blank.
- 8 Complete the following steps to set up Siebel Composer Application Edit to work in Siebel Composer:
 - a Access *\$SIEBEL_ROOT\javacontainer\javacontainer1*.
 - b Edit the *siebsrvr.properties* file.
 - c Set the property *MaxConnection* and restart *Comp SiebelRestComplntMgr* using this command:


```
MaxConnection = 100

Restart comp SiebelRestComplntMgr
```
 - d In Server Manager, run this command:


```
change param busobjcachesize = 0 for comp eaiobjmgr_enu

Restart comp eaiobjmgr_enu
```

IIS Reverse Proxy Configuration

When the system is set in Application Edit mode, it maintains two connections between the browser and the Siebel server: one connection uses the REST Repository API that is served by the Java Container and the other connection connects with the legacy Siebel web server. These connections must be served from the same domain so they can be accessible to the same Siebel Composer Application Editing Client on the browser. For this purpose, the Java Container and the Siebel web server must be jointly fronted by a reverse proxy.

The following software is required to implement the load balancing and reverse proxy configuration:

■ IIS 8 and above

Oracle Siebel implements a reverse proxy with IIS.

■ URL Rewrite

You can download URL Rewrite from <http://www.iis.net/downloads/microsoft/url-rewrite>.

■ Application Request Routing (ARR)

You can use ARR as a load balancer for multiple servers and as a reverse proxy that hides other web servers from the internet.

This software can be downloaded from <http://www.iis.net/downloads/microsoft/application-requestrouting>.

While setting up Reverse Proxy for Siebel Composer Application Edit, note that:

- The current Siebel REST interface is served by a Java Web Container (also called *Tomcat* in Siebel Innovation Pack 2016). The remaining Siebel applications are all served by a different web server that runs the Siebel Web Extension Plug-in.
- The typical URLs for these two categories are formatted as in [Table 9](#). For Siebel Composer Application Edit to work, both URLs of Siebel Web Server Extension (SWSE) and Siebel REST must be on the same domain.
- These settings can be implemented using a reverse proxy that is hosted on *<HostName>* and served by the default HTTP port (or a specified non-default port).

Table 9. Typical URL formats and URLs on the same domain of Siebel implementations

Category of Siebel implementations	Typical URL format	URL on the same domain
SWSE	<a href="http://<SWSEHostName>/callcenter_enu">http://<SWSEHostName>/callcenter_enu	<a href="http://<HostName>/callcenter_enu">http://<HostName>/callcenter_enu
Siebel REST	<a href="http://<TOMCATHostName>:<port_number>/siebel-rest/v1.0/repository/Applet/SIS Account List Applet">http://<TOMCATHostName>:<port_number>/siebel-rest/v1.0/repository/Applet/SIS Account List Applet	<a href="http://<HostName>/siebel-rest/v1.0/repository/Applet/SIS Account List Applet">http://<HostName>/siebel-rest/v1.0/repository/Applet/SIS Account List Applet

NOTE: The configuring steps in the following section are based on the system that has the reverse proxy configured on the same host, which is running the SWSE and routing appropriate requests to the Siebel REST Java Web Container (Tomcat).

To set up reverse proxy for Siebel Composer Application Edit

1 Install URL Rewrite and ARR:

- a** Download URL Rewrite from <http://www.iis.net/downloads/microsoft/url-rewrite>.

b Download ARR from <http://www.iis.net/downloads/microsoft/applicationrequest-routing>.

For more information on how to download and install URL Rewrite and ARR, visit the IIS website.

- 2** In IIS, access Application Request Routing.
- 3** Click on the server proxy settings.
- 4** In the Application Request Routing page, select the *Enable proxy* option.
- 5** In the Proxy Type section, select the *Use URL Rewrite to inspect incoming requests* option.
- 6** Enter the reverse proxy URL in the *Reverse Proxy* text box.
- 7** Click the *Apply* button.
- 8** Navigate to Sites in IIS and then navigate to Default Site.
- 9** Click the URL Rewrite icon.
- 10** Add a rewrite rule.

The Edit Inbound Rule dialog appears.

- 11** Perform the following steps in the Edit Inbound Rule dialog:
 - a** Enter a rewrite rule name in the *Name* field.
 - b** Enter the following value into the *Pattern* text box in the Match URL section:
(Siebel -rest. +)
 - c** Select the Rewrite option from the *Action type* drop-down list.
 - d** Enter the REST Java Web Container URL in the *Rewrite URL* text box.
For example, <http://localhost:9001/{R:0}>.

