Siebel

Chat Guide

January 2020



Siebel
Chat Guide

January 2020

Part Number: F12671-02

Copyright © 2020, Oracle and/or its affiliates. All rights reserved

Authors: Siebel Information Development Team

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs 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 Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

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

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

Contents

Preface i

1 What's New in This Release 1

What's New in Siebel Chat Guide, Siebel CRM 20.1 Update 1
What's New in Siebel Chat Guide, Siebel CRM 19.1 Update 1

2 Overview of Siebel Chat 3

Overview of Siebel Chat 3
About Siebel Chat 3
Siebel Chat Process Flows 6

3 Setting Up Siebel Chat 11

Setting Up Siebel Chat 11
Installing Customer-Facing Components 11
Roadmap for Configuring Siebel Call Center for Siebel Chat 11
Required Parameter Configuration When Using Siebel Chat on Oracle Solaris 24

4 Working with Siebel Chat 25

Working with Siebel Chat 25
Siebel Chat Interface 25
Processing Siebel Chat Requests 27
Verifying That Siebel Chat Is Installed for Siebel Call Center 37
Reviewing Inbound Siebel Chat Activities 38
About the Communications Toolbar 41
Logging in to Siebel Chat 41
Receiving a New Chat 42
Accepting an Incoming Chat 42
Responding to an Incoming Chat 44
Handling Siebel Chat Sessions 46
About the Chat Dashboard 50

Changing Agent Status 52
Agent and Siebel Chat Interaction States 53
Setting Up Logging and Tracing for Siebel Chat Components 55
About Web Notifications 57

5 Customizing Siebel Chat 59
Customizing Siebel Chat 59
About Customizing Siebel Chat 59
Siebel Call Center and Siebel Chat-Related Applets 59
Configuring the Appearance of Transcript Attachments 60
Configuring the Behavior of the Push URL Functionality 60
Customizing the Siebel Chat User Interface 61
Process of Configuring Siebel Chat Feedback 74

6 Configuring Siebel Chat Auto Answer 79
Configuring Siebel Chat Auto Answer 79
Roadmap for Configuring Siebel Chat Auto Answer for Agents 79
Process of Configuring Communications for Siebel Chat Auto Answer 81

7 Localizing Siebel Chat 87
Localizing Siebel Chat 87
Setting Up Siebel Chat for Languages Supported by Third-Party Chat Solutions 87
Setting Up Siebel Chat for Languages Not Supported by Default by Third-Party Chat Solutions 88

8 Siebel Chat Deployment Using Third-Party Chat Solutions 89
Siebel Chat Deployment Using Third-Party Chat Solutions 89
Deployment of Siebel Chat Using Third-Party Chat Solutions 89

9 Methods for Customizing Siebel Chat Process Flow and Interface Functionality 93
Methods for Customizing Siebel Chat Process Flow and Interface Functionality 93
Methods for Customizing Siebel Chat Process Flow and Interface Functionality 93

Preface

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

Using Oracle Applications

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

Documentation Accessibility

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

Contacting Oracle

Access to Oracle Support

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

Comments and Suggestions

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

1 What's New in This Release

What’s New in Siebel Chat Guide, Siebel CRM 20.1 Update

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

What’s New in Siebel Chat Guide, Siebel CRM 19.1 Update

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

2 Overview of Siebel Chat

Overview of Siebel Chat

This chapter describes Siebel Chat, its features, and the Siebel Chat process flows in a typical contact center. It includes the following topics:

- *About Siebel Chat*
- *Siebel Chat Process Flows*

About Siebel Chat

Siebel Chat is a customer contact channel whereby users in real-time can chat with, for example, a customer service representative (otherwise known as an agent) if they encounter a problem, with a view to troubleshooting and solving that problem.

Siebel Chat is a Web-based application for handling secure chat communications, including the following:

- Handling Siebel Chat interactions using chat-related buttons on the communications toolbar of the Siebel Call Center application.
- Conducting Siebel Chat conversations with a customer by using frequently used text, the URL library, smart share, and by searching the knowledge base and sharing resolutions.
- Creating activities and service requests from Siebel Chat interactions.
- Transferring Siebel Chat interactions to workgroups and agents.
- Terminating Siebel Chat sessions.



Note: The procedures in this guide assume that you do not use the Tree navigation control option to access screens and views. However, you can choose to use the Tree navigation control if required. For more information about setting navigation options, see Siebel Fundamentals for Siebel Open UI.

No programming experience is required to set up and use Siebel Chat. However, basic Siebel application navigation knowledge is required to effectively use the Siebel Chat client. Some planning and information gathering are necessary to set up your Siebel Chat application and have it running quickly and smoothly.

This guide includes information about how to:

- Plan and set up Siebel Chat.
- Configure a communications driver for the Siebel Chat channel.
- Use the Siebel Chat application to handle chat requests.

This guide describes the features of Siebel Chat that are available to a contact center agent. To learn more about setting up and configuring Siebel Chat, review *Setting Up Siebel Chat*.

Refer to the following Siebel server-side documents on the Siebel Bookshelf:

- Siebel Installation Guide for the operating system you are using

- Siebel System Administration Guide
- Siebel CTI Administration Guide
- Siebel Email Administration Guide
- Configuring Siebel Business Applications
- The Certifications tab on My Oracle Support.

For information about Siebel system requirements and supported platforms, see the Certifications tab on My Oracle Support. For more information about Certifications, see article 1492194.1 (Article ID) on My Oracle Support.



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

Siebel Chat Features

Common features of Siebel Chat include the following:

General Siebel Chat Features

The following general features are available for Siebel Chat:

- Communications Toolbar Enhancements. Using the communications toolbar, agents can sign in to Siebel Chat and make themselves available for incoming chat requests, put chats on hold, transfer chats, release chat sessions, and close chat interactions.
- Customer Dashboard Enhancement. The customer dashboard alerts the agent if a customer requesting a chat provides profile information that does not match the contact record.
- Siebel Chat Interaction Tracking. Every incoming chat interaction is tracked as an Activity under either a Contact or Service Request record. The chat transcript is also attached to the Activity record.
- Workspace Persistence. During multiple simultaneous Siebel Chat sessions where the agent can switch between chat sessions, the application holds the context for each customer. In addition, the customer dashboard persists with the chat session.
- Agent Anonymity. This feature enables agents to have an alias (or use a different first name).
- Anonymous Chat. This feature allows a customer to chat with an agent anonymously.

Siebel Chat Dashboard Features

The following features are available on the Siebel Chat dashboard:

- Smart Knowledge Management Integration. Siebel Chat is integrated with Oracle's Siebel Knowledge Base system (such as, solutions). The customer-facing application (for example, any self-service Web application) remembers the knowledge base articles that the customer reviews and uses this information when suggesting further articles (Have you tried this?) to review. The knowledge base visited (KB Visited) feature, available in the Siebel Chat dashboard, allows the agent to view the list of knowledge base articles that the customer has already viewed.
- Quick Actions. Agents can perform popular actions quickly (for example, create and view service requests, view activities and contacts) by selecting them from the Action drop-down menu in the Siebel Chat dashboard. Actions performed this way are contextual and smart.

Siebel Chat Message Features

The following message features are available for Siebel Chat:

- Smart Share. Clicking Smart Share allows the agent to share highlighted text, service request details, and solution details with the customer.
- Frequently Used Text. Agents have a global list of predefined message responses at their disposal to send to customers, as required, when handling Siebel Chat interactions.

Siebel Chat Push URL Features

The following push URL features are available for Siebel Chat:

- Push URL. This feature enables the agent to share a Uniform Resource Locator (URL) with a customer, which is automatically displayed in the customer's browser. The customer does not have to manually copy and paste the URL in to a browser window.
- URL Library. Agents have a global list of URLs at their disposal to share with customers using push URL technology.

Other Features

Other features available for Siebel Chat include the following:

- Business Intelligence. Supervisors and agents can use Oracle Business Intelligence (formerly known as Siebel Business Analytics), if installed, against the Siebel Chat data.
- Third-Party Chat Integration. Siebel Chat can be fully integrated with the communications technology of a third-party chat solution.

Siebel Chat Users

The following types of users are involved in the setup and usage of Siebel Chat:

- Agents who use Siebel Chat to handle chat requests. Agents log in to Siebel Chat using the communications toolbar in the Siebel Call Center application.
- Administrators who set up and configure Siebel Chat, and supervisors who monitor agent activities and workloads.

The following table outlines the main access level permissions for agents.

User	Access Level Permission
Agent	Agents can: • Log in to the Siebel Call Center application and Siebel Chat • Accept and respond to Siebel Chat requests • Transfer Siebel Chat interactions to other workgroups and agents • Release Siebel Chat sessions • Create inbound service requests
Administrator	Administrators can:

User	Access Level Permission
	• Log in to Siebel Chat • Manage critical data to set up and configure Siebel Chat • Manage user accounts for agents and supervisors • Create and run reports
Supervisor	Supervisors can: • Monitor agent statistics, workgroups, projects, and Siebel Chat interactions • Set alarms • Create and run reports

Siebel Chat Process Flows

Siebel Chat process flows consist of customer-facing and agent-facing activities. Customer facing activities are conducted within a customer-facing application, such as, any self-service Web application. Siebel Chat, by default, provides an API which can be used to integrate the customer-facing application with the Siebel Chat functionality. This API also contains a chat window to be embedded inside the customer-facing application. As such, the customer-facing Siebel Chat interface is a public Web-based interface and the means through which a customer requests a chat with an agent. By using browser-based controls on the customer-facing chat interface, a customer can send text to and receive text from an agent, as well as review content pushed by the agent.

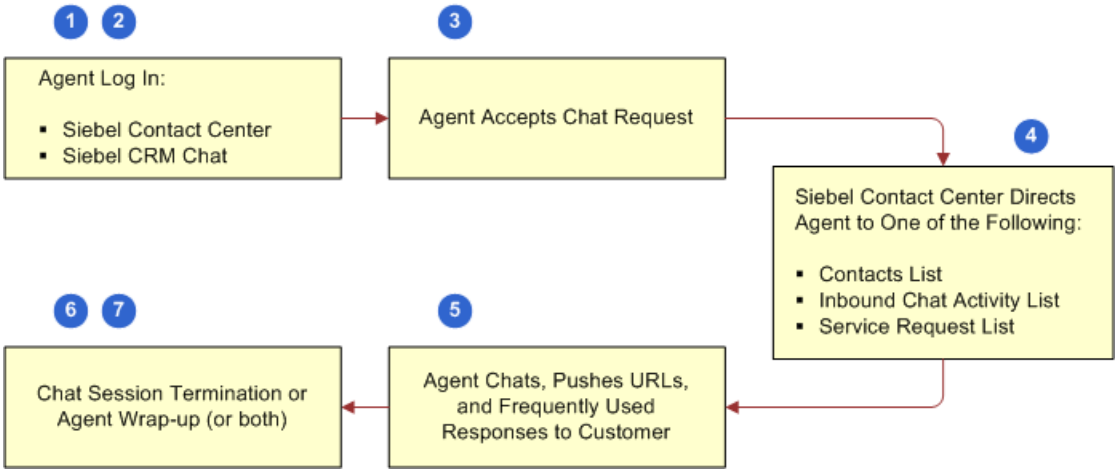
Agent-facing interactions are conducted using the Siebel Chat application that is embedded inside the Siebel Call Center application. The agent-facing Siebel Chat interface is the means through which an agent accepts and conducts a chat conversation with a customer. Through the agent-facing Siebel Chat interface, the agent can send and receive text, push URLs to, and select frequently used text to send to the customer.

For more information about the chat process flows for the customer and the agent, see the following:

- *Agent-Facing Chat Process Flow* describes a typical chat process flow for an agent.
- *Customer-Facing Chat Process Flow* describes a typical chat process flow for a customer.

Agent-Facing Chat Process Flow

The following image describes an agent-facing chat process flow.



The steps involved in a typical agent-facing chat process flow, shown in this image, are as follows:

1. The agent logs in to the Siebel Call Center application.
2. The agent logs in to Siebel Chat by clicking the Log in button on the communications toolbar.

   To use Siebel Chat, agents must be set up for the application, as follows:

   a. A user account must be set up for each agent in the third-party chat server.

      For information about setting up user accounts for agents, see the documentation provided by the third-party chat vendor.

   b. Each agent must be added to at least one chat communications configuration.

      For more information, see *Configuring Communications in Siebel Call Center for Siebel Chat* and *Adding Agents to the Communications Configuration for Siebel Chat*. For information about configuring agents after an agent has been added to a communications configuration, see *Siebel CTI Administration Guide* .
3. The agent clicks the flashing Accept Work Item button on the communications toolbar to accept an incoming chat request.

   For more information about accepting a chat, see *Accepting an Incoming Chat*.
4. Siebel Call Center attempts to identify the customer requesting a chat using the information provided on the chat request form.

   The information on the chat request form determines whether an activity event or a service request is created, and consequently the application view that the agent is directed to in the Siebel Call Center application. For example, if the customer provides only a Contact ID on the chat request form, then the agent is directed to the Contact screen in Siebel Call Center. However, if the customer provides a valid Service Request ID that Siebel Call Center can identify, then the agent is directed to the Service Requests screen in Siebel Call Center. For more information about how Siebel Call Center identifies the customer requesting a chat, see *Chat Standard Accept Process Workflow*.
5. The agent is directed to an appropriate application view in Siebel Call Center and engages in a chat conversation with the customer.

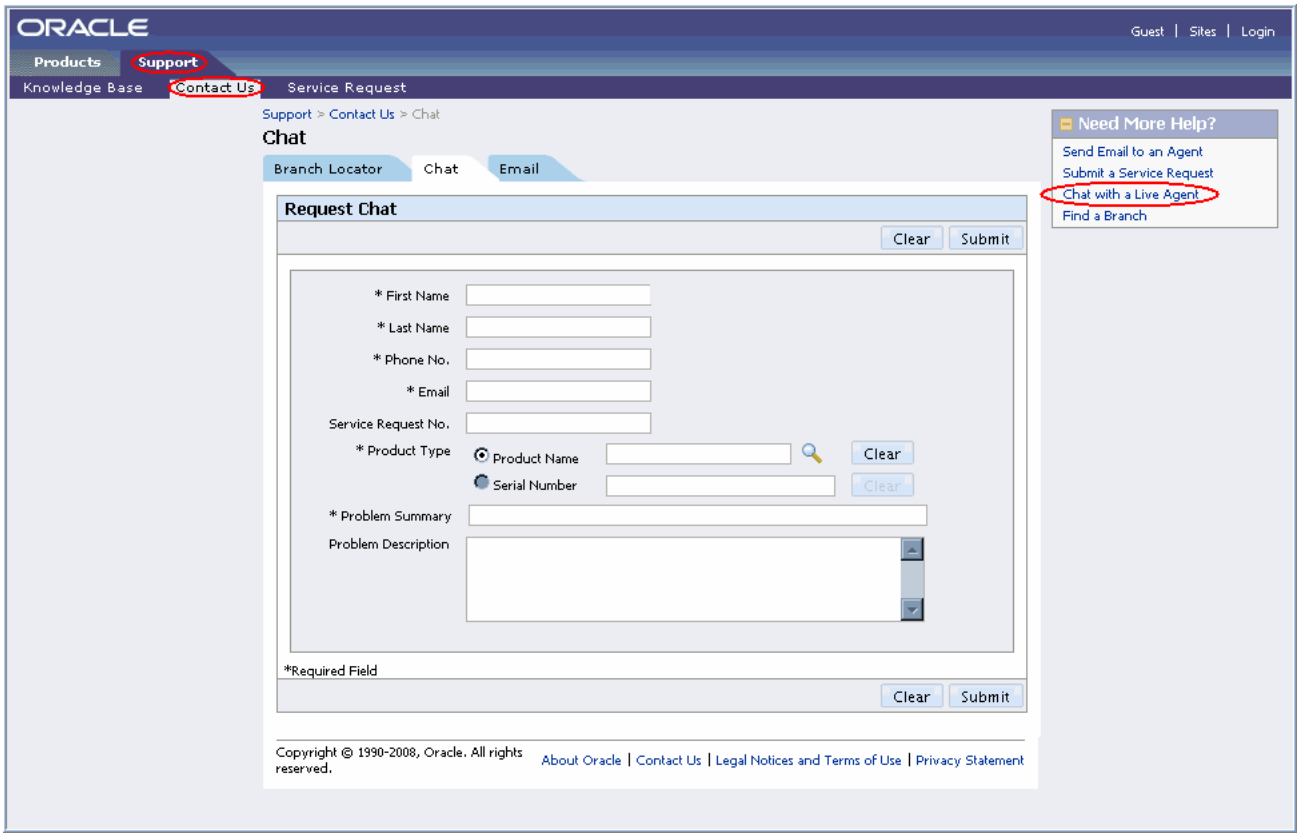
   For more information about how an agent handles a chat, review the following:

   - *Responding to an Incoming Chat*
   - *Reviewing Inbound Siebel Chat Activities*
   - *Handling Siebel Chat Sessions*

6. The agent releases the chat session, see *Releasing Chat Sessions*.

Chat Request Form

The following image shows an example of a chat request form.

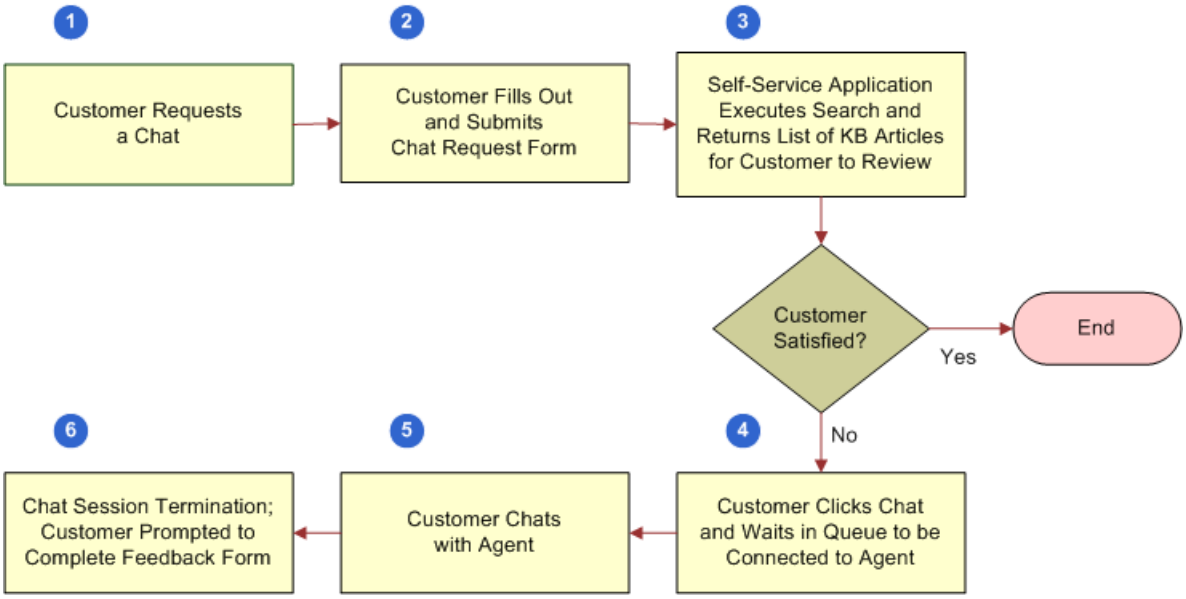


Some information might be prepopulated in the form, as follows:

1. Profile information. The first name, last name, phone number, and email address information is prepopulated if the user is logged in to the self-service Web application.
2. Service request information. The service request number, product name, problem summary, and problem description information is prepopulated if the chat is requested from a Service Request Detail page.

Customer-Facing Chat Process Flow

The following image describes a customer-facing chat process flow.



The steps involved in a typical customer-facing chat process flow, shown in this image, are as follows:

1. A customer, looking for an answer to a problem, requests a chat with an agent from any non-Siebel self-service Web application.

   The customer can request a chat as an anonymous user, or can log in to the self-service Web application first and then request a chat (for example, as a follow up to a service request). In both cases, the customer can do one of the following to request a chat:

   - Click Contact Us, then Chat (or navigate to Support, Contact Us, and then Chat).
   - Click the *Chat with a Live Agent* shortcut in the Need more Help? frame.

2. The customer fills out the chat request form by entering personal information (name or user ID, email address, address), product type details (product name or product category), problem summary, problem description, then clicks Submit and waits in the queue. For identified customers, the self-service Web application prepopulates as many of these values as possible. An example chat request form is shown in the image in *Chat Request Form*.
3. Using the information on the chat request form as search parameters, the self-service Web application executes a search against the Oracle Knowledge Base, and returns a list of suggested knowledge base articles and solutions for the customer to review.

   Customers can review these articles without losing their position in the queue.
4. If still not satisfied, then the customer clicks Chat and waits to be connected to an agent.

   Based on configuration, pushed URLs are visible to the customer in one of the following ways:

   - In a frame alongside the customer's chat UI in the same browser, or
   - In a new browser window

   For more information about push URL configuration and chat project setup, see *Configuring the Behavior of the Push URL Functionality*.
5. When the customer is connected to an agent, the customer engages in a chat conversation with the agent.
6. The customer releases the chat session.

