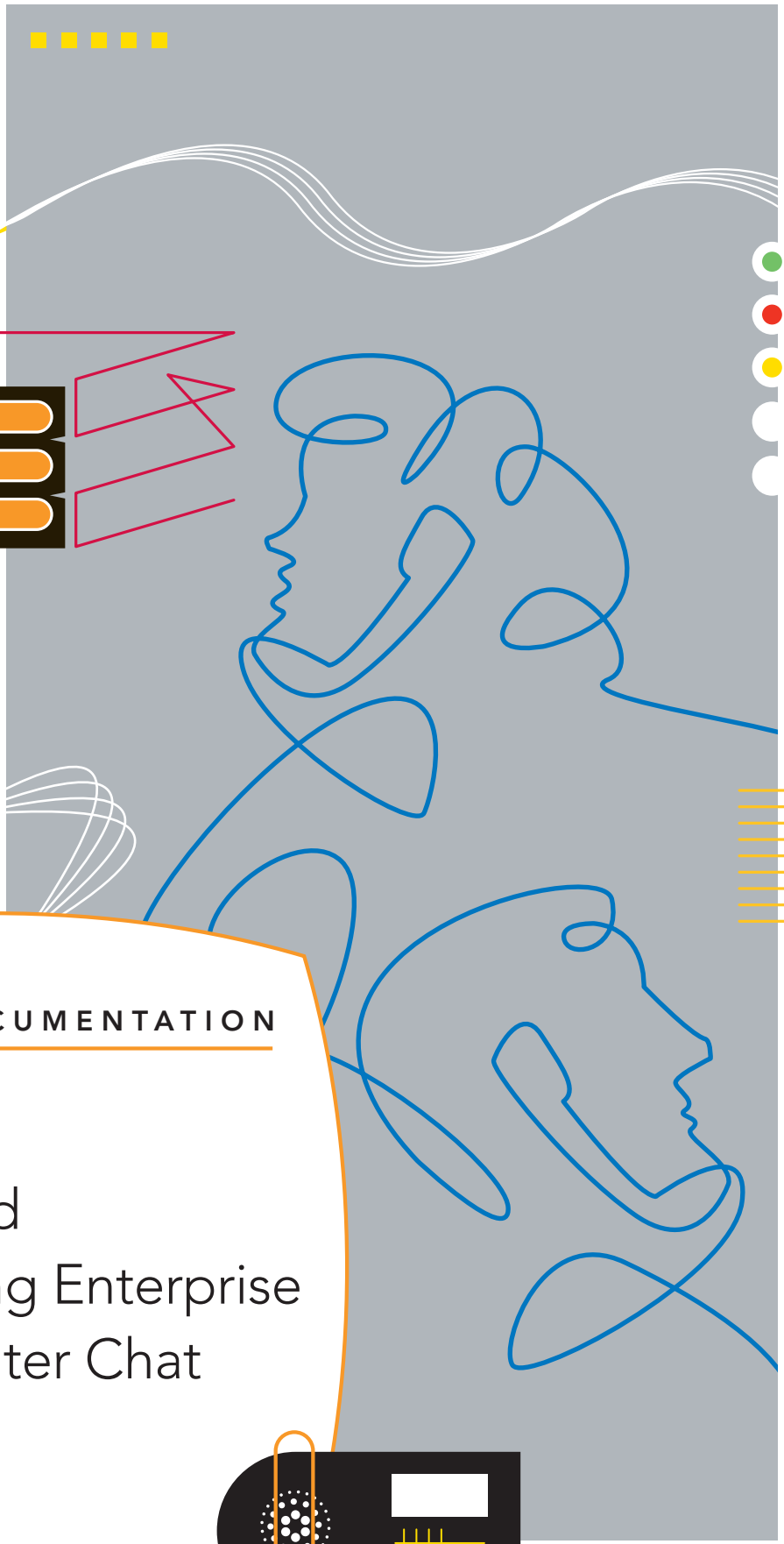
ShoreTel UNIVERSITY



PRODUCT DOCUMENTATION

# Installing and Implementing Enterprise Contact Center Chat

RELEASE 5.1

# Document and Software Copyrights



# Trademarks



# Patents



# Version Information

Using Chat Toolkit
Contact Center 5
Part Number: 800-1261-01
Version: Chat_51_21 September 2009