- 12** Apply the changes and restart IIS.

The reverse proxy setup is completed.

- 13** Test the configuration.
- 14** Send the REST request with web server as host name (without port number).

An appropriate message appears informing you that the settings are successfully completed.

For more information on how to add the rewrite rules for the URL rewrite module, visit the IIS website at www.iis.net/learn/extensions/url-rewrite-module/creating-rewrite-rules-for-the-url-rewrite-module.

4

Use Cases for Siebel Composer

This chapter includes information about use cases for Siebel Composer. It includes the following topics:

- [“Changing Logos in Application Banners”](#)
- [“Changing Applet Fields to Read-Only Fields”](#)
- [“Changing Labels for Applet Fields”](#)
- [“Making Applets Visible in Specific Screen Resolutions”](#)
- [“Making Applets Visible to Specific Users”](#)
- [“Repositioning Elements of Siebel Composer”](#)
- [“Resizing Elements of Siebel Composer”](#)
- [“Modifying the List Applet Columns”](#)
- [“Publishing Modifications in Siebel Composer”](#)

Changing Logos in Application Banners

In this use case, you change the logo in the banner of the Siebel application from the Oracle logo to the Oracle Open World logo. The logo in the banner of the Siebel application is included in the CCFrameBanner Web template.

To change the logo in the application banner

- 1 In the Siebel application, complete the following steps to determine the item identifier and the class for the logo in the banner:
 - a Navigate to the Home screen.
 - b Right-click the Oracle logo in the banner, and select Inspect Element from the menu that appears.

A window that shows the HTML source code for the screen appears.
 - c In the line of highlighted source code, note that the logo is associated with the div_70 item identifier and the siebui-logo class.
 - d Close the window that shows the HTML source code.
- 2 In the Siebel application, complete the following steps to determine the runtime version of the Siebel application that you are using:
 - a In the application-level menu, click Help, and select About SRF from the menu that appears.

- b** In the About SRF dialog box, note the version number of the Siebel Repository File (SRF) in the File Name field, and click OK.
- 3** In Siebel Composer, complete the following steps to change the logo in the banner of the Siebel application:
 - a** Navigate to the Web Template screen, and query for CCFrameBanner in the Name field.
 - b** Click the cogwheel icon, and select Lock Object from the menu that appears.
 - c** Scroll down to the Web Template Items list, and query for div_70 in the Item Identifier Name field.
 - d** In the record for the div_70 item identifier, change the value in the HTML Attributes field to: class = "siebui-logo siebui-oow-logo" to include the class for the Oracle Open World logo.
 - e** In the Web Template screen, click the cogwheel icon, and select Unlock Object from the menu that appears.
- 4** In Siebel Composer, publish the changes to the Web Template object.

For more information, see [“Publishing Modifications in Siebel Composer” on page 55](#).
- 5** Log out of the Siebel application, log in to it again, and notice the following changes:
 - The logo in the banner of the home page is now the Oracle Open World logo.
 - The version number in the File Field in the About SRF dialog box is increased by 1 to the version number that you published.

Changing Applet Fields to Read-Only Fields

In this use case, you change the field for the main phone number to a read-only field. This field is located in the Accounts List view of the Accounts screen in the Siebel application.

To change an applet field to a read-only field

- 1** In the Siebel application, complete the following steps to determine the name of the field that you want to change and the name of the applet that contains that field:
 - a** Navigate to the Accounts screen, then the Accounts List view.
 - b** Note that the display name for the main phone number field in the accounts list is Main Phone # and that you can edit this field value.
 - c** In the application-level menu, click Help, and select About View from the menu that appears.
 - d** In the About View dialog box, note that the name of the applet for the accounts list is SIS Account List Applet, and click OK.
- 2** In the Siebel application, complete the following steps to determine the runtime version of the Siebel application that you are using:
 - a** In the application-level menu, click Help, and select About SRF from the menu that appears.