   If configured for Siebel Chat, then when a customer releases a chat session, the customer can:

- Be prompted to fill out a feedback form
- Receive an email from the agent with the chat transcript attached

3 Setting Up Siebel Chat

Setting Up Siebel Chat

This chapter outlines how to set up Siebel Chat, including what aspects to consider when planning your setup and deploying your application. The installation guidelines refer to Siebel Call Center components. This chapter includes the following topics:

- *Installing Customer-Facing Components*
- *Roadmap for Configuring Siebel Call Center for Siebel Chat*
- *Required Parameter Configuration When Using Siebel Chat on Oracle Solaris*

Installing Customer-Facing Components

The self-service Web application that customers use can be any third-party or in-house Web application to which Siebel Chat is integrated. For information about installing non-Siebel self-service Web applications, see the documentation that accompanies that product.

After you have installed your self-service Web application, you must set it up for Siebel Chat. For more information about setting up Siebel Chat for non-Siebel self-service Web applications, see the documentation that accompanies that product.

Roadmap for Configuring Siebel Call Center for Siebel Chat

You must log in with administrator access to Siebel Call Center to configure it for Siebel Chat. To configure Siebel Call Center for Siebel Chat, perform the following tasks and process:

- *Verifying Organization Access Control*
- *Setting Up Agent Responsibilities and Views in Siebel Call Center*
- *Configuring Communications in Siebel Call Center for Siebel Chat*
- *Adding Agents to the Communications Configuration for Siebel Chat*
- *Setting Up an Alias for Siebel Chat Users in Siebel Call Center*
- *Setting Up Frequently Used Text in Siebel Call Center for Siebel Chat*
- *Configuring the Communications Toolbar*
- *Setting Up a Global URL Library in Siebel Call Center for Siebel Chat*
- *Configuring the RedoPopupInMenu Command*
- *Setting Up Outbound Email Communications in Siebel Call Center*
- *Configuring Support for F5 Refresh in Siebel Chat*

- *Setting Up System Preferences in Siebel Call Center for Siebel Chat*
- *Configuring Siebel Chat Auto Answer*
- *Customizing the Siebel Chat User Interface*
- *Process of Configuring Siebel Chat Feedback*

Verifying Organization Access Control

Siebel Chat supports multitenancy, so access to data in general is restricted to those records belonging to the agent's Position Organization. The Position Organization is the organization to which the agent's current position belongs. To verify an agent's position and position organization, complete the steps in the following procedure.

This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To verify an agent's position and position organization

1. Navigate to the Administration - User screen, then the Employees view.
2. From the Employees list, select or query for the agent whose position and position organization you want to verify.
3. Scroll to the end of the page. Notice the agent's Position (for example, Siebel Administrator) and Position Organization (for example, Default Organization).

   Only records belonging to the Position Organization are displayed for this agent. For more information about organization access control, see *Siebel Security Guide* .

Setting Up Agent Responsibilities and Views in Siebel Call Center

To handle chat interactions, each user of Siebel Chat must be given access to certain views and responsibilities. By default, the Universal Agent responsibility in Siebel Call Center is already set up for Siebel Chat. However, if you want to use a different responsibility (other than Universal Agent), then you must set up a new agent responsibility in Siebel Call Center, and add the appropriate chat views and users to that responsibility.

This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To set up agent responsibilities and views in Siebel Call Center for Siebel Chat

1. Navigate to the Administration - Application screen, then the Responsibilities view.

   A list of responsibilities displays with the responsibilities tab open.
2. Click New to create a new universal agent responsibility, and complete the fields as described in the following table.

Name	Description	Example Value
Responsibility	Enter a name for the universal agent responsibility that you want to create.	Universal Agent B2B+B2C
Description	Enter a description for the responsibility.	Siebel Universal Agent

Name	Description	Example Value
Organization	Click the single select button, and select the organization to which the responsibility belongs.	Active Systems

3. In the Views section of the page, click Add and add the following views to your agent responsibility:
    - Chat View
    - Chat Activity View
    - Chat Language Mapping View
    - Chat Parameter and Routing Administration View
    - Chat Popup Base View
    - Service Request Detail View
    - All Service Requests List View
    - Persistent Customer Dashboard View
    - Contact Details View
    - Contact Details View, Detail Tab
    - Visible Contact List View
    - Solution Resolution Documents View
    - Activity Attachment View
    - Search View
    - Basic Search View
    - Basic Search Results View
    - Advanced Search View
    - Advanced Search Results View
    - Search Selection View
    - Search Looking View
4. In the Users section of the page, click Add and add your chat users to the agent responsibility.

Configuring Communications in Siebel Call Center for Siebel Chat

To configure communications in Siebel Call Center for Siebel Chat, you must configure a communications driver, add agents to the chat communications configuration, configure chat routing, and finally set up an email profile for your chat communications so that an email is sent to users when they release a chat session.

For third-party chat integrations, you must configure a custom communications driver for the third-party chat solution. The purpose of this driver is to provide a mechanism through which chat messages are exchanged with the third-party chat server. For more information about how to create and configure a custom driver for third-party products using the Adaptive Communications API, see Siebel CTI Administration Guide.

This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To configure communications in Siebel Call Center for Siebel Chat

1. Locate the multichannel_enhanced_chat.def file as required.

   You must use the multichannel_enhanced_chat.def file if you are using the default workflow to handle incoming chat requests. For more information, see *Chat Standard Accept Process Workflow*.

   The communication file is delivered with your Siebel Call Center software, and it defines all the configuration details (parameters, profiles, commands, event handlers) for your application. Communication files are usually located in the following directory:

   ```
   SIEBSRVR_ROOT\bin\language_code
   ```

   If you cannot locate the communication file, then contact your system administrator or Oracle account support representative.

   > **Note:** You can optionally make a copy of multichannel_enhanced_chat.def and modify it using a text editor before importing it into your communications configuration record.

2. Configure chat communications and a communications driver for your company:

   a. Navigate to the Administration - Communications screen, then the All Configurations view.
   b. Click New to set up a new configuration record, enter a name for the record, enter a comment (this is optional), then save the record by clicking Ctrl+S.
   c. Import the configurations defined in the multichannel_enhanced_chat.def file:

      - Select the configuration record that you created, click Import Configuration, then Next.
      - On the Import Configuration window, specify the communications configuration elements to import. Choose all of the following options: Configuration Parameters, Drivers and Profiles, Commands, and Events.
      - Browse to the location of your multichannel_enhanced_chat.def file, and click OK.

        When the import process is complete, the Import Configuration window closes, and the Parameters, Profiles, Commands, and Event Handlers tabs are now populated with communication configuration information. For example, clicking the Event Handlers tab displays a list of available chat-related event handlers (such as, ChatAccepted, ChatMessageReceived, ChatSuspended, and so on). You can add or delete Parameters, Profiles, Commands, and Event Handlers, as required.

   d. Configure a driver for your company to communicate with the third-party chat server:

      - Select the configuration record that you created, and click the Profiles tab.
      - Click the Driver name.

        > **Note:** If you want to configure a new driver to communicate with the third-party chat server, then click New, select a communication profile from the list that is displayed, and then click Driver name.

      - On the Communications Drivers screen, use the vertical navigation buttons to navigate to the Profile Parameter Overrides section, and set the driver parameters, as required.

Note: If no profile exists for the driver, then one must be created. If one does exist, but it is not the profile that you want to configure, then add a new profile now.

The following table briefly describes some important driver parameters that can be configured. For more information about all driver parameters that can be configured, see Siebel CTI Administration Guide.

Driver Parameter	Value
Driver:CompanyName	Required. Set to the company alias.
Driver:AdminUserName	Required. Set to the administrator name.
Driver:AdminPassword	Required. Set to the administrator password (of AdminUserName).
Driver:URL	Required. Set to the location of the third-party chat services, for example, as follows: `http://east/<tp_chat_server>/services/`
Driver:ProxySupport	Optional. Set to script if you want to use wpad.dat to access third-party chat services using an HTTP proxy. Set to Disabled if you do not want to use a proxy.
Driver:ProxyScriptLocation	Optional. If Driver:ProxySupport is set to script, then set this parameter to the location of the script, for example, as follows: `http://wpad.us.oracle.com/wpad.dat`
Driver:EnableSOAPTrace	Optional. Set to True to log all communications between the driver and the third-party chat server, and set to False to ignore all communications.
Driver:LogFile	Optional. Specify a name for the driver log file.
Driver:LogSOAPFile	Optional. Specify a name for the log file, into which all messages between the driver and the third-party chat server are written.

3. Add agents to the chat communications configuration record.

   This step involves mapping an agent's details as defined for the third-party chat solution to the corresponding agent user in the Siebel Call Center application. For more information about adding agents, see *Adding Agents to the Communications Configuration for Siebel Chat*.
4. Save the chat configuration record if required so that you can readily access it in the future, for example, if you want to update and amend it.

5. (Optional) If changes have been made to the configuration, driver parameters, or profile parameters, then save the communications configuration to another file as follows:
    a. Navigate to the Administration - Communications screen, then the All Configurations view.
    b. Select the chat configuration record that you want to save, and click Export Configuration.
    c. On the Export Configuration window, specify the communications configuration elements to export. Choose from the following options:
        - Configuration Parameters
        - Drivers and Profiles
        - Commands
        - Events
    d. When prompted, save the communications configuration file to a suitable location.
6. Configure chat routing parameters.

    This step involves configuring the routing for chat data, see *Setting Up an Alias for Siebel Chat Users in Siebel Call Center*.
7. Configure an email profile for chat communications, see *Setting Up Outbound Email Communications in Siebel Call Center*.

Adding Agents to the Communications Configuration for Siebel Chat

When adding agents to the communications configuration for Siebel Chat in Siebel Call Center, it is important that you use the same Agent Login and Password that were set up for the agent in the third-party chat server.

To add agents to the communications configuration set up in Siebel Call Center, complete the steps in the following procedure. This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To add agents to the Siebel Chat communications configuration in Siebel Call Center

1. Navigate to the Administration - Communications screen, then the All Configurations view.
2. Select the (chat) communications configuration record for which you want to add agents, and click the Agents tab.
3. Click New to add an agent.
4. On the Add Agents window, select or query for the agent that you want to add, and then click OK.
5. Specify the Agent Login and Password that the agent must use to log in to the third-party chat server:
    a. Click on the agent name to go to the Agent General Profile page.
    b. Enter the Login and Password in the Agent Login and Password fields respectively.
6. Assign a teleset to each agent:
    a. Click an Agent, then the Telesets tab, then New.
    b. On the Add Telesets window, select or query for the teleset that you want to assign, and click OK.

For information about configuring agents after an agent has been added to a communications configuration, see Siebel CTI Administration Guide.

Setting Up an Alias for Siebel Chat Users in Siebel Call Center

For each Siebel Chat user, it is important to set the Alias value on the Person record within Siebel Call Center to the corresponding First Name value provided for the user account in the third-party chat server. This is the name that the customer requesting the chat sees in the chat transcript, and you want it to be the same in the agent transcript. If no alias value is specified in Siebel Call Center, then the user's log in name is used in the transcript that is emailed to the customer, and it might not match the name that the customer sees during the chat session.

This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To set the alias in Siebel Call Centerfor Siebel Chat users

1. Navigate to the Administration - User screen, then the Persons or the Employees view.
2. Select the person record for which you want to set an alias.
3. Enter an alias value in the Alias field.

> **Note:** The alias value that you enter in this field for the user must correspond to the First Name value that is defined for the user account in the third-party chat server.

Setting Up Frequently Used Text in Siebel Call Center for Siebel Chat

Administrators can set up a library of message response templates (or frequently used text) for chat in Siebel Call Center where each template holds a predefined message response, which agents can choose to send to customers as required. When responding to chat requests, agents can access this library of predefined message responses by enabling Frequently Used Text (FUT) matching in the message input area of the chat pane. For more information about responding to chat requests, see *Responding to an Incoming Chat*.

To set up predefined message responses for chat in Siebel Call Center, complete the steps in the following procedure. This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To set up a library of frequently used text templates in Siebel Call Center for Siebel Chat

1. Navigate to the Administration - Communications screen, then the All Templates view.
2. Click New to create a FUT template and on the Simple tab, specify the following:

Field	Description
Name	The name of your text template.
Channel Type	Select Chat as the channel type from the drop-down list.
Template Type (Optional)	Select Greeting, Body, or Closing as required:

Field	Description
	○ A *Greeting* template type response can be: Hello. How may I help you? ○ A *Body* template type response can be: Good morning. How can I help you? ○ A *Closing* template type response can be: Thank you. Good bye.
Language	Specify the language to be used in your text template.
Locale	Specify the Locale Code for your country.
Public (Optional)	Select this check box if you want the template to be made available to everyone outside the creator's organization. **Note:** This check box is selected by default for chat.
Pick Available Substitutions (Optional)	Select the Chat UI substitution object from the object drop-down list if you want to use any of the available Chat UI data substitutions when composing your text response. The available Chat UI data substitutions are: ○ [Chat UI.Customer First Name] ○ [Chat UI.Customer Last Name] ○ [Chat UI.Employee Alias] ○ [Chat UI.InputText] ○ [Chat UI.Primary Owner Id]

3. Enter the text for your template into the message input area; the subject line is not used for templates.

Configuring the Communications Toolbar

This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*. The communications toolbar displays the functions related to chat that are available for Siebel Chat. *Working with Siebel Chat* describes the chat-related behavior of the functions on the communications toolbar. The communications toolbar is defined and configured within Siebel Tools, like other toolbars in Siebel applications. For information about configuring the communications toolbar in Siebel Tools, including object definitions, toolbar items, commands, and bitmaps, see the following:

- *Configuring Siebel Business Applications*
- *Siebel Developer's Reference*

For information about modifying the communications toolbar, including modifying the appearance of existing toolbar buttons, and moving, adding, or removing a toolbar button, see *Siebel CTI Administration Guide* .

Setting Up a Global URL Library in Siebel Call Center for Siebel Chat

Administrators can make a list of URLs, save it, and store it in a URL library so that an agent can push the URLs to users when required. In Siebel Call Center, resolution documents of type URL are used to populate the URL library list.

To set up a URL library for chat in Siebel Call Center, complete the steps in the following procedure. This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.



CAUTION: Ensure that URLs can be reached by users who are outside your company firewall.

To set up a URL library in Siebel Call Center for Siebel Chat

1. Navigate to the Administration - Resolution Documents screen.
2. Click New to add a new (URL) resolution to the library:
    a. In the File Name field, click the single select button to add a resolution attachment.
    b. Enter the URL address (starting with http://) of the URL that you want to share with customers.
    c. Click Add.

When handling chat interactions, agents can access a list of available URLs. For more information, see *Accepting an Incoming Chat* and *Responding to an Incoming Chat*.

Configuring the RedoPopupInMenu Command

If the RedoPopupInMenu command is configured in Siebel Call Center for Siebel Chat, then chat response messages that are not successfully accepted for a chat session are cached for the agent for that chat session. For example, when agents accept a new work item or resume work on an existing work item that has not been closed yet, any chat response messages that the agent has not successfully accepted for the chat session are cached provided that the RedoPopupInMenu command is configured. Agents can then replay the chat response messages that were cached by navigating to Tools, then Communications in Siebel Call Center, and clicking Redo Popup. Use the following procedure to configure the RedoPopupInMenu command in Siebel Call Center for Siebel Chat.

To configure the RedoPopupInMenu command

1. Navigate to the Administration - Communications screen, then the All Configurations view.
2. Query for the communications configuration record for which you want to configure the RedoPopupInMenu command, then click the Commands tab.
3. Go to the RedoPopupInMenu command.

    If the RedoPopupInMenu command does not exist, then click New to add it, and then click Save.
4. Drill down on the RedoPopupInMenu command by clicking the RedoPopupInMenu link and configure the command parameters as follows:

a. Click the Command Parameters tab.
b. Click New and add the command parameters for RedoPopupInMenu with the values shown in the following table.

Name	Value
Description	Redo Popup in menu
HotKey	Shift+Alt+R **Note:** Make sure that the keyboard shortcut does not conflict with any keyboard shortcuts that are defined already.
MenuPosition	15 **Note:** Change the menu position if it conflicts with the menu position of another command.
ServiceMethod	Communications Client.RedoPopup
Title	Redo Popup

5. Navigate to `WINDOWS\Downloaded Program Files`, and clear the Internet Explorer cache.
6. Log in to Siebel Call Center again, and test your configuration.

Setting Up Outbound Email Communications in Siebel Call Center

If the Chat: Auto Email Mode system preference is set to POPUP or SEND, then outbound email communications must be set up for Siebel Chat so that an email response is correctly sent to customers when chat sessions are released. If Chat: Auto Email Mode is set to NONE, then setting up chat outbound communication in Siebel Call Center is optional.

Use the following procedure to set up outbound email communications in Siebel Call Center. This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To set up outbound email communications in Siebel Call Center for Siebel Chat

1. Navigate to the User Preferences screen, then the Outbound Communications view.
2. Set up outbound communications as follows:

    a. In the Send Email section, set the following options, as required:

        - Default Profile
        - Default Recipient Class

- Email Client
- Siebel/Outlook Form
- Siebel/Lotus Form
- Default Message Format

> **Note:** Default Profile must be set to the email profile that is used to send emails to users (for example, when users release a chat session). You must set up this email profile first. For more information about email profile setup, see Siebel Email Administration Guide.

b. In the Email Response section, set the following options, as required:

- Default Greeting Template: Options include Formal Greeting, and Informal Greeting.
- Default Closing Template: Options include Address Verification Closing, Formal Info Closing, Formal Support Closing, Informal Info Closing, and Informal Support Closing.
- Default Message Format
- Include Original Message in Reply
- Remain on Same View After Send (Cancel)

c. In the Advanced Features section, set the following options, as required:

- Language
- Locale

d. Set the following system preferences as shown in *Setting Up System Preferences in Siebel Call Center for Siebel Chat*:

- Chat: Default Email Template. Set this to the name of the (chat) email profile that you set up in Siebel Call Center.
- Chat: Auto Email Mode. Set this to POPUP, SEND, or NONE.

For more information about each of these options, about setting up outbound email communications, and configuring the Internet SMTP/POP3 server driver, see Siebel Email Administration Guide.

Configuring Support for F5 Refresh in Siebel Chat

To configure support for F5 refresh in Siebel Chat, you must remove the ApplicationUnload run-time event. After the ApplicationUnload run-time event is removed, you can use the F5 refresh function key during chat sessions without losing any focus.

> **Note:** Remove the run-time event record only. Do not remove the Action Sets record.

To configure support for F5 refresh in Siebel Chat

1. Navigate to Administration - Runtime Events, then the Events view.
2. Query for an event record with the following values, and if the record is present, then delete it:
   - Event name: ApplicationUnload
   - Action Set Name: Comm. Web Session Close Action
   - Object Type: application

3. Restart the Siebel Server for the changes to take effect.

   Now when an agent accepts a chat and starts exchanging chat messages with a customer, the chat pane and service request view are open, and if the agent then selects F5 on the keyboard or Refresh on the browser toolbar, the Siebel application refreshes. But after the refresh, the Siebel application remains the same, the chat session is still connected, the chat pane and service request view are open, and the agent can continue the chat conversation with the customer.

Setting Up System Preferences in Siebel Call Center for Siebel Chat

There are a number of system preferences that you can optionally set in Siebel Call Center for Siebel Chat. The following procedure describes these system preferences and how to set them. This task is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

To set the system preferences in Siebel Call Center for Siebel Chat

1. Navigate to the Administration - Application screen, then the System Preferences view.
2. Query for **Chat** to get a list of all the chat-related system preferences that can be set up.

   For each Siebel Chat system preference that you want to set up, specify the system preference name, value, and description, as defined in the following table.

System Preference Name	System Preference Value	Description
Chat: Default Email Template	**Note:** You must define the default email template first, and then specify it in this field. For more information, see *Setting Up Outbound Email Communications in Siebel Call Center*.	Name of the default email template used to automatically generate the email that is sent when a user releases a chat session.
Chat: AutoCreate Service Req	FALSE	Flag that determines whether a service request is created automatically when agents accept a chat request. Valid values are TRUE or FALSE.
Chat: Auto Search Mode	ON	Determines the mode used to automatically perform a search when agents accept a chat request. Valid values are ON or OFF.
Chat: Auto Email Mode	SEND	Determines the mode used for automatically sending an email when

System Preference Name	System Preference Value	Description
		a chat is terminated. Valid values are POPUP, SEND, and NONE: o POPUP automatically displays the email, allowing the agent to modify it before sending. o SEND automatically sends the email, without prompting the agent. o NONE does not send an email.
Chat: BlinkTab	ON	Determines whether the chat tab blinks (ON) or not (OFF) when a new chat is accepted. For more information, see *Configuring the Siebel Chat BlinkTab System Preference*.
Chat: CustomerTypeDisplayTime	5	The time interval in seconds in which to continue showing the *customer is typing* message in the chat pane. The value must be greater than zero. For example, if Chat: CustomerTypeDisplayTime is set to 5 when a customer starts typing, then the message shows for 5 seconds in the chat pane. When the customer stops typing, the *customer is typing* message in the chat pane disappears 5 seconds later.
Chat: AgentTypeDetectInterval	5	The time interval in seconds after which the system checks if the agent is typing a message. The value must be greater than zero.
Chat: SendAgentTypeMsgCmd	SendAgentTypingMsg	The CTI command to call when sending the agent's message response to a customer. The CTI command must be a valid command when configuring in CTI.