# Company Contact Information

ShoreTel, Inc.
960 Stewart Drive
Sunnyvale, California 94085

(408) 331-3300
(408) 331-3333 fax

www.shoretel.com

# Table of Contents

# Preface

## Objectives

This guide provides information on using ShoreTel Chat Toolkit for the ShoreTel Contact Center (CC) and Enterprise Contact Center (ECC) editions.

## Documentation Overview

The following additional ShoreTel documentation is available for Contact Center:

- Administrator Guide
- Context-Sensitive Online Help
- Supervisor User Guide
- Installation Guide
- CRM Integration Guide.
- Agent Toolbar User Guide
- Using Reports
- Release Notes

## Document Conventions

The following conventions are used in this guide:

- Data-entry fields, hypertext links, control buttons, keywords, and other items in the interface are in a **boldface** font.
- Information that you enter in data fields is in a `data_entry` font.

CHAPTER 1

# Overview

ShoreTel's Enterprise Contact Center Chat Tookit provides a chat browser for handling chat contacts. The Toolkit is a set of scripts, applets and HTML templates that work together to provide the chat, and web call-back capabilities.

Chat sessions are initiated through requests coming from a chat Web browser. The customer clicks a Chat button on their page to open a chat window while waiting for an agent to respond. A chat browser window opens at the agent end, allowing agents to start the chat session.

Contact Center supports the following chat services:

- Chat conversation - Customers browsing your Web site can initiate a chat session with a live agent.
- Web callback - Customers can leave telephony callback information. The system uses this information to initiate a telephone call with an available agent.

## Contents of the Chat Toolkit

The Toolkit consists of a set of files containing:

- JavaScript code that should be used as is—without any modification.
- Sample Forms that can be embedded in a customer web site.
- Mandatory HTML Pages templates: Some mandatory pages should reside on the application server but can be customized by the customer to provide the look and feel of the customers' site.
- A collection of servlets that should be deployed on a Servlet Engine.

## System Requirements

To use the web features (chat and call back), the following items are required:

- Agent stations should have Internet Explorer v 6.0 or higher installed as the default browser with JavaScript, Java, and cookies enabled for the web address of the chat server.
- Pop-up blocking on the agent stations should be disabled for the chat server's domain.

To use the web services, the Contact Center installation should include the servlets that are part of the Contact Center. These servlets should work on a Jakarta Tomcat server connected to the Internet.

The Contact Center server and the chat servlet communicate with each other over a TCP/IP connection. Therefore, this connection should be open on the organization firewall. (The required IP address and ports can be configured in the WEB.XML deployment descriptor of the Contact Center Chat application.)

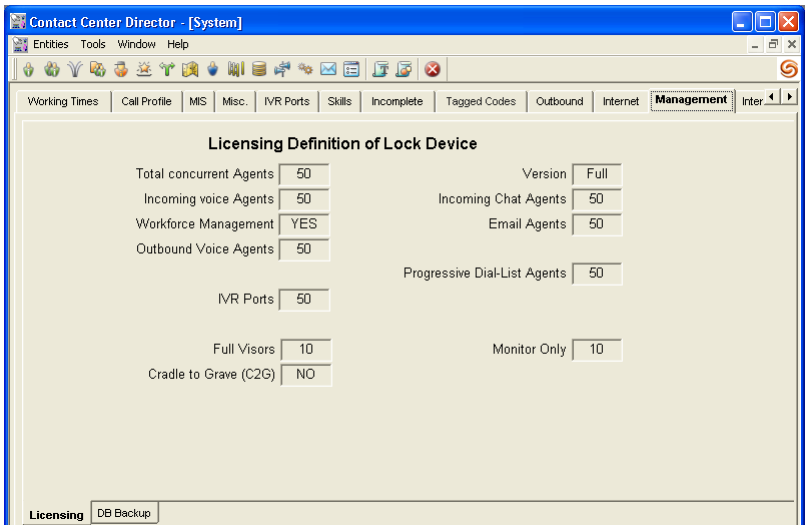
CHAPTER 2

# Installing and Upgrading Chat

Before installing Enterprise Contact Center Chat Toolkit, ensure that:

- You have valid chat licenses, for the expected number of inbound connections, on the Contact Center Server dongle.

  This information is available from Contact Center Director. Click the **System** icon, and then select the **Management** tab. In the **Licensing** page, the total number of chat licenses is displayed.




- A separate chat server with Tomcat 5.5 or higher, and Java 1.6 or higher, installed. Tomcat is used to run the chat Java servlets. The chat server receives connections on port 80.

  The chat server should be in a DMZ or other location available to the public internet.

  Tomcat can be downloaded from http://tomcat.apache.org/download-55.cgi. Java can be downloaded from http://www.java.com/en/download/manual.jsp.

- Network connectivity exists between the chat server and the Contact Center server via TCP/IP on port 31452.
- You have access to the company website to add forms to start chat or web-based callbacks.

There is no limitation on using the same web server on the same domain and the same ports for both components.

# Tips When Installing Tomcat

Although not required, it is recommended that the Tomcat directory (Apache Software Foundation) be installed on the chat server at the root level of your C or D drive. Installing the directory at the root level makes it easy to locate the directory for update when the Chat Toolkit is installed.

Make sure that the Java path to Tomcat is correct. The Tomcat installer prompts for the Java path during installation. This path is in the Program Files directory. The port value should be 8080.

# The ECC Chat Servlet

You need to install the ECC Chat servlet on the chat server, and configure it to run on the chat server and the Contact Center server.

## Installing the ECC Chat Servlet

To install the ECC Chat servlet on the chat server:

1. If not already running, start Tomcat.
2. Copy the **ECCChat.war** file from the Chat Toolkit folder (on the DVD) to the Tomcat webapps directory.

   The file is automatically decompressed by the Tomcat server and creates the ECC Chat folder in the webapps directory.

## Configuring the ECC Chat Servlet on the Chat Server

Once the ECC Chat servlet is installed on the chat server, it needs to be configured.

1. Open the **web.xml** file with a text editor. This file controls web application connectivity with Contact Center and displays the Java log window on the server for troubleshooting.

   The **web.xml** file can be found in root:\Apache Software Foundation\Tomcat\webapps\ECCChat\WEB-INF directory.
2. In the **web.xml** file, under the <servlet> section, change the parameter value **CCA14** to the valid host name or IP address of the Contact Center server. The host name must be a fully qualified domain name (FQDN).

   For example:

   <param-name>ECCServerName</param-name>
   <param-value>CCA14</param-value>

   would be changed to:

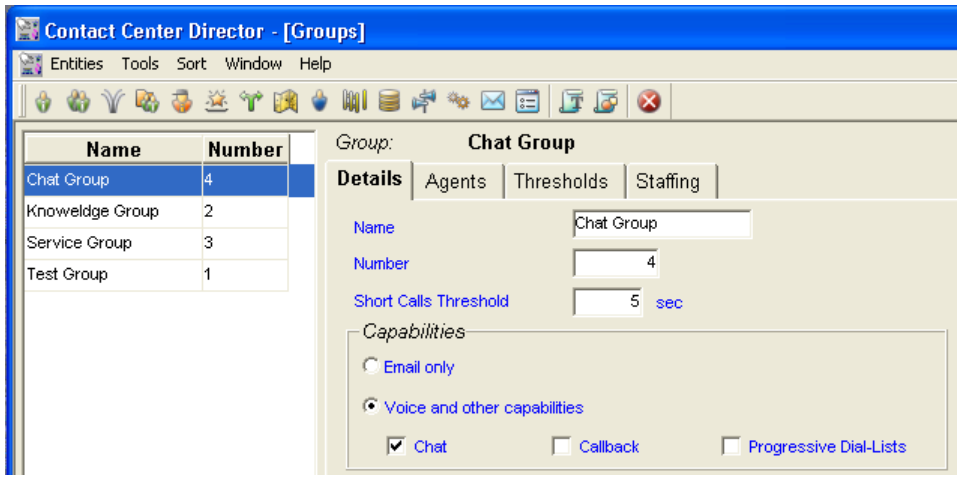
   <param-name>ECCServerName</param-name>
   <param-value>10.23.55.110</param-value>.
3. Open the **parameters.js** file with a text editor. This file can be found in root:\Apache Software Foundation\Tomcat\webapps\ECCChat.
4. In the **parameters.js** file, under the //Server Base Address section, change the parameter of the value **var baseAddr** to the valid host name or IP address of the Contact Center server. The host name must be a fully qualified domain name (FQDN).

   For example, the line var baseAddr = "http://mychatserver.com:8080"; would be changed to var baseAddr = "http://MSmith-HX380.shoretel.com:8080".

## Configuring the ECC Chat Servlet on the Contact Center Server

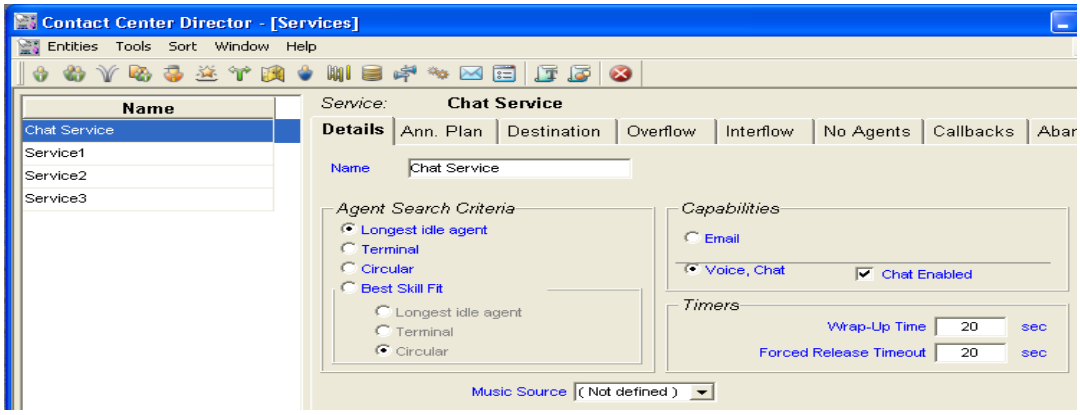
Now the ECC Chat servlet must be configured on the Contact Center server, using Contact Center Director.

1. Create or edit an ECC group to handle the incoming chat requests. This group must have the **Chat** option selected, in the Groups < Details tab, to accept chat calls.



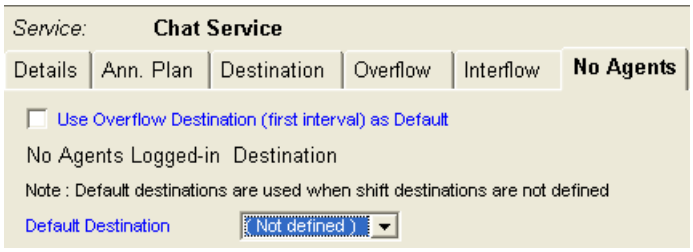

You may want to change the statistics to better represent the slower nature of chat (as opposed to voice calls) so that reporting is more valid.

2. Create or edit an ECC service to queue the incoming chat requests. This service must have the **Chat Enabled** option selected, in the Services < Details tab, in order to accept chat calls.