- b** In the About SRF dialog box, note the version number of the Siebel Repository File (SRF) in the File Name field, and click OK.
- 3** In Siebel Composer, complete the following steps to change the main phone number field in the accounts list applet of the Siebel application:
 - a** Navigate to the Applet screen, and query for SIS Account List Applet in the Name field.
 - b** Click the cogwheel icon, and select Lock Object from the menu that appears.
 - c** Scroll down to the List Columns list, and query for Main Phone # in the Display Name field.
 - d** In the record for the main phone number field, select the Read Only field to make the main phone number field a read-only field.
 - e** In the Applet screen, click the cogwheel icon, and select Unlock Object from the menu that appears.
- 4** In Siebel Composer, publish the changes to the applet object.
For more information, see [“Publishing Modifications in Siebel Composer” on page 55](#).
- 5** Log out of the Siebel application, log in to it again, and notice the following changes:
 - The main phone number field for the records in the Accounts List view of the Accounts screen is now read only.
 - The version number in the File Field in the About SRF dialog box is increased by 1 to the version number that you published.

Changing Labels for Applet Fields

In this use case, you change the label for the Site field to Account Site. This field is located in the Accounts List view of the Accounts screen in the Siebel application.

To change the label for an applet field

- 1** In the Siebel application, complete the following steps to determine the name of the field that you want to change and the name of the applet that contains that field:
 - a** Navigate to the Accounts screen, then the Accounts List view.
 - b** Note that the display name for the site field in the accounts list is Site.
 - c** In the application-level menu, click Help, and select About View from the menu that appears.
 - d** In the About View dialog box, note that the name of the applet for the accounts list is SIS Account List Applet, and click OK.
- 2** In the Siebel application, complete the following steps to determine the runtime version of the Siebel application that you are using:
 - a** In the application-level menu, click Help, and select About SRF from the menu that appears.
 - b** In the About SRF dialog box, note the version number of the Siebel Repository File (SRF) in the File Name field, and click OK.

- 3 In Siebel Composer, complete the following steps to change the label for the site field in the accounts list applet of the Siebel application:
 - a Navigate to the Applet screen, and query for SIS Account List Applet in the Name field.
 - b Click the cogwheel icon, and select Lock Object from the menu that appears.
 - c Scroll down to the List Columns list, and query for Site in the Display Name field.
 - d In the record for the site field, change the value in the Display Name field from Site to Account Site.
 - e In the Applet screen, click the cogwheel icon, and select Unlock Object from the menu that appears.
- 4 In Siebel Composer, publish the changes to the applet object.

For more information, see [“Publishing Modifications in Siebel Composer” on page 55](#).
- 5 Log out of the Siebel application, log in to it again, and notice the following changes:
 - The label for the site field in the Accounts List view of the Accounts screen is now Account Site.
 - The version number in the File Field in the About SRF dialog box is increased by 1 to the version number that you published.

Making Applets Visible in Specific Screen Resolutions

In this use case, you make the form applet in the Accounts List view of the Accounts screen in the Siebel application visible only on computer screens with a screen resolution width of more than 1024 pixels.

To make an applet visible in specific screen resolutions

- 1 In the Siebel application, complete the following steps to determine the name of the view that contains the applet for which you want to change visibility and to determine the name of that applet:
 - a Navigate to the Accounts screen, then the Accounts List view.
 - b In the application-level menu, click Help, and select About View from the menu that appears.
 - c In the About View dialog box, note that the name of the view is Account List View and the name of the form applet is SIS Account Entry Applet, and click OK.
- 2 In the Siebel application, complete the following steps to determine the runtime version of the Siebel application that you are using:
 - a In the application-level menu, click Help, and select About SRF from the menu that appears.
 - b In the About SRF dialog box, note the version number of the Siebel Repository File (SRF) in the File Name field, and click OK.

- 3 In Siebel Composer, complete the following steps to make the form applet in the Accounts List view of the Siebel application visible only on computer screens with a screen resolution width of more than 1024 pixels:
 - a Navigate to the View screen, and query for Account List View in the Name field.
 - b Click the cogwheel icon, and select Lock Object from the menu that appears.
 - c Scroll down to the View Web Template Items list, and enter the following value in the Expression field of the record for the SIS Account Entry Applet: `GetProfileAttrAsNum ("Viewport:width")>1024`.
 - d In the View screen, click the cogwheel icon, and select Unlock Object from the menu that appears.
- 4 In Siebel Composer, publish the changes to the applet object.
For more information, see [“Publishing Modifications in Siebel Composer” on page 55](#).
- 5 Change the resolution of your computer screen to a width of less than 1024 pixels.
The first number of the resolution is the screen width in pixels. For example, a resolution of 800 x 600 designates a screen width of 800 pixels. In Microsoft Windows, you can change the resolution of your computer screen by navigating to Control Panel, Display, and then Adjust Resolution.
- 6 Log out of the Siebel application, log in to it again, and notice the following changes:
 - The form applet no longer appears in the Accounts List view of the Accounts screen.
 - The version number in the File Field in the About SRF dialog box is increased by 1 to the version number that you published.