Required Parameter Configuration When Using Siebel Chat on Oracle Solaris

The following parameters must be configured appropriately if using Siebel Chat on Oracle Solaris: CommSessionMgr, EnterpriseServer, GatewayAddress, and RequestServer.

To configure the required parameters if using Siebel Chat on Oracle Solaris

1. Navigate to the Administration - Communications screen, then the All Configurations view.
2. Select the communications configuration record that you want to modify, then click the Parameters tab.
3. Configure the parameters as shown in the following table:

Parameter Name	Value
CommSessionMgr	CommSessionMgr
RequestServer	[Windows Server where CommMgmt is running]
GatewayAddress	[Gateway address of the computer where CommMgmt is running : gateway port]
EnterpriseServer	[Siebel]

4 Working with Siebel Chat

Working with Siebel Chat

This chapter describes how an agent works with Siebel Chat. This chapter includes the following topics:

- *Siebel Chat Interface*
- *Processing Siebel Chat Requests*
- *Verifying That Siebel Chat Is Installed for Siebel Call Center*
- *Reviewing Inbound Siebel Chat Activities*
- *About the Communications Toolbar*
- *Logging in to Siebel Chat*
- *Receiving a New Chat*
- *Accepting an Incoming Chat*
- *Responding to an Incoming Chat*
- *Handling Siebel Chat Sessions*
- *About the Chat Dashboard*
- *Changing Agent Status*
- *Agent and Siebel Chat Interaction States*
- *Setting Up Logging and Tracing for Siebel Chat Components*
- *About Web Notifications*

Siebel Chat Interface

The Siebel Chat interface (or chat pane) is a part of the communications panel.

You access the communications panel by clicking Communications on the menu toolbar in your client application. The communications panel contains the following elements: a header area, the communications toolbar, the customer dashboard, and the chat pane. Depending on the deployment of your client application, the communications panel might not contain all of these elements.

The communications panel appears on either side of your Siebel Call Center application, depending on configuration. The main Siebel Call Center application area appears next to the communications panel.

The communications panel, which includes the Siebel Chat pane, can appear in a docked position or as a floating pane in your client application. You can dock or undock the panel by clicking Dock or Undock. When the communications panel is floating (undocked), you can move and position it anywhere within the confines of the main application client area. In addition, you can do the following:

- Control the size of the communications panel by choosing a wide or a narrow display for the panel.
- Modify the display of the communications toolbar by choosing to display more or less of the toolbar.
- Customize the appearance of the communications toolbar.

To do this, customers must create their own customized CSS file in which custom icons are specified and then associate this CSS file to the application theme for which these customizations are to be enabled. For more information about performing this type of customization, see *Configuring Siebel Open UI* .

- Developers can modify the display of the communications panel and its elements.

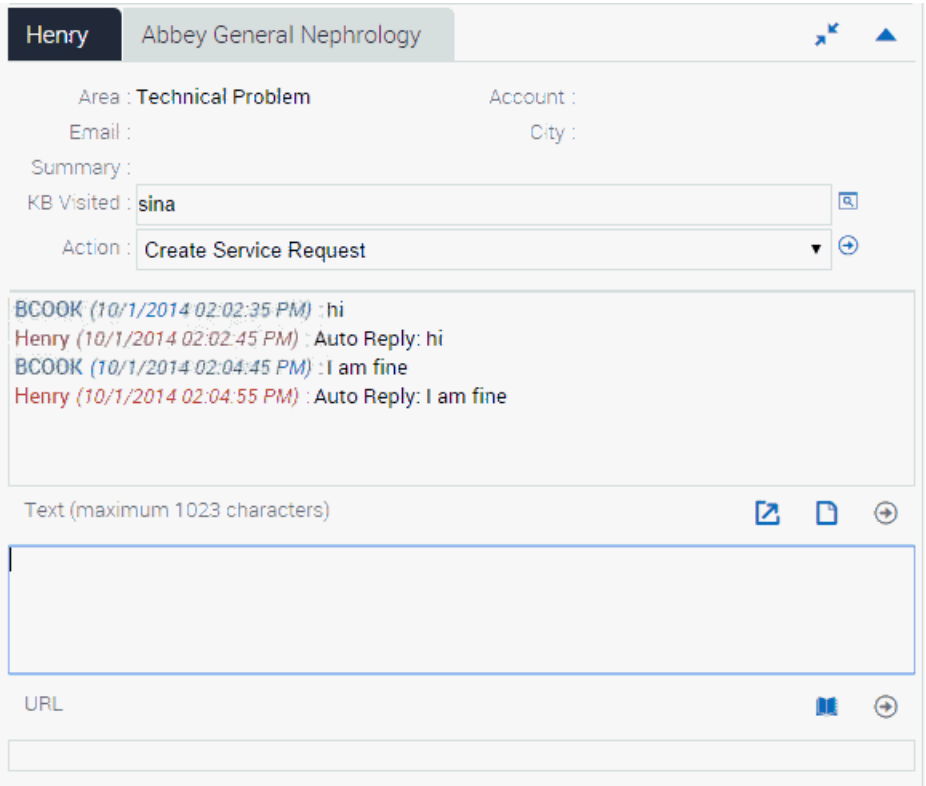
  For example, by modifying the applicable style sheet, developers can modify which items appear when an agent displays more or less of the toolbar. Or, by modifying the Siebel Web template file CCFrameContentHI in the database, developers can modify the order of the elements within the communications panel.

- Close the communications panel by clicking Close (X).

For more information about the communications panel and the communications toolbar, see *Siebel CTI Administration Guide* . For more information about the chat pane and customer dashboard, see *Accessing Siebel Chat in Siebel Call Center* and *About the Chat Dashboard*.

Agents can engage in multiple chat sessions with many users simultaneously. If more than one chat session is open, the chat sessions are represented by tabs in the chat pane. For more information, see *Handling Multiple Simultaneous Chat Sessions*.

The following image shows the Siebel Chat interface or chat pane.



The various sections of the chat pane show the following:

- Dashboard. The dashboard displays details about the problem that the customer is having. Agents can carry out a number of popular actions quickly using the chat dashboard. For more information about the chat dashboard, see *About the Chat Dashboard*.

- Transcript area. This is where agents can view the transcript of the chat conversation so far. Each part of the chat conversation is clearly labelled with either the agent's or the customer's name, is dated, and time stamped. When a chat session ends, the chat transcript is attached to an activity. For more information about accessing chat transcripts and attachments, see *Reviewing Inbound Siebel Chat Activities*.
- Input text box. This text box is where an agent enters the responses to a chat request, executes a search, and smart shares service request and solution information. For more information about these tasks, see *Responding to an Incoming Chat*.
- URL area. This is where an agent can enter a Web URL to share with a customer, which is then automatically displayed in the customer's browser. To do this, enter the URL, and then press enter to push the URL to the customer. This eliminates the need for the customer to manually copy and paste the URL into a browser window. Agents can also click the URL Library button to access a global list of predefined URLs, which they can push to customers as required. For more information about push URL, see *Pushing a Web URL*.

Processing Siebel Chat Requests

The processing of Siebel Chat requests involves identifying the customer requesting a chat, then creating one or more activity records linked to a contact record or a service request. An activity is always created when a chat is accepted by an agent, and each time a chat is transferred.

This topic includes information about the following:

- *Chat Standard Accept Process Workflow*
- *Siebel Call Center Behavior When Processing Siebel Chat Requests*
- *Guidelines for Processing Siebel Chat Requests*

Chat Standard Accept Process Workflow

Siebel Chat provides a sample workflow called Chat Standard Accept Process - vx to handle incoming chat requests, where vx represents the version number. Currently, the latest version is v3. The Chat Standard Accept Process workflow shown in the image in this topic is the default workflow that is called when an incoming chat request is accepted by an agent. When an agent accepts an incoming chat, Siebel Call Center attempts to identify the customer requesting the chat from the information provided on the chat request form. This information determines whether a service request is created, and consequently the application view that the agent is directed to in the Siebel Call Center application.

The Chat Standard Accept Process workflow provides the following enhanced chat functionality to handle incoming chat requests:

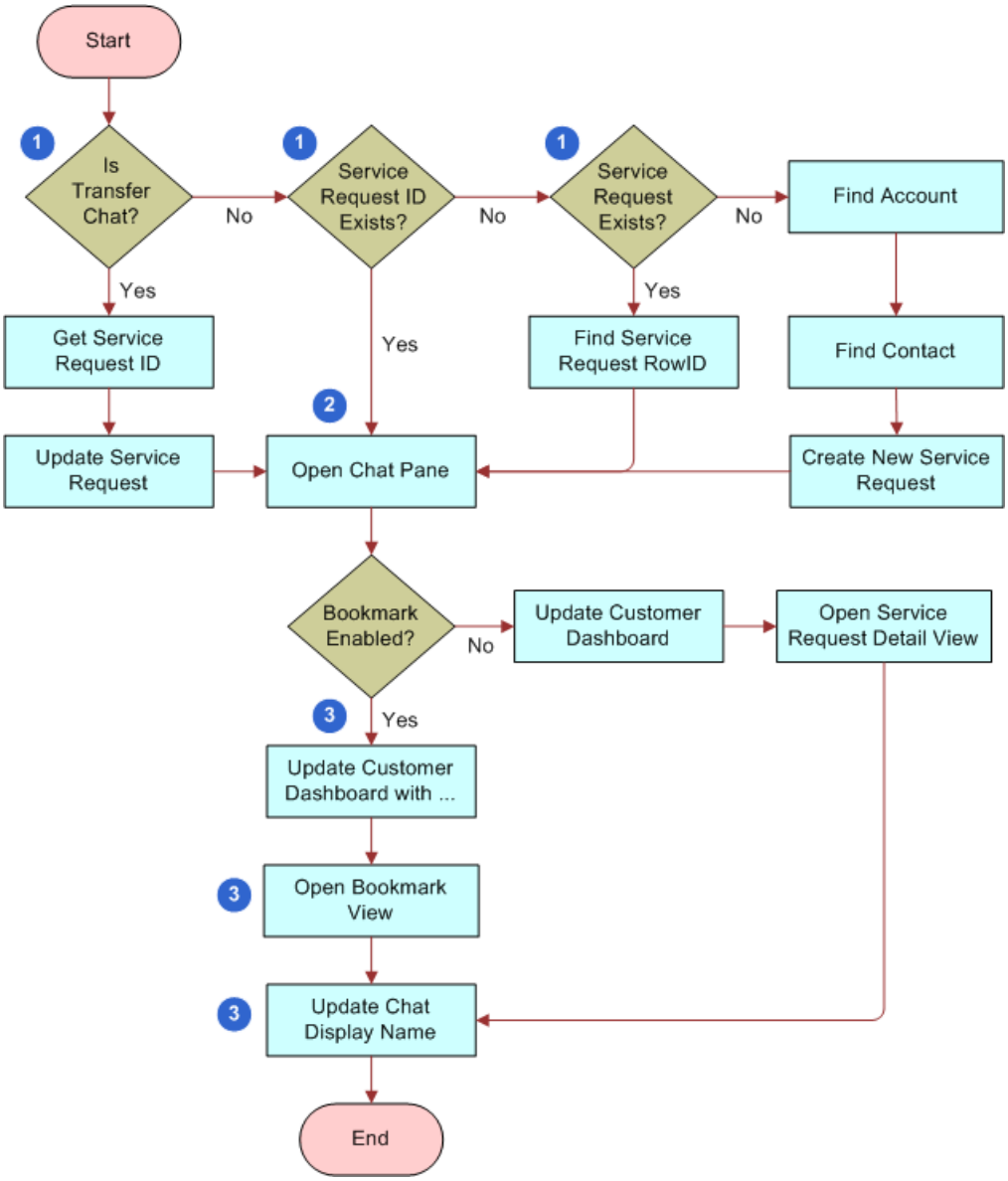
- Configuring which parameters to pass from the chat request page
- Configuring the chat accept process using Siebel Workflow
- Configuring the RedoPopupInMenu command
- Siebel Chat transfer to agent and workgroup
- Siebel Chat bookmark transfer
- Siebel Chat auto answer for agents
- Configuring the display name for Siebel Chat sessions

For more information about how to configure and set up this enhanced chat functionality, see the following:

- *Configuring Siebel Chat Auto Answer*

- *Customizing Siebel Chat*

The Chat Standard Accept Process workflow shown in the following image is a Siebel Workflow, and you can configure the workflow as required using Siebel Tools to suit your own business needs.



Workflow Description. This workflow performs the following actions:

1. When an agent accepts a chat, the chat UI business service does one of the following:
    - Obtains the service request ID if the chat is a transferred chat.
    - Identifies the service request ID if one is associated with the incoming chat.
    - Identifies the service request number if one is associated with the incoming chat.

- Identifies the account or contact if the customer provides a valid contact name or user ID and account name, then it creates a new service request.

2. The chat UI business service opens the chat pane.
3. The chat UI business service directs the agent to the appropriate view (service request detail view or bookmark view) in the Siebel Call Center application, and the chat display name is updated accordingly in the chat pane.

   For example:

   - If bookmark is enabled, then the customer dashboard is updated, the bookmarked view opens, and the chat display name is updated in the chat pane.
   - If bookmark is disabled, then the customer dashboard updates, the service request detail view opens, and the chat display name updates accordingly in the chat pane.

Siebel Call Center Behavior When Processing Siebel Chat Requests

The tables in this topic summarize the default workflow for accepting Siebel Chat requests by describing what happens in Siebel Call Center when various conditions are met (or not, as the case might be). Depending on the information that the customer provides on the chat request form, and the ability of Siebel Call Center to identify that customer, the workflow for accepting an incoming chat can vary.

The following table describes what happens in Siebel Call Center when the user's contact ID only is identified, along with the presence or absence of various other conditions.

<table>
<tr><th>SR ID Present</th><th>Contact ID Present</th><th>Auto Create SR On</th><th>Identify Chat Data Present</th><th>Other Chat Data Present</th><th colspan="3">Actions in Siebel Call Center Application</th></tr>
<tr><td>No</td><td>Yes</td><td>Yes</td><td>Yes</td><td>Yes</td><td rowspan="2">Creates a new service request using the contact ID and other chat data present, such as, area, subarea, or problem description.

Shows the service request detail view.

Opens the search frame and searches using other chat data present, such as, problem description or product category.</td><td rowspan="4">Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.</td><td rowspan="4">Creates a new activity with the contact ID.

Updates the activity with the service request ID when the service request is created.</td></tr>
<tr><td>No</td><td>Yes</td><td>Yes</td><td>No</td><td>Yes</td></tr>
<tr><td>No</td><td>Yes</td><td>Yes</td><td>Yes</td><td>No</td><td rowspan="2">Creates a new service request using the contact ID.

Shows the service request detail view.</td></tr>
<tr><td>No</td><td>Yes</td><td>Yes</td><td>No</td><td>No</td></tr>
</table>

SR ID Present	Contact ID Present	Auto Create SR On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application		
					Opens the search frame with no search criteria.		
No	Yes	No	Yes	Yes	Shows the contact detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the contact ID.
No	Yes	No	No	Yes	Opens the search frame and searches using other chat data present.		
No	Yes	No	Yes	No	Shows the contact detail view.		
No	Yes	No	No	No	Opens the search frame with no search criteria.		

The following table describes what happens in Siebel Call Center when the user's service request ID and contact ID are both identified, along with the presence or absence of various other conditions.

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
Yes	Yes	Yes	Yes	Yes	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.
Yes	Yes	Yes	Yes	No	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.
Yes	Yes	Yes	No	Yes	Shows the service request detail view.	Populates the customer	Creates a new activity with the

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
						dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	service request ID, and contact ID.
Yes	Yes	Yes	No	No	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.
Yes	Yes	No	Yes	Yes	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.
Yes	Yes	No	Yes	No	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.
Yes	Yes	No	No	Yes	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
Yes	Yes	No	No	No	Shows the service request detail view.	Populates the customer dashboard using the contact ID, and identify chat data (if any), such as, contact first name, contact last name, email address, or phone number.	Creates a new activity with the service request ID, and contact ID.

The following table describes what happens in Siebel Call Center when the user's service request ID only is identified, along with the presence or absence of various other conditions.

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
Yes	No	No	Yes	Yes	Shows the service request detail view.	Searches for the contact using the service request ID and identify chat data (if any), such as, problem description or product category. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID when the contact is determined.
Yes	No	No	Yes	No	Shows the service request detail view.	Searches for the contact using the service request ID and identify chat data (if any), such as, problem description or product category. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID when the contact is determined.
Yes	No	No	Yes	Yes	Shows the service request detail view.	Searches for the contact using the service request ID and identify chat data (if any), such as, problem	Creates a new activity with the service request ID. Updates the activity with the contact ID

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
						description or product category. Populates the customer dashboard.	when the contact is determined.
Yes	No	No	Yes	No	Shows the service request detail view.	Searches for the contact using the service request ID and identify chat data (if any), such as, problem description or product category. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID when the contact is determined.
Yes	No	Yes	No	Yes	Shows the service request detail view.	Searches for the contact using the service request ID. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID when the contact is determined.
Yes	No	Yes	No	No	Shows the service request detail view.	Searches for the contact using the service request ID. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID when the contact is determined.
Yes	No	No	No	Yes	Shows the service request detail view.	Searches for the contact using the service request ID. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID when the contact is determined.
Yes	No	No	No	No	Shows the service request detail view.	Searches for the contact using the service request ID. Populates the customer dashboard.	Creates a new activity with the service request ID. Updates the activity with the contact ID

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
							when the contact is determined.

The following table describes what happens in Siebel Call Center when neither the user's service request ID nor the contact ID are identified, but various other conditions are either present or absent.

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
No	No	Yes	Yes	Yes	Opens the search frame and searches using other chat data present, such as, problem description or product category. Searches for the contact using identifying chat data present, such as contact first name, contact last name, email address, or phone number. Creates a new service request (with the contact if one is found, or without a contact). Shows the service request detail view.	If one contact is found, populates the customer dashboard using the contact information and identifying chat data. If no contact could be identified, populates the customer dashboard with the identifying contact data, and flags it as an unknown contact, such as, contact email address or contact phone number.	Creates a new activity with the service request ID and contact ID (if found).
No	No	Yes	Yes	No	Opens the search frame with no search criteria. Searches for the contact using the identifying chat data present, such as contact first name, contact last name, email address, or phone number. Creates a new service request (with the contact if one is	If one contact is found, populates the customer dashboard using the contact information and identifying chat data. If no contact could be identified, populates the customer dashboard with the identifying contact data, and flags it as an unknown contact, such as, contact email address or contact phone number.	Creates a new activity with the service request ID and contact ID (if found).

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
					found, or without a contact). Shows the service request detail view.		
No	No	Yes	No	Yes	Opens the search frame, and searches using other chat data present, such as, problem description or product category. Creates a new service request without the contact information. Shows the service request detail view.	This is an anonymous chat, where the customer dashboard is not populated with any information.	Creates a new activity with the service request ID.
No	No	Yes	No	No	Opens the search frame, with no search criteria. Creates a new service request without the contact information. Shows the service request detail view.	This is an anonymous chat, where the customer dashboard is not populated with any information.	Creates a new activity with the service request ID.
No	No	No	Yes	Yes	Opens the search frame and searches using other chat data present, such as, problem description or product category. Searches for the contact using identifying chat data present, such as contact first name, contact last name, email address, or phone number. Shows the contact detail or contact list view.	If one contact is found, populates the customer dashboard using the contact information and identifying chat data. If a contact could not be identified, populates the customer dashboard with the identifying chat data, and flags it as an unknown contact, such as, contact email address or contact phone number.	Creates a new activity with the contact ID (if found).
No	No	No	Yes	No	Opens the search frame with no search criteria.	If one contact is found, populates the customer dashboard using the contact	Creates a new activity with the contact ID (if found).

SR ID Present	Contact ID Present	Auto Create Service Request On	Identify Chat Data Present	Other Chat Data Present	Actions in Siebel Call Center Application (1)	Actions in Siebel Call Center Application (2)	Actions in Siebel Call Center Application (3)
					Searches for the contact using the identifying chat data present, such as contact first name, contact last name, email address, or phone number. Shows the contact detail or contact list view.	information and identifying chat data. If a contact could not be identified, populates the customer dashboard with the identifying chat data, and flags it as an unknown contact, such as, contact email address or contact phone number.	
No	No	No	No	Yes	Opens the search frame, and searches using other chat data present, such as, problem description or product category.	This is an anonymous chat, where the customer dashboard is not populated with any information.	Creates a new activity without the contact ID.
No	No	No	No	No	Opens the search frame with no search criteria.	This is an anonymous chat, where the customer dashboard is not populated with any information.	Creates a new activity without the contact ID.

Guidelines for Processing Siebel Chat Requests

Consider the following guideline for processing chat requests:

- Set the system preference Chat: AutoCreate Service Req to TRUE to automatically create a new service request for every incoming chat request that is accepted by an agent.

For more information about system preference setup for chat, see *Setting Up System Preferences in Siebel Call Center for Siebel Chat*.

Verifying That Siebel Chat Is Installed for Siebel Call Center

Before allowing agents to log in and start working with Siebel Chat, it is recommended that administrators verify the following:

- That Siebel Chat has been installed for Siebel Call Center. For more information, see *Accessing Siebel Chat in Siebel Call Center*.
- That all server component groups from Siebel Call Center are started. Server component groups must be started so that agents can receive inbound chats. For more information, see *Checking That All Server Component Groups are Started for Siebel Call Center*.

Accessing Siebel Chat in Siebel Call Center

The Siebel Chat interface (or chat pane) is a part of the communications panel. The following procedure shows you how to access the chat pane in Siebel Call Center.

To access Siebel Chat in Siebel Call Center

- Click Communications on the menu toolbar in Siebel Call Center.

  The communications panel appears containing a header area, the communications toolbar, the customer dashboard, and the chat pane. The menu toolbar is located on the application banner of the application window.

Checking That All Server Component Groups are Started for Siebel Call Center

To enable agents to receive inbound chats, administrators must ensure that all relevant server component groups are up and running for Siebel Call Center. If server component groups are not started, then administrators must start them from either Siebel Call Center or the command line, as described in the following procedures.

Starting Server Component Groups from Siebel Call Center

To start the server component groups from Siebel Call Center, complete the steps in the following procedure.

To start the server component groups from Siebel Call Center

1. Navigate to the Administration - Server Configuration screen, then the Enterprises view, and click the Component Groups tab.
2. Query for the following component groups one at a time, then click the Enable button and ensure that there is a check mark in the Enabled on Server? Component Group Assignment panel.
   - Communication Management
   - Workflow Management

- Enterprise Application Integration (EAI)

> **Note:** Without enabling the Communication Management, Workflow Management, and EAI component groups, Siebel Chat agents are not able to receive inbound chats.

3. Click the Synchronize tab, then the Synchronize button.
4. Restart the Siebel Server.

Starting Server Component Groups from the Command Line

Use the following procedure to start server component groups from the command line.

To start server component groups from the command line

1. Log in to the Siebel Server to connect to the application database:

```
SIEBSRVR_ROOT\bin >smgr
```

2. Enable the server component groups:

```
srvrmgr> enable compgrp CommMgmt
Command completed successfully.

srvrmgr> enable compgrp EAI
Command completed successfully.

srvrmgr> enable compgrp Workflow
Command completed successfully.

srvrmgr> exit
```

   For information about setting environment parameters in Windows and UNIX, see *Siebel System Administration Guide* .
3. Connect back to the application database:

```
SIEBSRVR_ROOT\bin>smgr
```

4. To synchronize:

```
srvrmgr>sync
```

5. Restart the Siebel Server.

Reviewing Inbound Siebel Chat Activities

Agents can review inbound chat activities, and inbound chat activities associated with individual contacts, and service requests.

- To review inbound chat activities, complete the steps in the following procedure.
- To review inbound chat activities associated with contacts and service requests, see the following:
  - *Reviewing Inbound Siebel Chat Activities Associated with Service Requests*

- *Reviewing Inbound Siebel Chat Activities Associated with Contacts*

- To review cached chat response messages, see *Replaying Cached Chat Response Messages*.

To review inbound chat activities

1. Navigate to the Activities - Activities List view.
2. Click Query and select Chat - Inbound from the Type drop-down list to display a list of inbound chat activities.
3. Click an inbound chat activity to open and review it. In the Chat - Inbound screen that opens:

   a. Go to the Chat Info tab, and review the chat-related information associated with the inbound chat activity.

      The following table describes the fields within the Chat Info tab.

Field Name	Description
Name Provided	User name.
Email Address Provided	User email address.
Phone Provided	User telephone number.
Product	The product that the user is using.
Problem Description	Description of the problem encountered by the user.
Area	The product area where the problem occurred.
Sub Area	The product subarea where the problem occurred.
SR Number	The service request number associated with this inbound chat activity, if any. Click SR Number to drill down on a detailed view of the service request.
SR Description	A brief description of the service request associated with this inbound chat activity.

   b. Go to the Attachments tab and click the chat_transcript file to open and review the entire transcript of the inbound chat activity.

      Click any other file attachment to open up, and review the solution content.

Reviewing Inbound Siebel Chat Activities Associated with Service Requests

Agents can review inbound chat activities associated with individual service requests. To review inbound chat activities associated with a service request, complete the steps in the following procedure.

To review inbound chat activities associated with a service request

1. Navigate to the Service screen, then the Service Requests List view.
2. Click or query for the service request that you want to review.
3. Go to the Activities tab, click Query, select Chat - Inbound from the Type drop-down list, then click Go to display the inbound chat activities associated with this service request.
4. Click an inbound chat activity to open and review it. In the Chat - Inbound screen that opens:
    a. Go to the Chat Info tab and review the chat-related information associated with the inbound chat activity. For a description of the fields within the Chat Info tab, see *Reviewing Inbound Siebel Chat Activities*.
    b. Go to the Attachments tab and click the chat_transcript file to open and review the entire transcript of the inbound chat activity. Click any other file attachment to open up, and review the solution content.

Reviewing Inbound Siebel Chat Activities Associated with Contacts

Agents can review inbound chat activities and solutions associated with individual contacts. To review inbound chat activities associated with a contact, complete the steps in the following procedure.

To review inbound chat activities associated with a contact

1. Navigate to the Contacts screen, then the Contacts List view.
2. Click or query for the contact that you want to review.
3. Go to the Activities tab, click Query, select Chat - Inbound from the Type drop-down list, then click Go to display the inbound chat activities associated with this contact.
4. Click an inbound chat activity to open and review it. In the Chat - Inbound screen that opens:
    a. Go to the Chat Info tab and review the chat-related information associated with the inbound chat activity. For a description of the fields within the Chat Info tab, see *Reviewing Inbound Siebel Chat Activities*.
    b. Go to the Attachments tab and click the chat_transcript file to open and review the entire transcript of the inbound chat activity. Click any other file attachment to open up, and review the solution content.

Replaying Cached Chat Response Messages

When agents accept a new work item or resume work on an existing work item that has not been finished yet, any chat responses that the agent has not successfully accepted for the chat session are cached provided that the RedoPopupInMenu command is configured. Agents can then replay the chat response messages that were cached as shown in the following procedure. After replaying the cache, the cache is cleared, and all cached chat response messages are removed for the chat session.

To replay cached chat response messages for a chat session

1. Select the active chat session that you want.
2. Navigate to Tools, then Communications in Siebel Call Center, and click Redo Popup.

Related Topic

Configuring the RedoPopupInMenu Command

About the Communications Toolbar

The communications toolbar in Siebel Call Center displays the functions related to chat that are available for Siebel Chat. Installing and setting up Siebel Chat updates the functionality of the communications toolbar to start supporting chat interactions.

The communications toolbar is defined and configured in Siebel Tools, like other toolbars in Siebel applications. For more information about configuring the communications toolbar and about the functions available on the communications toolbar, see *Siebel CTI Administration Guide* .

Using the communications toolbar, agents can sign in to the Siebel Chat channel and make themselves available for incoming chat requests, put chats on hold, transfer, and release chats. For more information about these tasks, see the following:

- *Logging in to Siebel Chat*
- *Receiving a New Chat*
- *Accepting an Incoming Chat*
- *Responding to an Incoming Chat*
- *Handling Siebel Chat Sessions*

Note: Most functions available on the communications toolbar support voice, email, and chat.

Logging in to Siebel Chat

When you have successfully installed Siebel Chat, you can launch it from your Siebel Call Center application. To log in to Siebel Chat, complete the steps in the following procedure.

To log in to Siebel Chat

1. Log in to your Siebel Call Center application.
2. Click the Log in button on the communications toolbar to log in to Siebel Chat.

   You are now ready to:

   - Receive, accept, and respond to inbound chat requests

- Transfer and release chat interactions
- Change your status from ready to not ready (and conversely)

You perform all of these actions using the communications toolbar.

Receiving a New Chat

Generally, a customer requests a chat by completing a chat request form. This form can include the following information:

- Product Type. The Product Name or Serial Number.
- Problem Summary. A short description of the problem.
- Problem Description. A detailed description of the problem, including the steps or activities that let up to the problem's occurrence.
- Name. The customer's first name and last name.
- Email Address. The customer's email address.
- Phone Number. The customer's contact phone number.

Siebel Chat uses the information that the user provides on the chat request form to route the chat interaction to a particular workgroup queue.

When an agent (using the Siebel Call Center application) receives a new chat:

- Either the Accept Work Item button on the communications toolbar flashes to notify that a chat request has been queued, or the chat tab flashes to indicate that a new chat has been automatically accepted and queued to the agent.
- At the same time, the communications toolbar displays a message to identify the user requesting a chat, for example, as follows:

  ```
  Chat Request from customer@example.com
  ```

- When the agent accepts the chat, the customer dashboard automatically prepopulates with the following information, if provided by the customer on the chat request form:

  Customer Name

  Phone Number

  Email Address

  For anonymous chat users, the customer dashboard remains blank. For more information about the customer dashboard, see *About the Customer Dashboard*.

Accepting an Incoming Chat

Agents accept an incoming chat that has been queued to them by clicking the Accept Work Item button, when it flashes, on the communications toolbar of the Siebel Call Center application. However if chat auto answer is enabled for

agents in Siebel Call Center, then agents do not need to manually accept the incoming chat as the chat either opens up automatically on the agent's screen or the chat tab flashes to indicate that a new chat has been automatically accepted and queued.

By default, when an agent accepts a chat, the agent automatically becomes not ready or unavailable for other channels. If an agent does not accept an incoming chat:

- The status of that agent automatically changes to not ready, and no further incoming chats are queued to that agent until that agent changes the status back to ready. Agent status is indicated by the status button (State: ready or State: not ready) on the communications toolbar.

  If chat auto answer is enabled and the agent is configured to accept a maximum number of chats at the same time in the third-party chat server, then chat auto answer accepts incoming chats, up to the maximum number of chats allowed. For more information about agent status, see *Changing Agent Status*.

Accepting an Incoming Chat Manually

Use the following procedure to manually accept an incoming chat. Agents must manually accept incoming chats when chat auto answer is disabled for agents. For more information about chat auto answer, see *Roadmap for Configuring Siebel Chat Auto Answer for Agents*.

To accept an incoming chat manually

1. Click the flashing Accept Work Item button on the communications toolbar.

   A chat pane opens in an action pane of your Siebel Call Center application. The chat dashboard is prepopulated with the information provided by the user on the chat request form. The customer dashboard also opens, showing customer information.
2. Respond to the incoming chat as described in *Responding to an Incoming Chat*.

Opening an Automatically Accepted Incoming Chat

Use the following procedure to open an incoming chat that has been automatically accepted. This happens when chat auto answer is enabled for agents. This procedure assumes that there are other active chat sessions or that the agent is already working on an existing chat.

Note: If there are no other active chat sessions or the agent is not working on an existing chat, then there is no need to do anything to open the incoming chat as the chat pane opens up automatically.

For more information about chat auto answer, see *Roadmap for Configuring Siebel Chat Auto Answer for Agents*.

To open an automatically accepted incoming chat

1. Click the flashing chat tab to open the chat session.

   The chat tab stops flashing and a chat pane opens in an action pane of your Siebel Call Center application. The chat dashboard is prepopulated with the information provided by the user on the chat request form. The customer dashboard also opens, showing customer information.
2. Respond to the incoming chat as described in *Responding to an Incoming Chat*.

Responding to an Incoming Chat

When an agent accepts an incoming chat, the agent can respond to the chat in several different ways as described in the following procedure.

To respond to an incoming chat

1. Accept the incoming chat as described in *Accepting an Incoming Chat*.
2. In the message input area of the chat pane, respond to the incoming chat using one or a combination of the following:
    a. Send a message to the customer as described in *Sending a Message*.
    b. Push a Web URL to the customer as described in *Pushing a Web URL*.
    c. Share information with the customer as described in *Sharing Information Using Smart Share*.
3. Depending on the information provided by the customer on the chat request form (contact ID, service request ID), create a new service request if configured to do so, where no service request already exists.

    For anonymous chats, a service request is always created and also a corresponding activity. An activity is created either under a Service Request or a Contact record. For more information about activities, see *Reviewing Inbound Siebel Chat Activities*. For more information about manually creating a service request, see *Creating a Service Request Manually*.
4. Release the chat as described in *Releasing Chat Sessions*.

Sending a Message

Agents have a global list of predefined message responses at their disposal to send to customers, as required, when handling chat interactions. In responding to chat requests, agents can type a message response or they can send a predefined message response, as described in the following procedure. Note the following about using frequently used text (FUT) and composing a message response:

- You can formulate a message response using any combination of FUT and typed in text.
- You can manually edit the FUT before sending the message response.
- All responses that you send to a customer appear in the transcript area of the chat pane.

To use predefined message responses, you must set up FUT for Siebel Chat first, see *Setting Up Frequently Used Text in Siebel Call Center for Siebel Chat*.

To send a message

1. In the message input area of the chat pane, type in the message response, then click Send.
2. If required, send a message response using FUT matching as follows:
    a. On the text button bar, click the Toggle FUT button to enable FUT matching.

        After FUT matching is enabled, matching is based on input text by default, but you can change this to filter on a different match field as shown in a later step in this procedure.

> **Note:** The keyboard shortcut to enable or disable FUT matching is Ctrl+Shift+F. FUT matching is disabled by default.

b. Type in your message.

   While typing in your message, any matches found appear directly in a drop-down list and any matched words show in bold.

c. Click a suggested match to select and use that option.

3. If required, filter FUT matching on a different match field as follows:

   a. On the text button bar, click the Change FUT Match Field button and then select one of the following:

      - Name. Select to filter FUT matching on FUT template name.
      - Type. Select to filter FUT matching on FUT template type (Greeting, Body or Closing).
      - Subject. Select to filter FUT matching on subject.
      - Text. Select to filter FUT matching on FUT template text.

   > **Note:** The keyboard shortcut to change the FUT match field is Ctrl+Shift+U. FUT matching is based on input text by default.

   b. Type in your message.

      While typing in your message, any matches found appear directly in a drop-down list and any matched words show in bold.

   c. Click a suggested match to select and use that option.

Pushing a Web URL

Agents have a global list of Uniform Resource Locators (URLs) at their disposal to share with customers using push URL technology. When agents share a URL with a customer, it is automatically displayed in the customer's browser; the customer does not have to manually copy and paste the URL in to a browser window.

In responding to an incoming chat, agents can type the URL address of a page that they want to share or they can push a predefined URL, as described in the following procedure. All URLs that you send to a customer appear in the transcript area of the chat pane.

To push predefined Web URLs, you must set up a global URL library for Siebel Chat first. For more information about URL library setup, see *Setting Up a Global URL Library in Siebel Call Center for Siebel Chat*.

To push a Web URL

1. In the URL area of the chat pane, enter the URL address of the page you want to send, then click the Push URL button.
2. If required, push a predefined Web URL as follows:

   a. On the URL button bar, click the URL Library button.
   b. Pick a URL option from the list that displays, then click OK.

Note: Pushed URLs can either appear in a frame within the customer's chat window, or in a completely new browser window, depending on configuration. Where a new browser window is launched, subsequent URL pushes simply refresh the same window. For more information about push URL configuration, see *Configuring the Behavior of the Push URL Functionality*.

Sharing Information Using Smart Share

Using the Smart Share button in the chat pane, agents can share highlighted text, service request details, and solution details with customers.

To share information using Smart Share

- On the text button bar of the chat pane, click the Smart Share button to do one of the following:
  - Copy the selected text in an attribute field from the currently active Siebel Call Center view into the message input area.
  - Paste the service request number and summary of a selected Service Request into the message input area; that is, where a single service request record is selected from either a Siebel Call Center list or form view
  - Paste the solution ID and description of a selected Solution into the message input area; that is, where a single solutions record is selected from either a Siebel Call Center list or form view.

Handling Siebel Chat Sessions

This topic includes information about the following:

- *Showing and Hiding the Chat Pane*
- *Handling Multiple Simultaneous Chat Sessions*
- *Transferring Chat Interactions to Others*
- *Handling Transferred Chat Interactions*
- *Releasing Chat Sessions*
- *Closing Chat Sessions*
- *Guidelines for Handling Unsuccessful Chat Interactions*
- *Process of Configuring Siebel Chat Feedback*

Showing and Hiding the Chat Pane

Use the following procedure to show (open) and hide (close) the chat pane. Closing the chat pane does not destroy any active chat sessions.

To show and hide the chat pane

- Click Chat on the global toolbar of Siebel Call Center to open (show) and close (hide) the chat pane, as required.

Note: To close the chat pane, agents can also click the Close (or X) button in the corner of the chat pane. For more information, see *Closing Chat Sessions*.

Handling Multiple Simultaneous Chat Sessions

Agents can handle multiple chat sessions simultaneously.

Where multiple chat sessions are ongoing, agents can move between each chat session without losing context. If multiple chat sessions are open, they are represented by tabs in the chat pane where:

- Each chat tab shows the first name of the customer.
- A number appears after the customer name (for example, John: 3) representing the number of messages sent by the customer since the chat was last viewed by the agent.
- If you hover the mouse over the tab, the last message received is shown without making the agent switch to the chat.

If multiple chat sessions are ongoing, you can switch between chat sessions using either the chat tabs or the communications toolbar; however, the behavior in each case is different as described in the following procedures:

- *Switching Between Chat Sessions Using Chat Tabs*
- *Switching Between Chat Sessions Using the Communications Toolbar*

Note: It is recommended that you use chat tabs to switch between chat sessions and avoid using the communications toolbar, which is less efficient and prone to error.

Switching Between Chat Sessions Using Chat Tabs

By default, all open chat sessions are automatically put on hold except for the chat session that is currently active. When an agent moves away from the currently active chat session by clicking another chat tab, the previously active chat is automatically placed on hold and the previously suspended chat is resumed.

Where all chat sessions are on hold and an agent manually accepts a new incoming chat, then that new chat becomes the active chat session and all other chat sessions remain on hold.

To switch between chat sessions using chat tabs

1. In the chat pane, click the chat tab belonging to the chat session that you want to move to.
2. Resume work on the active chat session as described in, for example, *Responding to an Incoming Chat* and *Reviewing Inbound Siebel Chat Activities*.

Switching Between Chat Sessions Using the Communications Toolbar

Agents can manually switch between chat sessions using the drop-down list of currently open chats, the Hold Work Item button, and the Resume Work Item button on the communications toolbar. If you put a chat session on hold using the communications toolbar, then you must use the communications toolbar to resume another chat session as well. Switching between chat sessions using the communications toolbar is a two-step process described in the following procedure.

To switch between chat sessions using the communications toolbar

1. Put the active chat session on hold by selecting it from the drop-down list of currently open chats, then click the Hold Work Item button.

   At this point, all chat sessions are on hold because no chat has been resumed yet. In addition, all chat tabs are disabled and remain disabled until you resume another chat session using the communications toolbar.
2. Resume a different chat session by selecting it from the drop-down list of currently open chats, then click the Resume Work Item button.

Transferring Chat Interactions to Others

It is possible for an agent to transfer or reassign a chat, which the agent has already accepted, to another workgroup or agent. Transferred chats are processed exactly the same way as other incoming chats.

When a chat is transferred, the chat transferred activity created by the transferring agent is updated to associate the receiving agent with the activity, and the chat transcript is attached to the activity. While a chat is being transferred, the customer is not able to submit any messages to the agent until the (new) receiving agent accepts the transferred chat.

Transferring a Chat to a Workgroup

To transfer a chat to a workgroup, complete the steps in the following procedure.

To transfer a chat to a workgroup

1. Select the chat that you want to transfer:

   In the chat pane, click the tab of the chat interaction that you want to transfer.
2. Click the Bind Transfer Work Item button on the communications toolbar of Siebel Call Center.
3. Select a workgroup on the Bind Transfer Work Itemlist that opens, then click OK.

   The chat is transferred to the selected workgroup. The chat transcript includes all previous conversations and interactions that the previous agent had with the customer.

   The Bind Transfer Work Item list shows only the available chat workgroups. It does not show any available voice (or CTI) workgroups.



Note: Any messages that the customer sends while the agent is transferring a chat are lost; that is, they are not captured in the chat window or transcript.

Transferring a Chat to Another Agent

To transfer a chat to another agent, complete the steps in the following procedure.

To transfer a chat to another agent

1. Select the chat that you want to transfer:

   In the chat pane, click the tab of the chat interaction that you want to transfer.
2. Click the Transfer to Agent button on the communications toolbar of Siebel Call Center.
3. Select an agent that has a status of Available on the Transfer to Agent list that opens, then click OK.

The chat is transferred to the selected agent. The chat transcript includes all previous conversations and interactions that the previous agent had with the customer.

You cannot transfer a chat to an agent that has a status of Busy, On Break, or Logged Out.

Note: Any messages that the customer sends while the agent is transferring a chat are lost; that is, they are not captured in the chat window or transcript.