Making Applets Visible to Specific Users

In this use case, you make the form applet in the Accounts List view of the Accounts screen in the Siebel application visible only to the user with a user ID of SADMIN.

To make an applet visible to a specific user

- 1 In the Siebel application, complete the following steps to determine the name of the view that contains the applet for which you want to change visibility and to determine the name of that applet:
 - a Navigate to the Accounts screen, then the Accounts List view.
 - b In the application-level menu, click Help, and select About View from the menu that appears.
 - c In the About View dialog box, note that the name of the view is Account List View and the name of the form applet is SIS Account Entry Applet, and click OK.
- 2 In the Siebel application, complete the following steps to determine the runtime version of the Siebel application that you are using:
 - a In the application-level menu, click Help, and select About SRF from the menu that appears.

- b In the About SRF dialog box, note the version number of the Siebel Repository File (SRF) in the File Name field, and click OK.
- 3 In Siebel Composer, complete the following steps to make the form applet in the Accounts List view of the Siebel application visible only to the user with a user ID of SADMIN:
 - a Navigate to the View screen, and query for Account List View in the Name field.
 - b Click the cogwheel icon, and select Lock Object from the menu that appears.
 - c Navigate to the Applet Web Template view.
 - d Scroll down to the Applet Web Template Items list, and enter the following value in the Expression field of the record for the SIS Account Entry Applet: `LoginName()='SADMIN'`
 - e In the View screen, click the cogwheel icon, and select Unlock Object from the menu that appears.
- 4 In Siebel Composer, publish the changes to the applet object.
 For more information, see [“Publishing Modifications in Siebel Composer” on page 55](#).
- 5 Log out of the Siebel application, log in to it again with a user ID other than SADMIN (for example, QADMIN), and notice the following changes:
 - The form applet no longer appears in the Accounts List view of the Accounts screen.
 - The version number in the File Field in the About SRF dialog box is increased by 1 to the version number that you published.

Repositioning Elements of Siebel Composer

As a layout editor, the Siebel Composer user interface (UI) can be used to reposition existing elements. Users can reposition both Container and Content Item elements into the available placeholders as determined by the Siebel Composer application. While repositioning, Siebel UI automatically represents how the UI will eventually look by rearranging the other elements in the context.

To reposition elements of Siebel Composer

- 1 Hover around the Siebel application screen to determine the element or group of elements to be repositioned.
- 2 The Hover Context represents what will be repositioned.
 Use this visual cue to determine the correct elements to be affected.
- 3 Once the element is determined, click a spot within the hover context, which is not an actual Siebel application element.
 This step is necessary because repositioning is mouse-driven and using an actual Siebel element may trigger unexpected actions on it.
- 4 Use the mouse to start dragging the elements represented by the Hover Context.

- 5 As the mouse is moved around with the elements, valid drop zones are dynamically calculated by the Siebel Composer applications and are represented by a green border.
- 6 Drop the elements in any valid area to complete the repositioning action.
The Siebel Composer application records the change.
- 7 In the Execution Bar, verify that the change was recorded by noting an increment in the Pending Commit Counter.

This example shows a drag action-in-progress of a container in the Account Form applet. The container consists of the elements corresponding to Query, Add, and Delete actions on the applet. This container element, while previously to the left of the Menu element, is now being dragged to the right of the pagination elements. The green border represents a valid drop zone for this element.

The screenshot shows the Siebel Composer interface for editing the 'A. K. Parker Inc.' account. The form contains several fields: Account Name (A. K. Parker Inc.), Site (HQ-Corporate), Account Team (YSILVER), Address (1500 Industrial), State (CA), Main Phone # ((415) 329-6500), City (Menlo Park), Country (USA), and Main Fax # ((440) 770-8800). A green border highlights a container element on the right side of the form, indicating it is a valid drop zone.