Handling Transferred Chat Interactions

Agents handle transferred chats more or less the same way as other incoming chats. When a chat is transferred, the only way to distinguish it from a new incoming chat is by the presence of a chat transcript (that is, by the presence of the chat exchanges that took place between the previous agent and the customer) in the transcript area of the chat pane. To handle a transferred chat, complete the steps in the following procedure.

To handle a transferred chat

1. Upon receipt of a transferred chat interaction, see *Transferring Chat Interactions to Others*, the receiving agent can do one of the following as required:
    - Click the flashing chat tab to open the chat session (when chat auto answer is enabled).
    - Click the Accept Work Item button on the communications toolbar to open the chat session (when chat auto answer is disabled).
2. Review the chat transcript (if there is one) in the transcript area of the chat pane.

    For more information about how agents review chat activities and transcripts, including inbound chat activities associated with service requests and contacts, see *Reviewing Inbound Siebel Chat Activities*.
3. Respond to the transferred chat, see *Responding to an Incoming Chat*.

Releasing Chat Sessions

Releasing a chat session means that you are ending the chat session. Agents use the communications toolbar of Siebel Call Center to release a chat. Customers click the Terminate (or similar) button in their browser window to release a chat.

When an agent or customer releases a chat session, a follow-up email is sent to the customer with a transcript of the chat session attached provided the system preference Chat: Auto Email Mode is set, and that outbound email communications and an email profile are set up in Siebel Call Center. In the case where a chat is transferred several times, the follow-up email is sent only upon final release of the chat session.

- For information about email profile setup in Siebel Call Center, see *Siebel Email Administration Guide* .
- For information about outbound email communications setup in Siebel Call Center, see *Setting Up Outbound Email Communications in Siebel Call Center*.

Closing Chat Sessions

Typically, when an agent or customer releases a chat session, all action buttons within the chat pane are disabled but agents can still read a transcript of the chat interaction in the chat transcript area. To close the chat session, agents must do one of the following:

- Where only one chat session is open, agents must exit the communications panel (by clicking the X button in the corner of the communications panel) to close the chat session.
- Where multiple chat sessions are open and the current chat session is released, agents must switch to another chat tab to automatically close the current chat session.
- Where multiple chat sessions are open and a chat session in the background is released, then the released chat session is automatically closed.

After a chat session is closed, the chat transcript is attached to the activity and also to the email that is sent to the customer at the end of the chat session. For more information about transcript attachments, see *Configuring the Appearance of Transcript Attachments*.

Guidelines for Handling Unsuccessful Chat Interactions

If an agent cannot find a satisfactory solution for an inbound chat request, then the agent can do one or more of the following:

- Search for and review other shared solutions to see if one might be useful.
- Create a new predefined message response capturing the information, see *Setting Up Frequently Used Text in Siebel Call Center for Siebel Chat*.
- Create a new solution and add it to the URL library, see *Setting Up a Global URL Library in Siebel Call Center for Siebel Chat*.
- Associate the (new) solution with the appropriate activity, contact, or service request. Agents can associate solutions with:
  - Inbound chat activities by navigating to the Activities - Activities List view.
  - Service requests by navigating to the Service screen, then the Service Requests List view.
  - Contact records by navigating to the Contacts screen, then the Contacts List view.

  For more information about inbound chat activities, see *Reviewing Inbound Siebel Chat Activities*.
- Reassign the chat interaction to another workgroup, see *Transferring Chat Interactions to Others*.

About the Chat Dashboard

The chat dashboard in the chat pane of the communications panel displays details about the problem that the customer is having based on information passed by the customer on the chat request form, including the following:

- Area. The area where assistance is needed.
- Subarea. The subarea within the area where assistance is needed.

- Product. The Product ID of the area or subarea where assistance is needed.
- Summary. A summary of the problem, if provided.
- KB Visited. A list of knowledge base articles that the customer has viewed prior to requesting a chat with an agent. The agent can choose an item from the list and navigate to its detail page.
- Action. The Action drop-down menu contains a number of quick actions that the agent can perform, including the following:
  - *Creating a Service Request Manually*
  - *Viewing Service Requests, Activities, and Contacts Using the Chat Dashboard*
  - *Refreshing Siebel Chat Dashboard Fields*

Related Topics

About the Customer Dashboard

About Customizing Siebel Chat

Customizing Siebel Chat

About the Customer Dashboard

The customer dashboard in Siebel Call Center automatically shows customer contact information (Customer Name, Phone Number, and Email Address), if provided on the chat request form.

The purpose of the customer dashboard is to alert the agent if the customer requesting a chat provides profile information which does not match the contact record. For example, if the phone number or email address provided by a known customer does not match the contact record on file, then an exclamation mark appears next to these fields indicating to the agent that this information is new. For more information about configuring the customer dashboard, see *Configuring Siebel Business Applications* .

Creating a Service Request Manually

To manually create a service request using the chat dashboard in the chat pane, complete the steps in the following procedure.

Note: An activity can be linked to only one service request at any given time.

To manually create a service request

1. Click the Action drop-down menu in the chat dashboard.
2. Select Create Service Request to manually create a new service request for an incoming chat.

   If this is the first service request that is created for the chat, then the inbound chat activity is associated with it. If this is not the first service request, then this subsequent service request is regarded as standalone - which means that the inbound chat activity is not associated (or linked) to this service request.

Note: Once a new service request is created, the agent can associate it with the current chat activity record by navigating to Tools, then Communications in Siebel Call Center, and clicking the Associate button.

By default, the Create Service Request action is contextually smart. It prepopulates the new service request form with data displayed in the customer dashboard and chat dashboard.

Viewing Service Requests, Activities, and Contacts Using the Chat Dashboard

To view service requests, activities, and contacts using the chat dashboard in the chat pane, complete the steps in the following procedure.

To view service requests, activities, and contacts using the chat dashboard

1. Click the Action drop-down menu in the chat dashboard.
2. Select one of the following, as required:
    a. Select View Service Requests to view the service requests linked to the current chat, or the service request detail screen for the contact.
    b. Select View Activities to view the activities linked to the contact.
    c. Select View Contact to view the contact details for the contact. This option applies to existing contacts only, not to anonymous chat users.

Showing and Hiding the Chat Dashboard

Use the following procedure to hide the chat dashboard in the chat pane.

To hide the chat dashboard

- Click the Hide Dashboard button located in the header bar of the chat pane.

After you have minimized the chat dashboard, use the following procedure to show the chat dashboard again.

To show the chat dashboard

- Click the Show Dashboard button located in the header bar of the chat pane.

Changing Agent Status

When an agent logs into Siebel Chat from the Siebel Call Center application, the agent's status is automatically set to State: ready as indicated by the status button on the communications toolbar. If agents want to change their status from

State: ready (available) to State: not ready (unavailable), then they must click the status button on the communications toolbar.

To change agent status

1. Log in to your Siebel Call Center application.
2. Click the Log in button on the communications toolbar to log in to Siebel Chat.

   Notice that your status, as indicated by the status button on the communication toolbar, is set to State: ready by default.
3. Click the status button on the toolbar to change your status from State: ready to State: not ready (and conversely).

Agent and Siebel Chat Interaction States

The following table lists the different chat interaction states that exist for Siebel Chat, which have a visible effect for the agent. Some of these states are automatically forced by Siebel Chat and others are manually triggered by the agent.

Interaction State	Description and Siebel Call Center Behavior	Automatic or Manual?
Active	The chat session is currently active, whereby the agent is actively chatting with the customer. • The corresponding chat tab in the chat pane is in the foreground. • The communications toolbar shows this chat as the active chat interaction in the drop-down list of currently open chats. • The Chat Activity has a status of In Progress.	Automatic.
On Hold	The chat is on hold. • The corresponding chat tab in the chat pane is in the background. • The Chat Activity still has a status of In Progress.	Manual. Agents can place the currently active chat session on hold by clicking another chat tab, or they can use the communications toolbar to switch between chats. For more information, see *Handling Multiple Simultaneous Chat Sessions*.
Transferred	The chat has been transferred. • When the chat is accepted by another agent, the interaction state changes to In Progress. • The transferring agent's chat tab is removed once the agent moves onto another chat, or closes the chat pane. • The Chat Activity has a status of Transferred. • When the transferred chat is accepted by the receiving agent, a new activity is created with a status of In Progress.	Manual. Agents must click the Get Transfer List button on the communications toolbar to transfer a chat interaction that they are currently working on to another workgroup. For more information, see *Transferring Chat Interactions to Others*.

Interaction State	Description and Siebel Call Center Behavior	Automatic or Manual?
Terminated	The chat session is terminated. • The chat tab in the chat pane is removed manually, if the agent moves on to another chat tab or closes the chat pane. • The Chat Activity has a status of Done.	Manual. Agents can manually release a chat session by clicking the Release Work Item button on the communications toolbar. Customers can also manually release a chat session by clicking the Terminate button in their browser window. For more information about releasing a chat, see *Releasing Chat Sessions*.
Idle	There has been no activity within a chat session from either the agent or the customer for over two minutes. **Note:** Currently, you cannot configure this two-minute time limit.	Automatic.
Aborted	Either the agent or customer ends the chat session by means other than clicking the Release Work Item or Terminate buttons (for example, a browser is killed or a network mains is unplugged).	Automatic.

The following table lists the different agent states that exist for Siebel Chat agents. Some of these states are automatically forced by Siebel Chat and others are manually triggered by the agent.

Agent State	Description	Automatic or Manual?
Chatting	The agent is engaged, or actively chatting, in at least one chat session.	Automatic.
Working	The agent is available to handle chat requests.	Manual. Agents must click the Status button on the communications toolbar to set their status to available (State: Ready). For more information about agent status, see *Changing Agent Status*.
On Break	The agent is currently unavailable to handle chat requests.	Manual. Agents must click the Status button on the communications toolbar to set their status to unavailable (State: Not Ready). For more information about agent status, see *Changing Agent Status*.

Setting Up Logging and Tracing for Siebel Chat Components

Logging and tracing can be set up for the following Siebel Chat components:

- Communications Management component group

  For more information, see *Logging for the Communications Management Component Group*.

  > **Note:** The Communications Management component group includes the following server components: Communications Session Manager, Communications Configuration Manager, Communications Inbound Receiver, Communications Inbound Processor, Communications Outbound Manager, Page Manager, and Email Manager. For more information about the Communications Management component group, see *Siebel CTI Administration Guide* .

- Siebel Server

  For more information, see *Setting Siebel Server Logging Parameters*.

- Siebel Call Center

  For more information, see *Siebel Chat Logging*.

Logging for the Communications Management Component Group

Logging for the Communications Management component group is carried out using the following mechanisms:

- Communications Session Logging. Logs information and errors related to the processing of chat interactions in the Communications Management component group.

  Communications session logging can be turned on by setting the LogFile and LogDebug parameters in the Communication Session Manager server component.

  All Chat UI Business Service messages (for example, in relation to Siebel Chat commands and events) are logged to this file.

- Driver Logging. Logs any third-party chat driver-specific information and errors.

  Driver logging can be turned on during driver configuration, using the Driver:LogFile parameter at the Profile Parameter Override level. For more information about driver configuration, see *Configuring Communications in Siebel Call Center for Siebel Chat*.

For more information about session communications and communications drivers, see *Siebel CTI Administration Guide* .

Setting Siebel Server Logging Parameters

Set Siebel Server logging parameters as described in the following procedure.

To set Siebel Server logging parameters

1. Navigate to the Administration - Server Configuration screen, then the Servers view.
2. Select your Siebel Server, and then its Call Center Object Manager component.
3. Set the following parameters to the value specified:

Parameter	Value
Log Debug Message	Set to True.
Log File	Set to the name of your log file.
Log Level	Set to a value between 1 and 5, where 1 represents the least logging and 5 the most logging (debug).

Siebel Chat Logging

In Siebel Call Center, chat logging is carried out using the standard application logging mechanisms:

- CCF Logging. Logs information and errors related to the processing of chat interactions in the Siebel Call Center application components.
- SARM Logging. Logs performance profiling messages from the chat application components.

The standard client-side logging capabilities of Siebel Call Center are also used for customer Siebel Chat logging, whereby logging is set up and controlled from the customer's workstation using the environment variables described in the following table.

Environment Variable	Description
SEBLCL_TRACEMODE	Controls the log mode/device choice. Set this variable to 2 to log to file.
SEBLCL_TRACEUNICODE	Enables tracing in Unicode mode. This variable is useful for tracing international language data, and is disabled by default.
SEBLCL_LOGDIR	Specifies the location on client disk where the log file is created.
SEBLCL_LOGFILESIZE	Specifies the maximum size of the log file.

Environment Variable	Description
SEBLCL_LOGARCHIVECOUNT	Specifies the maximum number of logs archived file count.

About Web Notifications

Siebel supports Web notifications for incoming calls and chats. The Siebel Web notifications feature is based on the standard Web Notifications API. Siebel integrates the Web Notifications API (*http://www.w3.org/TR/notifications/*) with Siebel CTI and Siebel Chat so that when any of the following happens, an alert or pop-up window appears notifying agents about incoming calls and chats so that customer calls or chats will not be missed:

- Agents navigate away from the main application browser window.
- The application browser window is minimized or hidden behind another application window.

Clicking inside the Web notification pop-up window brings the application browser window to the front. There is a Bring Siebel to Front user preference which you can set to one of the following options, to control when to display Web notifications for incoming calls and chats:

- On All Incoming Work Items
- On Matching Events
- Off

For more information about the Bring to Front user preference, see *Siebel CTI Administration Guide* and Doc ID 216960.1 on My Oracle Support.



Note: This topic is valid for Google Chrome and Mozilla Firefox browsers since they both provide native support for the Web Notifications API. Microsoft IE browsers, however, do not support the Web Notifications API out-of-the-box. The Desktop Integration Siebel Agent (DISA) is provided to support Web notifications functionality in IE browsers. For more information about DISA, see Desktop Integration Siebel Agent Guide.

5 Customizing Siebel Chat

Customizing Siebel Chat

This chapter describes how to customize Siebel Chat, how to configure keyboard shortcuts for chat, and how to configure chat feedback. It contains the following topics:

- *About Customizing Siebel Chat*
- *Siebel Call Center and Siebel Chat-Related Applets*
- *Configuring the Appearance of Transcript Attachments*
- *Configuring the Behavior of the Push URL Functionality*
- *Customizing the Siebel Chat User Interface*
- *Process of Configuring Siebel Chat Feedback*

Note: The functionality described in this chapter applies only if you are using the default workflow to handle incoming chat requests. For more information about the default workflow for Siebel Chat, see *Chat Standard Accept Process Workflow*.

About Customizing Siebel Chat

Administrators can configure the appearance of the Siebel Chat interface (or chat pane), the appearance of transcript attachments, and the behavior of the push URL functionality. However, some Siebel Call Center and Siebel Chat-related applets that are configured in Siebel Tools must not be changed.

For more information about the Siebel Chat interface, which is part of the communications panel, see *Siebel Chat Interface*. For more information about the communications panel, see Siebel Fundamentals for Siebel Open UI and Siebel CTI Administration Guide.

For more information about customizing Siebel Chat, see the following:

- *Siebel Call Center and Siebel Chat-Related Applets*
- *Configuring the Appearance of Transcript Attachments*
- *Configuring the Behavior of the Push URL Functionality*
- *Customizing the Siebel Chat User Interface*

Siebel Call Center and Siebel Chat-Related Applets

Some Siebel Call Center and Siebel Chat-related applets are configured in Siebel Tools. The names of these applets, which are listed in the following table, must not be changed because they might not function correctly.

Applet Name	Description	Business Component
Chat UI KB Visited Pick Applet	Shows a list of the knowledge base items visited (by a customer requesting a chat) to the agent.	Chat UI KB Visited
Chat UI URL Library Pick Applet	Shows a list of the available URLs to push to a customer during a chat.	Chat UI URL Library
Chat Web Projects Pick List Applet	Shows a list of third-party chat projects to which a chat interaction can be routed.	Chat Web Projects Pick List Applet

Configuring the Appearance of Transcript Attachments

When a chat session is released, a transcript of the chat session is attached to the activity and also to the email that is sent to the customer at the end of the chat session. Chat transcripts are stored in XML format, but before a transcript is sent as an attachment, it is transformed in to HTML using the chat.xsl file. This file is located in the following directory:

```
SIEBSRVR_ROOT\webtempl
```

Administrators can configure the appearance of transcript attachments by modifying the chat.xsl file. This file contains parameters that can be modified to get the required look and feel for transcript attachments. To configure the appearance of transcript attachments, complete the steps in the following procedure.

To configure the appearance of transcript attachments

1. Open the chat.xsl file.
2. Modify the parameters, as required.
3. Save the chat.xsl file to the following location on Siebel Server for the changes to take effect.

```
SIEBSRVR_ROOT\webtempl
```

Configuring the Behavior of the Push URL Functionality

You can configure the behavior of the push URL functionality for Siebel Chat, whereby pushed content can appear to the user in either one of the following ways:

- In a frame alongside the customer chat UI in the same browser.

  This is the default setup for Siebel Chat.

CAUTION: Some Web sites (for example, hotmail.com) with heavy scripting and other background processes might prompt the user with something similar to the following:You are about to leave the application. Clicking OK will take you away from the chat application, and terminate the chat interaction.

- In a completely new browser window.

  If you want your pushed content to display in a completely new browser window, then you must modify the web.xml configuration file, as described in the following procedure.

  If this option is configured, the first URL pushed opens in a new browser window, and subsequent URLs pushed refresh the original browser window.

To configure the behavior of the push URL functionality for Siebel Chat, complete the steps in the following procedure.

To configure the behavior of the push URL functionality

1. Open the web.xml configuration file.
2. Modify the following option in web.xml:

```
context-param>
 param-name>SiebelChatPushedURLAsPopup/param-name>
 param-value>off/param-value>
/context-param>
```

   a. Set this option to on if you want pushed content to display in a new browser window.
   b. Set this option to off if you want pushed content to display in a frame alongside the customer chat UI.
3. Save the web.xml file.

Note: It is not necessary to shut down and restart the servers for this change to take effect, nor for the cache to be cleaned.

Customizing the Siebel Chat User Interface

Administrators can configure the appearance of the Siebel Chat interface, the behavior of buttons and functionality on the chat interface, and keyboard shortcuts for chat functionality. For more information, see the following topics:

- *Configuring Siebel Chat Dashboard Fields*
- *Formatting Siebel Chat Dashboard Fields*
- *Configuring Siebel Chat Pane Drop-Down Controls*
- *Refreshing Siebel Chat Dashboard Fields*
- *Removing the URL Area from the Chat Pane*
- *Configuring the Button Sequence for the Button Bar*
- *Configuring Siebel Chat Quick Actions*
- *Configuring the Display Name for Siebel Chat Sessions*
- *Configuring Keyboard Shortcuts for Buttons on the Siebel Chat UI*

- *Process of Configuring Siebel Chat Feedback*

When you remove elements from the chat dashboard, you increase the chat transcript and message input areas of the chat pane as a result. Before starting to customize the Siebel Chat interface, it is recommended that you review the following topics:

- *About Customizing Siebel Chat*
- *Configuring the Appearance of Transcript Attachments*
- *Methods for Customizing Siebel Chat Process Flow and Interface Functionality*

Configuring Siebel Chat Dashboard Fields

You can configure the following fields for the chat dashboard: Area, Subarea, Product, Summary (or Problem). To add, remove, or modify chat dashboard fields, you must configure the DashboardFields control user property under the Chat Tab Applet in Siebel Tools. Because of the limited size of the chat pane, it is recommended that you configure up to a maximum of five or six fields for the chat dashboard. Use the following procedure to configure the chat dashboard fields.

To configure Siebel Chat dashboard fields

1. Start Siebel Tools.
2. In the Object Explorer, go to Project, find Chat UI, and then lock Chat UI.
3. In the Object Explorer, go to Applet, and then find Chat Tab Applet.
4. Drill down on Control, and then select DashboardFields.
5. Drill down on Control User Prop.
6. Configure, for example, the following chat dashboard fields as required:
    - Area
    - Subarea
    - Product
    - Summary (or Problem)
    - MaxRowNum

      MaxRowNum represents the number of rows to show on the chat dashboard. By default, the value in MaxRowNum is 6. However, if MaxRowNum is set 5, then only five rows appear on the chat dashboard.
7. To modify the name of a field as it is displayed on the chat dashboard, edit the Value field.
8. To remove a field from the chat dashboard, click the Inactive field next to the Value field.
9. Publish the repository updates.
10. Restart the Siebel Server.

Formatting Siebel Chat Dashboard Fields

For each chat dashboard field configured, you can define where it appears on the chat dashboard (the row and column) and the width of the field. Use the following procedure to format chat dashboard fields.

To format Siebel Chat dashboard fields

1. Start Siebel Tools.
2. In the Object Explorer, go to Project, find Chat UI, and then lock Chat UI.

3. In the Object Explorer, go to Applet, and then find Chat Tab Applet.
4. Drill down on Control, and then select DashboardFields.
5. Drill down on Control User Prop.
6. Configure, for example, the following fields as required:
    - Column. The column where the field appears. The value can be either 1 or 2 because each row on the chat dashboard can be divided into a maximum of two columns.
    - Length. The width of the field, which can be either half or full. The length value is case sensitive. A value of full indicates that the field width is the entire length of the row; half indicates that the field width is only half the length of the row. Any other value is treated as a full value.
    - Row. The row where the field is located. The value can be from 1 to MaxRowNum. MaxRowNum is configured for the DashboardFields control user property. Any value less than 1 or larger than MaxRowNum is ignored.
    - Value. Indicates the data source to which the field belongs. The value can be interaction data or a business component field. The format of the value is as follows: {BC Name:fieldName} or {CCA-Data:fieldName}.

    For example, the user properties for the Area and Product fields are shown in the following table.

User Property Name	Values for the Area Field	Values for the Product Field
Column	1	1
Length	half	half
Row	1	2
Value	CCA_Data:Area	ServiceRequest:Product

7. Publish the repository updates.
8. Restart the Siebel Server.

Configuring Siebel Chat Pane Drop-Down Controls

You can configure the controls for the following drop-down elements on the chat pane:

- KB Visited field control and button
- Action field control and button

Use the following procedure to configure the controls for the KB Visited and Action on the chat pane.

To configure Siebel Chat pane drop-down controls

1. Start Siebel Tools.
2. In the Object Explorer, select Applet, and then Chat Tab Applet.
3. In the Object Explorer, select Applet User Prop, then the appropriate control.
4. To configure the KB Visited control, do one of the following:

- Deselect the Visible property for KB to make the whole field control invisible on the chat dashboard. Select the Visible property for KB to make the whole field control visible.
- Set the Visible property for the KBButton control to False to make only the KB Visited button invisible on the chat dashboard. Set it to True to make only the KB Visited button visible on the chat dashboard.

5. To configure the QuickActionList control:
   - Set the Visible property for QuickActionList to False to make the whole field control invisible on the chat dashboard. Set it to True to make the whole field control visible.
   - Set the Visible property for the Go control to False to make only the Action button invisible on the chat dashboard. Set it to True to make only the Action button visible.
6. Publish the repository updates.
7. Restart the Siebel Server.

Refreshing Siebel Chat Dashboard Fields

By default you can manually refresh Siebel Chat dashboard fields. However, you can also set up chat dashboard fields to dynamically refresh.

Refreshing Siebel Chat Dashboard Fields Manually

After changing fields, such as, Account and Contact on the Chat Service Request detail view, you can manually refresh the chat dashboard fields during a chat session to show the latest changes.

To manually refresh the Siebel Chat dashboard fields

1. Click the Action menu in the chat dashboard.
2. Select Refresh Chat Dashboard Fields, and then click Go.

Configuring Dynamic Refresh for Siebel Chat Dashboard Fields

If you want the fields on the chat dashboard to update straight away after an agent changes the business component values, then you must configure a new action set and run-time event respectively. Use the following procedure to configure dynamic refresh for Siebel Chat dashboard fields so that after an agent updates, for example, the service request record, the chat dashboard fields are updated automatically as well.

To configure dynamic refresh for Siebel Chat dashboard fields

1. Log in to your Siebel Call Center application.
2. Create the action set to update the chat dashboard as follows:
   a. Navigate to Administration - Runtime Events, Actions Sets view, select the Personalization Action Set list applet, and then add a record with the values shown in the following table.

Field	Value
Name	Update Chat Dashboard
Active	Selected

Field	Value
Enable Export	Selected
Description	Update the BC field values on the chat dashboard.

b. Navigate to Administration - Runtime Events, Actions Sets view, select the Personalization Action list applet, and then add a record with the values shown in the following table.

Field	Value
Name	Update Chat Dashboard
Action Type	BusService
Sequence	1
Active	Selected **Note:** This field indicates whether the action is active or not.
Business Service Name	Chat UI Business Service
Business Service Method	RefreshChatPaneDashboard

3. Create the event alias, and then the event as follows:

a. Navigate to Administration - Runtime Events, Event Aliases view, and then add a record with the values shown in the following table.

Field	Value
Name	Service Request - Update
Object Type	BusComp
Object Name	Service Request
Event	WriteRecordUpdate

Field	Value

b. Navigate to Administration - Runtime Events, Events view, and then add a new event with the values shown in the following table.

Field	Value
Name	Service Request - Update After you select the event name from the drop-down list, the Object Type, Object Name, and Event fields are filled in automatically.
Object Type	BusComp
Object Name	Service Request
Event	WriteRecordUpdated
Sequence	Enter a sequence number (for example, 2) that does not conflict with the current event.
Action Set Name	Update Chat Dashboard.

4. Restart the Siebel Server for the run-time event to take effect.

   Whenever an agent updates a service request, an action is triggered to update all the BC fields (not only the service request fields) on the chat dashboard.

   You can configure more events, as required, using other business components whose fields display on the chat dashboard to trigger the update dashboard action. For example, you can configure an event triggered by updating a contact. In this case, all BC fields on the chat dashboard are updated when the agent updates the contact.

Removing the URL Area from the Chat Pane

You can remove the URL area from the chat pane by modifying the Siebel CRM cascading style sheet. When you remove the URL area from the chat pane, the chat transcript area by default expands. Complete the steps in the following procedure to remove the push URL area from the chat pane. The image in *Siebel Chat Interface* shows the different areas of the chat interface: chat dashboard, transcript area, message input area, and URL area. For more information about cascading style sheets, see *Siebel Developer's Reference* .

To remove the URL area from the chat pane

1. Open the Siebel CRM cascading style sheet file.
2. Modify the chatAxLayout parameter, as follows:
   a. Remove the old ButtonBarHeight property.
   b. Add the following two new properties:
      - TextButtonBarHeight. This value is the height of the text button bar, which contains the following buttons: Send, Frequently Used Text, and Smart Share.
      - URLButtonBarHeight. This value is the height of the URL button bar, which contains the following buttons: Push URL and URL Library.

      In the following code example, URLButtonBarHeight is set to 24, and UrlInputHeight is set to 24, which means that the URL area appears in the chat pane:

```
.chatAxLayout
{
 MinAxHeight:400;
 MinAxWidth:300;
 TitleBarHeight:24;
 TabBarHeight:22;
 TextButtonBarHeight: 24;
 TextInputHeight:75;
 URLButtonBarHeight: 24;
 UrlInputHeight: 24;
}
```

   c. To remove the URL area from the chat pane, set the URLButtonBarHeight and UrlInputHeight properties to zero, as shown in the following code example:

```
.chatAxLayout
{
 MinAxHeight:400;
 MinAxWidth:300;
 TitleBarHeight:24;
 TabBarHeight:22;
 TextButtonBarHeight: 24;
 TextInputHeight:75;
 URLButtonBarHeight: 0;
 UrlInputHeight: 0;
}
```

3. Save the Siebel CRM cascading style sheet file to the Siebel Application Interface for the changes to take effect.

Configuring the Button Sequence for the Button Bar

You can configure the order in which buttons appear on the text and URL button bars, by configuring the Sequence property for the appropriate button control user property in Siebel Tools. The Sequence property accepts any number greater than zero. The decimal part of a number (if there is one) is ignored. To insert a space between two adjacent buttons, skip a sequence number. After the sequence is configured for the buttons on the text and URL button bars, the buttons appear on the chat pane in ascending order moving from the outside border of the chat pane to the inside.

To configure the button sequence for the button bar

1. Start Siebel Tools.

2. In the Object Explorer, select Applet, then the Chat Tab Applet.
3. In the Object Explorer, select Control.
4. Configure the Sequence property for the buttons on the URL button bar as follows:

   Select the PushUrl or UrlLib control user property as required and configure Sequence as shown in the following table.

Name	Caption	Parent	Sequence
PushUrl	Push URL	Url	1
UrlLib	URL Library	Url	3

   where:

   - *Name* is the name of the control user property being configured.
   - *Caption* value is the name that appears when you hover over the button in the chat pane.
   - *Parent* value indicates the button bar (in this case, the URL button bar) to which the button belongs.
   - *Sequence* value represents the location where the button appears on the button bar.
5. Configure the Sequence property for the buttons on the Text button bar as follows:

   Select the Send, FUT, or SmartShare control user property as required, and configure Sequence as shown in the following table.

Name	Caption	Parent	Sequence
Send	Send	Text	1
FUT	Frequently Used Text	Text	4
SmartShare	Smart Share	Text	5
ChangeLanguage	Change Language	Text	6

   where:

   - *Name* is the name of the control user property being configured.
   - *Caption* value is the name that appears when you hover over the button in the chat pane.
   - *Parent* value indicates the button bar (in this case, the Text button bar) to which the button belongs.
   - *Sequence* value represents the location where the button appears on the button bar.
6. Publish the repository updates.

7. Restart the Siebel Server.

Configuring Siebel Chat Quick Actions

Agents by default can perform a number of quick actions using the Action menu on the chat dashboard. For more information about these quick actions, see *About the Chat Dashboard*. To configure a new Siebel Chat quick action, perform the following tasks:

- *Configuring a Quick Action for the Chat Dashboard in Siebel Tools*
- *Adding a Quick Action to the QuickActionList Control in Siebel Tools*

Configuring a Quick Action for the Chat Dashboard in Siebel Tools

Use the following procedure to configure a new quick action for the chat dashboard in Siebel Tools.

To configure a new quick action for the chat dashboard

1. Start Siebel Tools.
2. In the Object Explorer, select Applet, and then Chat Tab Applet.
3. In the Object Explorer, select Applet User Prop.
4. Add a new control property with the following information:
   - Name. The name of the action. For example: ViewContactSR.
   - Caption. The name of the action as it appears in the Action menu on the chat dashboard. For example: View Contact for Current Chat Service Request.
5. Configure the control user properties for ViewContactSR, for example, as shown in the following table.

   In this example, the ViewContactSR control calls the Workflow Process ManagerSaRunProcess with the input parameters `{ProcessName, ContactId}`.

User Property Name	Description	Example Value
Business Service Name	The business service that this action calls. If the action is a workflow, then the value in this field is Workflow Process Manager. If the action is not a workflow, then the value can be any service.	Workflow Process Manager
Business Service Method	The business service method that this action calls. If the action is a workflow, then the value in this field is RunProcess. If the action is not a workflow, then the value can be any service.	RunProcess
Process Name	If the action is to call a workflow, then the value in this field is the workflow name. You can ignore this property if the action does not call a workflow.	View Contact of Current Chat
[Parameters] For example, ContactId.	To pass parameters into the service method or workflow, you must define the name-value pairs here. *Name* is the input argument name. For *value*, two formats are supported:	ServiceRequest:ContactId

User Property Name	Description	Example Value
	○ A literal value, consisting of a string in double quotation marks ○ One of the following: - `{BC Name:fieldName}` - `{CCA-Data:fieldName}`	

6. Publish the repository updates.
7. Restart the Siebel Server.

Adding a Quick Action to the QuickActionList Control in Siebel Tools

Use the following procedure to add a quick action to the QuickActionList control in Siebel Tools.

To add a quick action to the QuickActionList control in Siebel Tools

1. Start Siebel Tools.
2. In the Object Explorer, select Applet, and then Chat Tab Applet.
3. In the Object Explorer, select Applet User Prop, then the QuickActionList control.
4. Configure the user properties for the QuickActionList control, for example, as shown in the following table.

User Property Name	Description	Example Value
Name	ActionItem*X*, where X represents the order in which the action appears in the Action menu on the chat dashboard. ActionItem2 indicates that the action appears as the second item on the list.	ActionItem5
Value	The control name of the action that you are adding to the quick action list.	ViewContactSR

5. Publish the repository updates.
6. Restart the Siebel Server.

Configuring the Display Name for Siebel Chat Sessions

You can configure and customize the display name for chat sessions. The following procedure shows how to configure the display name for chat sessions.

Note: Siebel Chat by default provides a run-time event and the Chat Display Name Update Process workflow for configuring the display name for Siebel Chat sessions. You can modify both the run-time event and the workflow as required.

To configure the display name for Siebel Chat sessions

1. Verify that the Accept Chat Workflow name in the ChatAcceptedLog is configured in your Siebel Business application as follows:
   a. Navigate to Administration - Communications, then the All Event Logs view.
   b. Query for ChatAcceptedLog, locate ServiceParam.ProcessName, then verify the workflow name, and make sure that it has included the workflow step to update the display name for Siebel Chat sessions.
   c. Query for ChatTransferAcceptedLog, locate ServiceParam.ProcessName, then verify the workflow name, and make sure that it has included the workflow step to update the display name for Siebel Chat sessions.

   Note: Siebel Chat provides a default sample workflow to handle incoming chat requests called Chat Standard Accept Process.

2. Verify that the run-time events for updating the chat display name have been added to your Siebel application and configured appropriately as follows:
   a. Navigate to Administration - Runtime Events, Actions Sets view, then select the Personalization Action Set list applet.
   b. Add a record with the values shown in the following table, if a record is not already configured with these values.

Field	Value
Name	Update Chat Display Name
Active	Selected
Enable Export	Selected
Description	Update Chat Display Name

   c. Navigate to Administration - Runtime Events, Actions Sets view, then select the Personalization Action list applet.
   d. Add a record with the values shown in the following table, if not already configured.

Field	Value
Name	Update Chat Display Name
Action Type	BusService
Sequence	1

Field	Value
Active	Selected **Note:** This field indicates whether the action is active or not. If this field is not selected, then the action is ignored when the event occurs.
Business Service Name	Workflow Process Manager
Business Service Method	RunProcess
Business Service Context	“ProcessName”, “Chat Display Name Update Process”

e. Navigate to Administration - Runtime Events, then the Events view.
f. Add a record with the values shown in the following table, if not already configured.

Field	Value
Action Set Name	Update Chat Display Name
Sequence	1
Object Type	BusComp
Object Name	Service Request
Event	WriteRecordUpdated

3. Update the Chat Display Name Update Process workflow in Siebel Tools as follows:

a. Start Siebel Tools.
b. Locate, verify, and open the Chat Display Name Update Process workflow.

Note: Siebel Chat by default provides the Chat Display Name Update Process workflow, and you can modify it as required.

c. Configure the following methods as required in the following order:

- UpdateChatDataField. This method resets the ChatDisplayName in the Chat Data map to an empty string.

- Update ChatDataFieldEx. This method updates the field in the Chat Data map with the value in a business component field.
- UpdateWithBCField. This method updates a field in the Chat Data map with the value in a business component field.
- UpdateWithBCFieldEx. This method updates the field in the Chat Data map with the value in a business component field.
- UpdateWithChatData. This method updates a field in the Chat Data map with the value in the third-party Chat Data field.
- UpdateChatUIField. This method updates the chat display name in the chat pane.

For more information about each of these methods, see *Methods for Customizing Siebel Chat Process Flow and Interface Functionality*.

Configuring Keyboard Shortcuts for Buttons on the Siebel Chat UI

You can configure a keyboard shortcut for any button that is available on the text button bar or URL button bar of the Siebel Chat UI. After you configure the keyboard shortcut for a button, the shortcut key is displayed automatically in the tooltip for the button when you hover over the button.

To configure keyboard shortcuts for buttons on the Siebel Chat interface

1. Start Siebel Tools.
2. In the Object Explorer, select Applet, and then Chat Tab Applet.
3. In the Object Explorer, select Applet User Prop, then the Button control user property that you want to configure.

   For example, select the FUT button control user property to configure a keyboard shortcut for the Frequently Used Text button.
4. Configure the HotKey user property, for example, as shown in the following table.

Name	Value
HotKey	`<Alt+K>`

The value can be a combination of CTRL, ALT, or SHIFT plus a keyboard letter or function key. For more information about the shortcut key formats that are allowed, see the topic about creating accelerator objects in *Siebel Object Types Reference* .

Make sure that your keyboard shortcut key combination does not conflict with any command accelerators in Siebel Tools or with any existing Communication Toolbar keyboard shortcuts already defined. Any command accelerators in Siebel Tools or existing Communication Toolbar keyboard shortcuts already defined take precedence over the keyboard shortcuts defined for the chat interface.

Process of Configuring Siebel Chat Feedback

When agents are typing a response to an incoming chat, the following feedback message appears on the recipient's chat pane, provided that Siebel Chat feedback is configured:

```
Agent is typing . . .
```

Likewise when customers are typing a response to the agent, the following message appears on the agent's chat pane, provided Siebel Chat feedback is configured:

```
Customer is typing . . .
```

In both cases, the feedback message appears after the chat transcript area in the chat pane, and it is displayed for approximately 5 seconds before disappearing.

To configure feedback messages to display during chat sessions between agents and customers, perform the following tasks. This process is a step in *Roadmap for Configuring Siebel Call Center for Siebel Chat*.

- *Configuring User Preferences for Siebel Chat Feedback*
- *Configuring the SendAgentTypingMsg Command*
- *Configuring the ChatCustomerTyping Event Response*
- *Configuring the ChatCustomerTyping Event Handler*

Configuring User Preferences for Siebel Chat Feedback

The following system preferences must be configured in Siebel Call Center for Siebel Chat feedback:

- Chat: CustomerTypeDisplayTime. Indicates the time in seconds that the following feedback message is displayed on the agent's chat screen to indicate that the customer is typing a message response:

  ```
  Customer is typing
  ```

  After this time, provided the customer has stopped typing, the feedback message disappears.
- Chat: AgentTypeDetectInterval. Indicates the period in seconds during which to detect the agent's typing status. The value must be greater than zero.
- Chat: SendAgentTypeMsgCmd. Indicates the CTI command to call when sending the agent's feedback message to a customer. The CTI command must be a valid command used when configuring CTI (Computer Telephony Integration).

Use the following procedure to configure the user preferences in Siebel Call Center for Siebel Chat feedback. This task is step in *Process of Configuring Siebel Chat Feedback*.

To configure the user preferences for Siebel Chat feedback

1. Navigate to the Administration - Application screen, then the System Preferences view.
2. Query for **Chat** to get a list of all the chat-related system preferences that you can set up.
3. Configure the following system preferences with, for example, the values shown in the following table.

System Preference Name	Value	Comment
Chat: CustomerTypeDisplayTime	5	Indicates to show the feedback message (Customer is typing) for 5 seconds on the agent's chat screen.
Chat: AgentTypeDetectInterval	5	Indicates to show the feedback message (Agent is typing) for 5 seconds on the customer's chat screen.
Chat: SendAgentTypeMsgCmd	SendAgentTypingMsg	Indicates the CTI command to use to send the agent's feedback message to a customer.

Configuring the SendAgentTypingMsg Command

Use the following procedure to configure the SendAgentTypingMsg command in Siebel Call Center for Siebel Chat feedback. This task is step in *Process of Configuring Siebel Chat Feedback*.

To configure the SendAgentTypingMsg Command

1. Navigate to the Administration - Communications screen, then the All Configurations view.
2. Select the communications configuration record that you want to modify, then click the Commands tab.
3. Configure the SendAgentTypingMsg command as follows:

    a. Drill down into the SendAgentTypingMsg command, click the Command Parameters tab, and add the command parameters with the values shown in the following table.

Name	Value
CmdData	SendAgentTypingMsgCmdData
DeviceCommand	SendAgentTyping

    b. Click the Command Data Parameters tab, and add the command data parameter with the value shown in the following table.

Name	Value
Param.InteractionId	{InteractionId}

Name	Value

For more information about configuring commands and events, see *Siebel CTI Administration Guide* .

4. For the changes to take effect, log out of the Siebel application, and then log in again.

Configuring the ChatCustomerTyping Event Response

Use the following procedure to configure the ChatCustomerTyping event response in Siebel Call Center for Siebel Chat feedback. This task is step in *Process of Configuring Siebel Chat Feedback*.

To configure the ChatCustomerTyping event response

1. Navigate to the Administration - Communications screen, then the All Event Responses view.
2. In the Event Responses list, configure ChatCustomerTyping with the values shown in the following table.

Name	Configuration	Comment
ChatCustomerTyping	Select your communications configuration record. For example: Devchat_wkflow.	Add a relevant comment.

3. Click the Event Response Parameters tab, and configure the following records for ChatCustomerTyping with the values shown in the following table.

Name	Value
ServiceMethod	Chat UI Business Service.handleCustomerTyping
ServiceParam.InteractioId	{InteractionId}

Configuring the ChatCustomerTyping Event Handler

Use the following procedure to configure the ChatCustomerTyping event handler in Siebel Call Center for Siebel Chat feedback. This task is a step in *Process of Configuring Siebel Chat Feedback*.

To configure the ChatCustomerTyping event handler

1. Navigate to the Administration - Communications screen, then the All Event Handlers view.
2. In the Event Handlers list, configure ChatCustomerTyping with the values shown in the following table.

Name	Configuration	Event Response	Device Event	Order
ChatCustomerTyping	Select your communications configuration record. For example: Devchat_wkflow.	ChatCustomerTyping	InteractionMessage	0

3. Click the Event Handlers Parameters tab, and configure the following records for ChatCustomerTyping with the values shown in the following table.

Name	Value
Filter.Channel	Chat
Filter.MsgType	TYPE_CUSTOMER_TYPING

4. Navigate to Administration - Communications screen, Configurations view, and then click the Event Handlers tab.
5. Select the ChatCustomerTyping record, and configure it as follows:
   a. In the Comments field, modify or add a comment, for example, as follows:

      ```
      Comments: Chat
      ```

   b. In the Event Response field, click the Look Up icon, and associate the ChatCustomerTyping event response with the ChatCustomerTyping event handler.