Resizing Elements of Siebel Composer

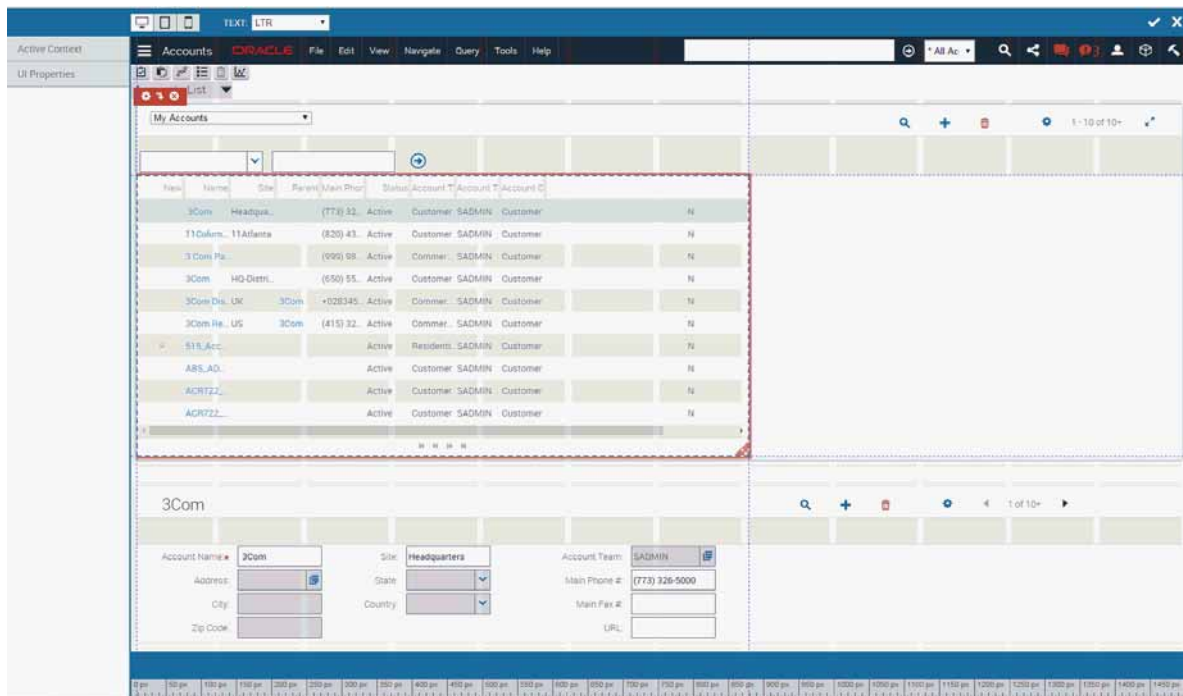
As a layout editor, the Siebel Composer UI can be used to resize the existing elements. Both Container and Content Item elements can be resized. A snap-to-grid model is followed during resize, which means that the resizing happens to predetermined breakpoints on the UI. This approach is beneficial because it enables users to vertically align elements on different rows to the same size. During the resizing, a background ruler is visible to serve as a visual cue to the possible breakpoint zones and to determine the final resulting size of the element.

To resize elements of Siebel Composer

- 1 Hover around the Siebel application screen to determine the element or group of elements to be resized.
- 2 The Hover Context represents what is being resized.
Use this visual cue to determine the correct elements to be affected. Valid resizable elements show a resize indicator.
- 3 Once the element is determined, position the mouse cursor over the resize indicator and start dragging horizontally.
A background ruler appears to indicate the potential resize breakpoints.
The Siebel UI dynamically resizes the element and represents how the UI will eventually look.
- 4 Drop the cursor when the expected resizing is complete.
The Siebel Composer application records the change.

- 5 In the Execution Bar, verify that the change was recorded by noting an increment in the Pending Commit Counter.

This example illustrates a resize action-in-progress of the grid container in the Account List applet. Inside the applet, the container hosts the grid itself. While previously occupying 100% of the applet's width, it is now resized to occupy about 60% of the width.



Modifying the List Applet Columns

As a data model editor, the Siebel Composer user interface (UI) can be used to add or remove list columns from the valid list applets in the Siebel application. Only valid List Column controls in the repository are displayed in the catalog for addition of new columns. Additionally, by virtue of being a UI-based editor, only the list columns that are currently displayed on the UI are available to delete.