6 Configuring Siebel Chat Auto Answer

Configuring Siebel Chat Auto Answer

This chapter describes how to configure Siebel Chat auto answer in Siebel Call Center. It contains the following topics:

- *Roadmap for Configuring Siebel Chat Auto Answer for Agents*
- *Process of Configuring Communications for Siebel Chat Auto Answer*

Note: The functionality described in this chapter applies only if you are using the default workflow to handle incoming chat requests. For more information about the default workflow for Siebel Chat, see *Chat Standard Accept Process Workflow*.

Roadmap for Configuring Siebel Chat Auto Answer for Agents

This topic describes how to enable the automatic acceptance of chat (hereafter referred to as chat auto answer) in Siebel Call Center for agents. Administrators can either configure chat auto answer for agents, or agents can configure it.

When chat auto answer is enabled for agents, agents do not have to manually accept chats that are offered to them. Instead, the chat that is offered automatically becomes the active chat session for the agent by opening on the agent's screen, provided that the agent has no other chat sessions assigned. The behavior is different if chat auto answer is enabled and the agent is busy handling one or more chat sessions. In these cases, the behavior is as follows:

- The chat that is offered does not automatically open on the agent's screen.
- The chat tab flashes to indicate that a new chat has been automatically accepted and queued to the agent. Administrators can configure the color that is used when the chat tab flashes.
- When the agent clicks the flashing chat tab, the chat session opens, and the chat tab stops flashing.
- If the customer decides to disconnect from the chat before the agent can open the chat session, then the chat closes as normal.

To configure Siebel Chat auto answer for agents in Siebel Call Center, perform the following tasks and process:

- *Configuring the Siebel Chat BlinkTab System Preference*
- (Optional) *Setting the Color for the Siebel Chat Blinking Tab*
- *Disabling the Service Integration Inbox Workflows*
- *Process of Configuring Communications for Siebel Chat Auto Answer*

Configuring the Siebel Chat BlinkTab System Preference

The BlinkTab system preference must be set to On for chat auto answer so that the chat tab flashes. This tab indicates that a new chat has been automatically accepted and queued. Use the following procedure to configure the BlinkTab

system preference for chat auto answer. This task is a step in *Roadmap for Configuring Siebel Chat Auto Answer for Agents*.

To configure the Siebel Chat BlinkTab system preference

1. Navigate to the Administration - Application screen, then the System Preferences view.
2. Query for *Chat* to get a list of all the chat-related system preferences you can set up.
3. Set the `Chat: BlinkTab` system preference to On.

   For information about other chat system preferences, see *Setting Up System Preferences in Siebel Call Center for Siebel Chat*.

Setting the Color for the Siebel Chat Blinking Tab

The color of the Siebel Chat blinking tab changes according to the status of incoming chat messages. By default, the color that the chat tab flashes is one of the following and these default colors apply to all Siebel Chat agents:

1. Green, which indicates a new automatically answered chat.
2. Yellow, which indicates that new chat messages are waiting.
3. Orange, which indicates that a new chat message is waiting for the new automatically answered chat.

To change the default colors that the chat tab flashes, complete the steps in the following procedure. This task is a step in *Roadmap for Configuring Siebel Chat Auto Answer for Agents*.

To set the color for the Siebel Chat blinking tab

1. Open the Siebel CRM cascading style sheet file.
2. Go to TabHoldNewColor parameter in .styleTitleTabBar.
3. Replace the color in the TabHoldNewcolor parameter with the new color that you want.

   To help you select the color that you want, go to *http://www.colorpicker.com* for color examples.
4. Clear your browser's cache, for example, as follows:
   a. Open Microsoft Internet Explorer.
   b. Click Tools, Internet Options, and then click Delete within Browsing History.
   c. Click Delete All on the Delete Browsing History dialog box and when prompted for confirmation, do the following:
      - Select the following check box: Also delete files and settings stored by add-on.
      - Click Yes.

Disabling the Service Integration Inbox Workflows

Use the following procedure to disable the service integration inbox workflows. This task is a step in *Roadmap for Configuring Siebel Chat Auto Answer for Agents*.

To disable inbox workflows

1. Navigate to the Administration - Business Process screen, then the Workflow Deployment view, and select the Active workflow process tab.

2. Query for and select the Inbox - Service Integration workflow.
3. Click Menu, then select Deactivate to deactivate the workflow.
4. Query for and select the Inbox - Service Main Integration workflow.
5. Click Menu, then select Deactivate to deactivate the workflow.

Process of Configuring Communications for Siebel Chat Auto Answer

To configure communications for Siebel Chat auto answer in Siebel Call Center, complete the following tasks:

- *Configuring the AutoAnswer System Parameter*
- *Configuring the Enable AutoAnswer User Preference at the Agent Level*
- *Configuring the ServiceParam.Work ID Event Log Parameter*
- *Configuring the Accept Chat Command Data Parameters for AcceptChatGroup*
- *Configuring the OnChatArrived Event Response*
- *Configuring the ChatArrived Event Handler*
- *Configuring the ChatTransferArrived Event Handler*

This process is a step in *Roadmap for Configuring Siebel Chat Auto Answer for Agents*.

Configuring the AutoAnswer System Parameter

The following procedure shows how to configure the AutoAnswer system parameter, which can be set to TRUE, FALSE, or User Preference. The following are the settings:

- If AutoAnswer is set to User Preference, then it indicates to defer to the Enable AutoAnswer user preference setting to determine whether or not to enable chat auto answer. For more information, see *Configuring the Enable AutoAnswer User Preference at the Agent Level*.
- If AutoAnswer is set to TRUE, then chat auto answer is enabled irrespective of whether or not the Enable AutoAnswer user preference is selected at the agent-level configuration.
- If AutoAnswer is set to FALSE, then chat auto answer is disabled irrespective of whether or not the Enable AutoAnswer user preference is selected at the agent-level configuration.

This task is a step in *Process of Configuring Communications for Siebel Chat Auto Answer*.

To configure the AutoAnswer system parameter

1. Navigate to the Administration - Communications screen, then the All Configurations view.
2. Select your communications configuration record from the Configurations list, then click the Parameters tab.
3. Set the value for the AutoAnswer parameter as shown in the following table.

Parameter	Value	Active
AutoAnswer	User Preference	Selected

Configuring the Enable AutoAnswer User Preference at the Agent Level

Use the following procedure to configure the Enable AutoAnswer user preference at the agent level in Siebel Call Center. The Enable AutoAnswer user preference setting determines whether to enable or disable chat auto answer when the AutoAnswer system parameter is set to User Preference. This task is a step in *Process of Configuring Communications for Siebel Chat Auto Answer*.

To configure the Enable Auto Answer user preference at the agent level

1. Select Tools, User Preferences, and then the Communications option.
2. Do one of the following:
   - Select the Enable AutoAnswer checkbox to enable chat auto answer at the agent level.
   - Deselect the Enable AutoAnswer checkbox to disable chat auto answer at the agent level.

   The Enable AutoAnswer user preference setting is relevant only if the AutoAnswer system parameter is set to User Preference. If AutoAnswer is not set to User Preference, then Enable AutoAnswer is ignored.
3. Click Save.

Configuring the ServiceParam.Work ID Event Log Parameter

The ServiceParam.WorkID event log parameter must be configured for ChatAcceptedLog and ChatTransferAcceptedLog.

To configure the ServiceParam.WorkID event log parameter

1. Navigate to the Administration - Communications screen, then the All Event Logs view.
2. In the Event Logs list, configure ChatAcceptedLog with the value shown in the following table.

Name	Configuration
ChatAcceptedLog	Select your communications configuration record.

3. Click the Event Log Parameters tab, and add a new record for ChatAcceptedLog with the values shown in the following table.

Name	Value
ServiceParam.WorkID	`{SiebelWorkItemID}`

4. Go back to the Event Logs list, and configure ChatTransferAcceptedLog with the value shown in the following table.

Name	Configuration
ChatTransferredAcceptedLog	Select your communications configuration record.

5. Click the Event Log Parameters tab, and add a new record for ChatTransferAcceptedLog with the values shown in the following table.

Name	Value
ServiceParam.WorkID	`{SiebelWorkItemID}`

Configuring the Accept Chat Command Data Parameters for AcceptChatGroup

The following procedure shows how to configure the accept chat command data parameters (AcceptChatCmdData) for AcceptChatGroup commands.

To configure the accept chat command data parameters for AcceptChatGroup

1. Navigate to the Administration - Communications screen, then the All Command Data view.
2. In the Command Data list, configure AcceptChatCmdData with the values shown in the following table.

Name	Configuration	Comment
AcceptChatCmdData	Select your communications configuration record.	Enter a comment, for example: Add for chat auto answer.

3. Click the Command Data Parameters tab, and configure the following records for AcceptChatCmdData with the values shown in the following table.

Name	Value
Param.AutoHold	`FALSE`
Param.TrackingID	`{InteractionId}`

4. Navigate to the Administration - Communications screen, then the All Configurations view, select your communications configuration record from the Configurations list, and then do the following:

a. Click the Commands tab, and configure the AcceptChatGroup command with the value shown in the following table.

Name	Command Data Name
AcceptChatGroup	AcceptChatCmdData

b. Click the Parameters tab, and verify that the command data parameters have the Param.TrackingID that you configured earlier in this procedure.

Configuring the OnChatArrived Event Response

Use the following procedure to configure the OnChatArrived event response.

To configure the OnChatArrived event response

1. Navigate to the Administration - Communications screen, then the All Event Responses view.
2. In the Event Responses list, configure OnChatArrived with the values shown in the following table.

Name	Configuration	Comment
OnChatArrived	Select your communications configuration record.	Enter a comment, for example: Add for chat auto answer.

3. Click the Event Response Parameters tab, and configure the following records for OnChatArrived with the values shown in the following table.

Name	Value
ServiceMethod	`Communications Client.TryAutoAnswer`
ServiceParam.Command	`AcceptChatGroup`
ServiceParam.InteractionId	`{InteractionId}`

Configuring the ChatArrived Event Handler

Use the following procedure to configure the ChatArrived event handler.

To configure the ChatArrived event handler

1. Navigate to the Administration - Communications screen, then the All Event Handlers view.
2. In the Event Handlers list, configure ChatArrived with the values shown in the following table.

Name	Configuration	Event Response	Device Event	Order
ChatArrived	Select your communications configuration record.	OnChatArrived	InteractionMessage	0

3. Click the Event Handlers Parameters tab, and configure the following records for ChatArrived with the values shown in the following table.

Name	Value
Filter.Channel	`Chat`
Filter.MsgType	`TYPE_NEW`
Filter.systemTransfer	`0`

4. Navigate to Administration - Communications screen, Configurations view, and then click the Event Handlers tab.
5. Select the ChatArrived record, and configure it as follows:
   a. In the Comments field, modify or enter a comment, for example, as follows:
      ```
      Comments: Chat
      ```
   b. In the Event Response field, click the Look Up icon, and associate the OnChatArrived event response with the ChatArrived event handler.

Configuring the ChatTransferArrived Event Handler

Use the following procedure to configure the ChatTransferArrived event handler.

To configure the ChatTransferArrived event handler

1. Navigate to the Administration - Communications screen, then the All Event Handlers view.
2. In the Event Handlers list, configure ChatTransferArrived with the values shown in the following table.

Name	Configuration	Event Response	Device Event	Order
ChatTransferArrived	Select your communications configuration record.	OnChatArrived	InteractionMessage	1

Name	Configuration	Event Response	Device Event	Order

3. Click the Event Handlers Parameters tab, and configure the following records for ChatTransferArrived with the values shown in the following table.

Name	Value
Filter.Channel	`Chat`
Filter.MsgType	`TYPE_NEW`
FilterSpec	`[systemTransfer]>0`

4. Navigate to the Administration - Communications screen, Configurations view, and then click the Event Handlers tab.
5. Select the ChatTransferArrived record, and configure it as follows:
   a. In the Comments field, modify or add a comment, for example, as follows:

      ```
      Comments: Chat
      ```

   b. In the Event Response field, click the Look Up icon, and associate the OnChatArrived event response with the ChatTransferArrived event handler.

7 Localizing Siebel Chat

Localizing Siebel Chat

This chapter describes Siebel Chat localization. It contains the following topics:

- *Setting Up Siebel Chat for Languages Supported by Third-Party Chat Solutions*
- *Setting Up Siebel Chat for Languages Not Supported by Default by Third-Party Chat Solutions*

Setting Up Siebel Chat for Languages Supported by Third-Party Chat Solutions

Siebel Chat currently supports all languages supported by Siebel applications. For a complete list of languages supported by Siebel applications, see the Certifications tab on My Oracle Support. For information about Certifications, see article 1492194.1 (Article ID) on My Oracle Support.

The Chat Language Mapping view in Siebel Call Center is used to map third-party chat language IDs to standard Siebel Call Center language code values.

To set up Siebel Chat for languages supported by third-party chat solutions, map the correct third-party chat language ID to the appropriate Siebel Call Center language code, as described in the following procedure.

To set up default chat language mapping

1. Navigate to the Administration - Communications screen, then the Chat Language Mapping view in Siebel Call Center.
2. Map the correct third-party chat language ID to the appropriate Siebel Call Center language code.

Adding New Language Mappings in Siebel Call Center

To add a new language mapping in Siebel Call Center (for example, if translating Siebel Chat to a new language), complete the steps in the following procedure.

To add a new language mapping in Siebel Call Center

1. Navigate to the Administration - Communications screen, then the Chat Language Mapping view in Siebel Call Center.
2. Click Menu and then New record to add a new language mapping.

Setting Up Siebel Chat for Languages Not Supported by Default by Third-Party Chat Solutions

The following tasks must be carried out if setting up Siebel Chat for languages not supported by default by third-party chat solutions:

1. Localize the unshipped language for Siebel Call Center, if necessary.

   For more information about adding an unshipped language to Siebel applications, see the topic about localizing an unshipped language in *Siebel Global Deployment Guide* .
2. Localize the Siebel Customer Chat UI in the third-party chat server. This involves the following subtasks:
   - Translate the resource file used by the Customer Chat UI
   - Add the new language mappings to the third-party chat language-locales table
3. Map the third-party chat Language ID to the appropriate Siebel applications language code.

8 Siebel Chat Deployment Using Third-Party Chat Solutions

Siebel Chat Deployment Using Third-Party Chat Solutions

This chapter describes how to deploy Siebel Chat using a third-party chat solution. It contains the following topics:

- *Deployment of Siebel Chat Using Third-Party Chat Solutions*

Deployment of Siebel Chat Using Third-Party Chat Solutions

Several additional tasks must be carried out if deploying Siebel Chat using a third-party chat solution (such as Genesys). In particular, third-party chat server vendors must integrate with Siebel Chat by addressing the following tasks:

- Create or extend an existing Siebel Communications API (SCAPI) driver.

  This task involves adding Siebel Chat to an existing driver, or creating a new driver for a third-party chat communication server. For information about developing and testing a new communications driver, see *Siebel CTI Administration Guide* .
- Take the existing third-party customer chat client and modify it to support the passing of Siebel Chat extended parameters. The following table lists the chat parameter extensions, Field Name and Field Size, that must be set up for proper integration with the Siebel Call Center application.
- Integrate with the Chat UI Business Service in order to process chat events from the agent-facing chat UI. The Chat UI business service is responsible for handling all Siebel Chat commands and events, and manages the state of all active chats.

Field Name	Description	Field Size
ContactId	Contact Identifier	15
SRId	Service Request ID	15
SRNumber	Service Request Number	30
ProblemDesc	Problem Description	100
Area	Service Request Area	30

Field Name	Description	Field Size
SubArea	Service Request Sub Area	30
ProductID	Product Identifier	15
ProductName	Product Name	100
ProductCategory	Product Category	100
KBItemVisitedNum1	KBItemVisitedNum1	30
KBItemVisitedDesc1	KBItemVisitedDesc1	100
KBItemVisitedURL1	KBItemVisitedURL1	100
KBItemVisitedNum2	KBItemVisitedNum2	30
KBItemVisitedDesc2	KBItemVisitedDesc2	100
KBItemVisitedURL2	KBItemVisitedURL2	100
KBItemVisitedNum3	KBItemVisitedNum3	30
KBItemVisitedDesc3	KBItemVisitedDesc3	100
KBItemVisitedURL3	KBItemVisitedURL3	100
KBItemVisitedNum4	KBItemVisitedNum4	30
KBItemVisitedDesc4	KBItemVisitedDesc4	100
KBItemVisitedURL4	KBItemVisitedURL4	100
KBItemVisitedNum5	KBItemVisitedNum5	30
KBItemVisitedDesc5	KBItemVisitedDesc5	100
KBItemVisitedURL5	KBItemVisitedURL5	100
KBItemVisitedNum6	KBItemVisitedNum6	30
KBItemVisitedDesc6	KBItemVisitedDesc6	100

Field Name	Description	Field Size
KBItemVisitedURL6	KBItemVisitedURL6	100
KBItemVisitedNum7	KBItemVisitedNum7	30
KBItemVisitedDesc7	KBItemVisitedDesc7	100
KBItemVisitedURL7	KBItemVisitedURL7	100
KBItemVisitedNum8	KBItemVisitedNum8	30
KBItemVisitedDesc8	KBItemVisitedDesc8	100
KBItemVisitedURL8	KBItemVisitedURL8	100
KBItemVisitedNum9	KBItemVisitedNum9	30
KBItemVisitedDesc9	KBItemVisitedDesc9	100
KBItemVisitedURL9	KBItemVisitedURL9	100
KBItemVisitedNum10	KBItemVisitedNum10	30
KBItemVisitedDesc10	KBItemVisitedDesc10	100
KBItemVisitedURL10	KBItemVisitedURL10	100

9 Methods for Customizing Siebel Chat Process Flow and Interface Functionality

Methods for Customizing Siebel Chat Process Flow and Interface Functionality

This chapter describes the methods that are available for customizing Siebel Chat process flow and interface functionality, including the chat screen pop-up logic. It contains the following topic:

- *Methods for Customizing Siebel Chat Process Flow and Interface Functionality*

Methods for Customizing Siebel Chat Process Flow and Interface Functionality

The following methods are available to support the customization of the Siebel Chat UI and the chat screen pop-up logic, which allows you to use workflows to design the chat screen pop-up process:

- *AcceptChatEx*
- *FindRecord*
- *FindRecordEx*
- *GetActiveChatActId*
- *GetChatDataByField*
- *GetLanguageCodeFromMapping*
- *GetSystemPreference*
- *GetTransferredActivityDataFromDB*
- *InsertRecord*
- *OpenView*
- *OpenViewEx*
- *PostScreenPopRequest*
- *PreScreenPopRequest*
- *PushToToolbarCallFunction*
- *RefreshChatPaneDashboard*
- *RestoreMainViewFromBookmark*
- *UpdateChatDataField*
- *UpdateChatDataFieldEx*
- *UpdateChatUIField*
- *UpdateRecord*

- *UpdateWithBCField*
- *UpdateWithBCFieldEx*
- *UpdateWithChatData*

AcceptChatEx

This method contains the update for the input argument. It is a newer version of the existing AcceptChat method. It is used whenever a new or transferred chat is accepted to open the chat pane and dashboard and to update the activity information, such as the Contact ID and Service Request ID. The following table shows the input PropertySet arguments for AcceptChatEx.

Input Argument	Description
`ActivityId`	The row ID of the activity created by the CTI event log.
`InteractionId`	The interaction ID of the active chat interaction.
`SRId`	The row ID of the service request record.
`AccountId` (Optional)	The row ID of the account record.
`ContactId` (Optional)	The row ID of the contact record.
`isTransfer` (Optional)	A value of 1 indicates a transferred chat. Any value other than 1 indicates a new inbound chat.
`Firstname` (Optional)	The first name of the customer.
`Lastname` (Optional)	The last name of the customer.
`WorkID`	The work ID of the active chat interaction.

The following table shows the output PropertySet argument for AcceptChatEx.

Output Argument	Description
`HasBookmark`	A value of True indicates the following: • That this chat has the bookmark feature enabled • That the chat screen is populated with the bookmark information • That this chat is a transferred chat

FindRecord

This method contains the update for the input argument. It is used to query records from a specific business component according to the search specification. You can define the business object name, business component name, query field value pairs, and value field list. By default, the row ID of the query results is returned as output, but you can define more output by using the input value field list. The following table shows the input PropertySet arguments for FindRecord.

Input Argument	Description
`BusObj`	The business object name.
`BusComp`	The business component name.
`QueryFields`	A comma-separated field list. This list indicates to use the field name list to query the business component. A corresponding key value pair must also be defined for the query search specification. For example, if you define Query Fields as follows **`(QueryFields = First Name, Last Name)`**, then you also must provide a list of key value pairs: **`First Name = John`** **`Last Name = Smith`** First Name and Last Name are the query field names. John and Smith are the corresponding field values used for the query.
`ValueFields` (Optional)	A comma-separated field list, indicating the output field name list. The default functionality is to generate the row IDs of the query results, but you use this parameter to generate the other fields as required.

The following table shows the output PropertySet arguments for FindRecord, which return the row IDs and value fields defined by the input of the query results.

Output Argument	Description
`Id`	The row ID of the first query record. You can query for more than one record by using the input arguments, but this argument saves only the row ID of the first record.
`IsMultipleRecord`	A value of True indicates that the number of query records is greater than one.
`IdList`	A comma-separated row ID list of the query results.
`<key, value>` pair	According to the values in the ValueFields input parameter, this method generates more field values of the first query record.

Example

If you want to query a contact by contact name and generate the contact row ID and work phone number, then the input is as follows:

```
BusObj = Contact
BusComp = Contact
QueryFields = First Name, Last Name, xxx,
First Name = John
Last Name = Smith
Xxxx =
ValueFields = Work Phone #
```

FindRecordEx

This method contains the update for the input argument. This method is an enhanced version of the FindRecord method and is used to query for multiple levels of business components. You can use this method to query for the child business component according to the query results of the parent business components. The following table shows the input PropertySet arguments for FindRecordEx.

Input Argument	Description
`BusObj`	The business object name.
`BusComp`	A comma-separated business component name list. **Note:** The order is important. The child business component must appear after the parent business component.
`<BusComp>:QueryField:XXX`	This is the query field of the specific business component. The business component is listed in the BusComp parameter. In the query, *XXX* is the query field name of the business component. For example, to query using the row ID of the Business Component Account, the argument values are as follows: `Account:QueryField:Id`
`<BusComp>:ValueFields` (Optional)	A comma-separated field list indicating the output field name list of the specific business component. The default functionality is to generate the row IDs of the query results, but you can use this parameter to generate the other fields as required.

The output PropertySet for FindRecordEx includes the query results of all the business components defined by the BusComp parameter. The query result for each business component is organized as a child PropertySet with the type set to the business component name.

Example

If you want to query for a contact that belongs to a specific Account and the account has many associated contacts, then you must use the work phone number to query the contact too. In this case, the input is as follows:

```
BusObj = Account
BusComp = Account, Contact Account:QueryField:Id = 42-FF6I6
```

```
Contact:QueryField: Work Phone # = 123456789
```

GetActiveChatActId

This method is used to get the action row ID of the current active chat. The following table shows the output argument for GetActiveChatActId.

Output Argument	Description
`ActivityId`	Returns the row ID of the current active chat.

GetChatDataByField

This method is used to get the chat data by field name. The following table shows the input arguments for GetChatDataByField.

Input Argument	Description
`InteractionId`	If an InteractionId is provided, then it indicates to set data to the related chat. If an InteractionId is not provided, then it indicates to set data to the current chat.
`FieldList`	The data field names that you want to get, separated by a comma.

For each field name in FieldList, there is a corresponding output argument with the same name.

GetLanguageCodeFromMapping

This method is a script method, and it is used to get the language code from the user defined mapping. The following table shows the input argument for GetLanguageCodeFromMapping.

Input Argument	Description
`RequestSiteID`	The ID for the request site; that is, the customer site requesting to chat to an agent.

The following table shows the output argument for GetLanguageCodeFromMapping.

Output Argument	Description
`Language`	The language code for the request site. An example of the method code is shown after this table.

Output Argument	Description

The following is an example of the method code:

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 if (MethodName == "GetLanguageCodeFromMapping")
{
 var sRequestSiteID = "";
 sRequestSiteID = Inputs.GetProperty("RequestSiteID");
 switch(sRequestSiteID)
 {
 case "0":
 Outputs.SetProperty ("LanguageCode", "ENU");
 break;
 case "101":
 Outputs.SetProperty ("LanguageCode", "ITA");
 break;
 case "71":
 Outputs.SetProperty ("LanguageCode", "FRA");
 break;
 default:
 Outputs.SetProperty ("LanguageCode", "ENU");
 break;
 }
 return CancelOperation;
 }
 return (ContinueOperation);
}
```

GetSystemPreference

This method is used to query for the system preference. You must insert the system preference name, and then the value is set to the output argument. The following table shows the input PropertySet arguments for GetSystemPreference.

Input Argument	Description
`Name`	The name of the system preference item.

The following table shows the output PropertySet argument for GetSystemPreference.

Output Argument	Description
`Value`	The value of the system preference.

GetTransferredActivityDataFromDB

This method is used to query the parent activity record. The query is based on the interaction ID, and it applies only to chat transfers. Generally, this method queries for the service request ID when a second agent accepts a chat. The following table shows the input PropertySet arguments for GetTransferredActivityDataFromDB.

Input Argument	Description
`InteractionId`	The interaction ID for the chat.
`QuerySpec` (Optional)	This optional input argument applies if you want to redefine your own query specifications.
`ValueFields`	This is a comma-separated field list that indicates the output field name list. If no Value Fields are defined, then the default value is Activity SR Id.

The output PropertySet for GetTransferredActivityDataFromDB contains the key value pairs that are defined by the ValueFields input parameter.

Example

When an agent accepts a transferred chat, the service request ID that Siebel automatically creates must be located. The Chat Business Service by default uses the interaction to query the activity created by the first agent (who transferred the chat), and then to query for that activity to obtain the service request ID. The input is as follows:

```
InteractionId = 1234423545
ValueFields = Id,Primary Contact Id,Activity SR Id
```

InsertRecord

This method is used to insert a new record into a specific business component. You can define the business object name, business component name, and field value pairs to be inserted as input. The row ID of the newly created record is returned as output. The following table shows the input PropertySet arguments for InsertRecord.

Input Argument	Description
`BusObj`	The business object name.
`BusComp`	The business component name.
`Field: XXX`	The fields to be set. In this argument, you must replace *XXX* with the field name.
`Use Locale` (Optional)	A value of True or False indicates whether to format the value or not.

The following table shows the output PropertySet argument for InsertRecord, which returns the row ID of the newly created record.

Output Argument	Description
`Id`	The row ID of the newly created record.

Example

If you want to insert a new service request record with a contact ID of 04-LLSQ5, then the input is as follows:

```
BusObj = Service Request
BusComp = Service Request
Field: Contact Id = 04-LLSQ5
```

OpenView

This method is used to open a chat screen pop-up according to the following input information: business component and view name. This method cannot be used directly and must always be called using the PushToToolbarCallFunction method. The following table shows the input PropertySet arguments for OpenView.

Input Argument	Description
`ViweBusObj`	The view's business object name.
`ViewBusComp`	The view's business component name.
`Id`	The row ID of the record to be populated.
ViewName	The name of the view to be populated. **Note:** The ViewName must be associated with the business component defined in ViewBusComp.
`SearchMode`	The value of this argument can be OFF, ON, or ALL: • If the value is set to OFF, then the search center is not populated. • If the value is set to ON, then the search center is populated when there is no query result according to the input ID. • If the value is set to ALL, then the search center is always populated irrespective of whether the query result is empty or not.
`ViewNameEx` (Optional)	The default view name if AllowEmptyRecord is not set, or if it is set to False when the ID is empty or invalid. The view must be associated with the business object defined in ViewBusObjEx.
`ViewBusObjEx` (Optional)	The view's business object name for ViewNameEx.

Input Argument	Description
ViewBusCompEx (Optional)	The view's business component name for ViewNameEx.
AllowEmptyRecord (Optional)	If set to True when the ID is empty or invalid, then it indicates to go to the same view but with an empty active record. If set to False, then it indicates to go to the view according to the values in ViewNameEX, ViewBusObjEx, and ViewBusCompEx.

Example

If you want to open the service request detail view by using the service request ID 42-DOOTG and the user wants to open the search center when there is no such service request with ID 42-DOOTG in the database, then the input is as follows:

```
ViewBusObj = Service Request
ViewBusComp = Service Request
Id = 42-DOOTG
ViewName = Service Request Detail View
SearchMode = ON
AllowEmptyRecord = FALSE
ViewBusObjEx = Service Request Home
ViewBusCompEx = Public and Private View Link
ViewNameEx = Service Request Screen Homepage View
```

OpenViewEx

This method is used to obtain the chat screen pop-up according to the following input information: business component and view name. This method shows the record list, and it can be used to query more than one field besides the row ID. The following table shows the input arguments for OpenViewEx.

Input Argument	Description
ViewName	The name of the view to be populated. The view must be associated with the defined business component.
ViewBusObj	The view's business object name.
ViewBusComp	The view's business component name.
SearchMode	The value of this argument can be OFF, ON, or ALL. • If set to OFF, then the search center is never populated. • If set to ON, then the search center is populated when there is no query result according to the input ID. • If set to ALL, then the search center is always populated irrespective of whether the query result is empty or not.

Input Argument	Description
`QueryFields`	The comma-separated list of field names, which you want to use to query the view business component. For each field, there is an input argument for the query value.
`NonEmptyQueryFields`	The comma-separated list of query field names for which the query value cannot be empty. If the query value for one of the fields is set to " ", then the query fails and no results are returned.

PostScreenPopRequest

This method is used to carry out the postrequest for the chat screen pop-up. The following table shows the input PropertySet arguments for PostScreenPopRequest.

Input Argument	Description
`InteractionId`	The interaction ID for the chat.

PreScreenPopRequest

This method is used to carry out the prerequest for the chat screen pop-up. The following table shows the input PropertySet arguments for PreScreenPopRequest.

Input Argument	Description
`InteractionId`	The interaction ID for the chat.

The following table shows the output PropertySet argument for PreScreenPopRequest.

Output Argument	Description
`HasBookmark`	Indicates whether or not a bookmark is enabled for the chat. A value of True indicates that the chat has the bookmark feature enabled.

PushToToolbarCallFunction

This method is used to call a chat screen pop-up method, such as OpenView or RestoreMainViewFromBookmark. This method is used, for example, if there are several asynchronous updates that must be made to the UI. By using

PushToToolbarCallFunction to call chat screen pop-up methods, you ensure that all the methods are called one-by-one. The following table shows the input PropertySet arguments for PushToToolbarCallFunction.

Input Argument	Description
`MethodName`	The name of the chat screen pop-up method to be called, such as OpenView.
`InteractionId` (Optional)	The interaction ID for the chat. If IsQueued is set to True, then you must provide the InteractionID.
`IsQueued` (Optional)	Indicates whether or not to queue the method. The method name is MethodName.

Except for this parameter, you must also enter all the necessary parameters for the chat screen pop-up method to be called later.

RefreshChatPaneDashboard

This method is used to refresh the business component fields on the chat dashboard with the latest data. There are no input or output arguments for RefreshChatPaneDashboard.

RestoreMainViewFromBookmark

This is an existing method that is used to populate screens using bookmarks. No inputs or outputs must be defined for this method.

UpdateChatDataField

This method is used to update the field in the Chat Data map with a given value. The following table shows the input arguments for UpdateChatDataField.

Input Argument	Description
`InteractionId` (Optional)	The value in this argument is the interaction ID that you want to update if IsSingleChat is set to True. The value in this argument is ignored if IsSingleChat is set to False. For new and transferred chats, IsSingleChat is always set to True.
`IsSingleChat` (Optional)	If set to True, then it indicates to update the value provided for InteractionId. If set to False, then it indicates to update the value for all chats. For new and transferred chats, IsSingleChat is always set to True.
`UpdateField`	The field that you want to update in the Chat Data map.

Input Argument	Description
UpdateValue (Optional)	The value that you want to update in the UpdateField. If UpdateValue is not provided, then an empty string is used by default.

Example

If you want to update the ChatDisplayName field in the Chat Data map with the default value for all chats, then the input is as follows:

```
InteractionId =
IsSingleChat = FALSE
UpdateField = ChatDisplayName
UpdateValue =
```

UpdateChatDataFieldEx

This method is used to update the field in the Chat Data map with the value in a business component field. This method is typically used as a preceding step for the UpdateWithBCFieldEx method. For example, if QueryField is set to the SR Number business component field in UpdateWithBCFieldEx and to the SRNumber value in the Chat Data map, then the SRNumber value in Chat Data map expires if the agent associates a new service request with the chat activity. As a result, you must update the SRNumber value in the Chat Data map before SR Number is used in UpdateWithBCFieldEx. The following table shows the input arguments for UpdateChatDataFieldEx.

Input Argument	Description
InteractionId (Optional)	The value in this argument is the interaction ID that you want to update if IsSingleChat is set to True. The value in this argument is ignored if IsSingleChat is set to False. For new and transferred chats, IsSingleChat is always set to True.
IsSingleChat (Optional)	If set to True, then it indicates to update the value provided for InteractionId. If set to False, then it indicates to update the value for all chats. For new and transferred chats, IsSingleChat is always set to True.
BusObj	The business object, for example, Action.
BusComp	The business component in the business object, for example, Action.
QueryFields	The Query Field list to use to query the value in ValueField (for example, Id). You must provide the value in Chat Data for the query field. For example, provide the Activity Id in Chat Data for Id. The comma-separated field list indicates the field name list to use to query the business component. For query search specification, you must also define a corresponding key value pair. For example, if QueryFields is set to Id and Contact First Name, then a list of key value pairs must also be provided as follows: **Id=Activity Id** **Contact First Name = firstname**

Input Argument	Description
	Id and Contact First Name are the query business component field names. Activity Id and firstname are the corresponding Chat Data fields.
`UpdateField`	The field that you want to update in the Chat Data map.
`Value Field`	The business component field that you want to get the value from.

Example

If you want to update the SRNumber field in Chat Data map with the SR Number value in the Action business component, Id is used as the query filter and is set to the value of the ActivityId field in Chat Data map. That is, the ActivityId value in Chat Data map is used as the query filter to query the Action business component record to obtain the SR Number value in the Action business component record. The SR Number value is then used to update the SR Number field in Chat Data map, so that it has the latest value if a new service request is associated with the chat activity.

The ActivityId field value in Chat Data map is not always changed in the chat session, but the SRNumber field value in Chat Data map is changed if a new service request is associated with the chat activity. Therefore, the UpdateChatDataFieldEx method is used to update SRNumber before the SRNumber field value in Chat Data map is used in the next workflow step. To update SRNumber for all chats, the input is as follows:

```
InteractionId =
IsSingleChat = FALSE
BusObj = Action
BusComp = Action
QueryFields = Id // the field of Activity Id in Action BC
Id = ActivityId // the field of Activity Id in Chat Data map
UpdateField = SRNumber // the field of SR Number in Chat Data map
ValueField = SR Number // the field of SR Number in Action BC
```

UpdateChatUIField

This method is used to update a field in the chat UI with the value in the Chat Data map field. The following table shows the input arguments for UpdateChatUIField.

Input Argument	Description
`InteractionId` (Optional)	The value in this argument is the interaction ID that you want to update if IsSingleChat is set to True. The value in this argument is ignored if IsSingleChat is set to False. For new and transferred chats, IsSingleChat is always set to True.
`IsSingleChat` (Optional)	If set to True, then it indicates to update the value provided for InteractionId. False indicates to update the value for all chats. For new and transferred chats, IsSingleChat is always set to True.
`UpdateField`	This is the field that you want to get the value from in Chat Data map. For example: ChatDisplayName in Chat Data map.

Example

If you want to use the ChatDisplayName field in the Chat Data map to update the Chat Display Name in the chat UI for all chats, then the input is as follows. For example, if a user wants to use Chat Data map field, ChatDisplayName, to update Chat Display Name in the Chat UI for all chats, then the input is as follows:

```
InteractionId =
IsSingleChat = FALSE
UpdateField = ChatDisplayName // the field of ChatDisplayName in Chat Data map
```

UpdateRecord

This method is used to update an existing record and a specific business component. You must provide the row ID as input and the field value pairs to be updated. The following table shows the input arguments for UpdateRecord.

Input Argument	Description
`BusObj`	The business object name.
`BusComp`	The business component name.
`Id`	The row ID of the record to be updated.
`Field: XXX`	The fields to be updated; replace *XXX* with the field name.
`Use Locale` [optional]	A value of True or False indicates if the value must be formatted.

The following table shows the output PropertySet argument for UpdateRecord, which returns the row ID of the newly created record.

Output Argument	Description
`Id`	The row ID of the newly created record.

UpdateWithBCField

This method is used to update a field in the Chat Data map with the value in a business component field. The business component must be included in the business object component of the *Chat Related Data* business object. For the Service Request business component, there must be a link with Action. For other business components, there must be a link with Service Request. The following table shows the input arguments for UpdateWithBCField.

Input Argument	Description
InteractionId (Optional)	The value in this argument is the interaction ID that you want to update if IsSingleChat is set to True. The value in this argument is ignored if IsSingleChat is set to False. For new and transferred chats, IsSingleChat is always set to True.
IsSingleChat (Optional)	If set to True, then it indicates to update the value provided for InteractionId. False indicates to update the value for all chats. For new and transferred chats, IsSingleChat is always set to True.
BusComp	The business component list that is added to the business object component in the Chat Related Data business object. For example: **Service Request, Account**
UpdateField	The field that you want to update in the Chat Data map.
UpdateValue (Optional)	The business component field that you want to get the value from.

The following table shows the output arguments for UpdateWithBCField.

Input Argument	Description
HasSetForAll (Optional)	Indicates whether or not to set (or update) the value for all chats. If set to True, then all chats get the update value.

Example

If you want to update ChatDisplayName in the Chat Data map with the Service Request Business Component Account field for all chats, then the Service Request Business Component must be included in the business object component in the Chat Related Data Business Object, and it must be linked with Action. The input is as follows:

```
InteractionId =
IsSingleChat = FALSE
BusComp = Service Request
UpdateField = ChatDisplayName
ValueField = Account
```

UpdateWithBCFieldEx

This method is used to update the field in the Chat Data map with the value in a business component field. This method is typically used together with the *UpdateChatDataFieldEx*. The following table shows the input arguments for UpdateWithBCFieldEx.

Input Argument	Description
InteractionId (Optional)	The value in this argument is the interaction ID that you want to update if IsSingleChat is set to True. The value in this argument is ignored if IsSingleChat is set to False.

Input Argument	Description
	For new and transferred chats, IsSingleChat is always set to True.
`IsSingleChat` (Optional)	If set to True, then it indicates to update the value provided for InteractionId. False indicates to update the value for all chats. For new and transferred chats, IsSingleChat is always set to True.
`BusObj`	The business object, for example, Service Request.
`BusComp`	The business component name in the business object, for example, Service Request.

`QueryFields`	The Query Field list that is used to query the value in ValueField (for example, SR Number). You must provide the value in Chat Data for the query field. For example, provide the SR Number in Chat Data for SR Number in the business component field.
	
	The comma-separated field list indicates the field name list to use to query the business component.
	
	For query search specification, you must also define a corresponding key value pair. For example, if QueryFields is set to Id and SR Number, then a list of key value pairs must also be provided as follows:
	
	`Id=Activity Id`
	
	`SR Number = SRNumber`
	
	Id and SR Number are the query business component field names. Activity Id and SRNumber are the corresponding Chat Data fields.

`UpdateField`	The field that you want to update in the Chat Data map.
`Value Field`	The business component field that you want to get the value from.

The following table shows the output arguments for UpdateWithBCFieldEx.

Input Argument	Description
`HasSetForAll` (Optional)	Indicates whether or not to set (or update) the value for all chats. If set to True, then all chats get the update value.

Example

You want to update ChatDisplayName in Chat Data map with the Service Request Business Component Account field for all chats, and you do not want to use the Chat Related Data Business Object in the method. For every business component field in QueryFields, you must provide the related Chat Data map field to give the query value. In this example, the QueryFields is the SR Number business component field, so you must provide the related SRNumber field in the Chat Data map, so that it can get and use the SRNumber field value as the QueryField value. The input is as follows:

```
InteractionId =
IsSingleChat = FALSE
BusObj = Service Request
BusComp = Service Request
```

```
QueryFields = SR Number // the field of SR Number in Action BC
SR Number = SRNumber // the field of SRNumber in Chat Data map
UpdateField = ChatDisplayName // the field of ChatDisplayName in Chat Data map
ValueField = Account // the field of Account in Action BC
```

UpdateWithChatData

This method is used to update a field in the Chat Data map with the value in the (third-party chat) Chat Data field. The following table shows the input arguments for UpdateWithChatData.

Input Argument	Description
`InteractionId` (Optional)	The value in this argument is the interaction ID that you want to update if IsSingleChat is set to True. The value in this argument is ignored if IsSingleChat is set to False. For new and transferred chats, IsSingleChat is always set to True.
`IsSingleChat` (Optional)	If set to True, then it indicates to update the value provided for InteractionId. False indicates to update the value for all chats. For new and transferred chats, IsSingleChat is always set to True.
`UpdateField`	The field that you want to update in the Chat Data map.
`Value Field`	The business component field that you want to get the value from.

The following table shows the output arguments for UpdateWithChatData.

Input Argument	Description
`HasSetForAll` (Optional)	Indicates whether or not to set (or update) the value for all chats. If set to True, then all chats get the update value.

Example

If you want to update ChatDisplayName in the Chat Data map with the (third-party chat) firstname field for all chats, then the input is as follows:

```
InteractionId =
IsSingleChat = FALSE
UpdateField = ChatDisplayName // the field of ChatDisplayName in Chat Data map
ValueField = firstname // the field of firstname in Chat Data map
 (third--party chat data)
```