To add a list column to a valid list applet

- 1 Use the Drilldown icon in the Context Menu of the list applet to set the context to a valid list applet.
- 2 Expand the Active Context section in the Context Panel.

- 3 At the bottom of the section, click the New List Column button.

A List Column Catalog appears displaying the unmapped List Column controls from the repository. The catalog is displayed with the available list columns for this applet.

NOTE: Entries with the greenish-blue shade are unavailable for mapping because the expressions that are applied on them do not satisfy in current state of the Siebel application.

- 4 Select one of the available list columns and click the Add button.

Note that:

- All uncommitted changes are committed to the server (Pending Commits Counter changes to 0).
- The Siebel application is refreshed.
- The new column is now added to the user interface.

To delete the list columns from the valid list applets

- 1 Hover inside a valid list applet on the List Column's header row.

This action displays the Context Menu for each of the column headers.

- 2 Determine the column to be deleted from the user interface.

- 3 Click the Delete button in the Context Menu.

Note that:

- All uncommitted change are committed to the server (Pending Commits Counter changes to 0).
- The Siebel application is refreshed.
- The deleted column is no longer visible in the user interface.

Publishing Modifications in Siebel Composer

After you change objects in Siebel Composer, you must publish your modifications to the repository tables of metadata so that your changes are implemented in the Siebel application, such as Siebel Call Center. The Siebel application uses this metadata to interact with the enterprise data and to interact with the people who use the Siebel application.

In Siebel Tools, you compile your modifications. In Siebel Composer, you publish your modifications.

If you are logged in to the Siebel application after you publish your modifications to an object in Siebel Composer, then your published version of the object is not reflected the user interface of the Siebel application. You must log out of the Siebel application, and then log in again to see the effect in the user interface of your published version of the object.

To publish modifications in Siebel Composer

- 1 Navigate to the Administration - Runtime Configuration screen, then the Publish view.

This view shows the objects that you have modified, but not yet published.

- 2 Complete either of the following steps:

- To publish all of the object records that appear on the screen, click Publish All.
- To publish some of the object records that appear on the screen, select the object records, and click Publish.

The object records that you publish are moved from the Publish view to the Published Versions view.

- 3 (Optional) Navigate to the Published Versions view to see information about the repository and version for the object records that you publish.
- 4 If you are logged in to the Siebel application, you must log out and then log in again to see your modifications in the Siebel application.

Index

- A**
- added fields, in Siebel CRM Composer** 14
- applet fields**
 - changed labels for 49
 - changing to read-only fields 48
- applets**
 - making visible in specific screen resolutions 50
 - making visible to specific users 51
- application banners, changing logos in Application Object Manager component** 47
 - about for Siebel CRM Composer 27
 - adding for Siebel CRM Composer 28
- C**
- Changed field, changing the automatic selection of** 34
- compiling, about** 10
- custom symbolic strings, setting the feature for** 34
- H**
- HTML tabs, migrated to Siebel CRM Composer** 22
- L**
- labels, changing for applet fields** 49
- logos, changing in application banners** 47
- M**
- migration script, manually running** 26
- N**
- new_comp_sia.dat file, editing** 28
- new_compdef_sia script, running** 30
- new_virdirs script**
 - editing 32
 - running 33
- P**
- publishing, about** 10
- R**
- read-only fields, changing applet fields**
 - to 48
- responsive Web design**
 - about designating for Siebel Business Applications 13
 - aspects of 12
- S**
- screen resolutions, making applet visible in specific** 50
- Siebel Business Applications, about designating responsive Web design for** 13
- Siebel CRM Composer**
 - about 7
 - about setting up 19
 - about user interface for 9
 - benefits of 8
 - configuring 34
 - planned enhancements to 17
 - process of setting up 20
 - publishing modifications in 55
- Siebel Tools features**
 - excluded from Siebel CRM Composer 11
 - included in Siebel CRM Composer 10
- Siebel Tools objects**
 - excluded from Siebel CRM Composer 12
 - included in Siebel CRM Composer 11
- Siebel Web templates**
 - publishing changes to 27
- SWE tags, migrated to Siebel CRM Composer** 23
- U**
- user interface, about for Siebel CRM Composer** 9
- users, making applets visible to specific** 51
- V**
- virtual directory, adding for Siebel CRM Composer** 31
- W**
- Web template migration process**
 - about 21
 - prerequisites for 22
 - verifying the results of 27