When creating an EEC service for chat, make sure that:

- The Music Source option is not defined, because a chat request can not go to the voice IVR.
- The overflow destination groups must be chat enabled.
- Chat request can only be interflowed to chat-enabled services and IRNs.
- The No Agent Default Destination is not defined.

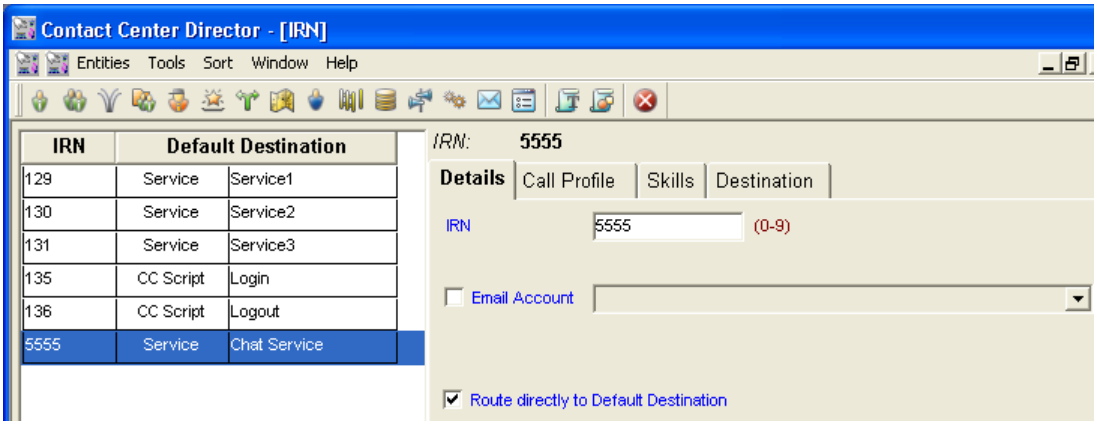



- An announcement plan can be used, but only use CCScripts with the Send Page, Send Chat, and Hang Up elements. Using any voice elements can have adverse effects and may cause instability within the system.

3. Create an IRN for routing of the chat requests. This is the default IRN number used for chat.

   When creating an IRN for chat, make sure that:

   - **Route Directly to Default Destination** is selected. This is because ANI Domain Routing and Customer Routing do not work with a chat request.




   - The call profile section can be used. Chat requests can have attached call profiles that can be passed along to the agent desktop.
   - Scheduling can be used, as long as the shift destination is valid for a chat request.

4. Assign to the chat ECC group to the chat ECC service.
5. For the Internet System entity, modify the Max Chat Calls field with the desired number of inbound chat calls the system will handle.

   **Note**: By default, the number of calls in Max Chat Calls field is 0.

# The Web Callback Servlet

You need to install and configure the Web Callback applet on the chat server.

## Installing the Web Callback Servlet

To install the Web Callback servlet on the chat server:

1. If not already running, start Tomcat.
2. Copy the **WebCallback.war** file from the Chat Toolkit folder (on the DVD) to the Tomcat webapps directory.

   The file is automatically decompressed by the Tomcat server and creates the WebCallback folder in the webapps directory.

## Configuring the Web Callback Servlet on the Chat Server

Once the Web Callback servlet is installed on the chat server, it needs to be configured.

1. Open the **web.xml** file with a text editor. This file controls web application connectivity with Contact Center and displays the Java log window on the server for troubleshooting.

   The **web.xml** file can be found in root:\Apache Software Foundation\Tomcat\webapps\WebCallback\WEB-INF directory.
2. In the **web.xml** file, under the <servlet> section, change the parameter value **CCA13** to the valid host name or IP address of the Contact Center server. The host name must be a fully qualified domain name (FQDN).

   For example:

   <param-name>ECCServerName</param-name>
   <param-value>CCA13</param-value>

   would be changed to:

   <param-name>ECCServerName</param-name>
   <param-value>10.23.55.110</param-value>.

# Verify Server Communication

Finally, verify that the chat server is communicating with the Contact Center server. The easiest way to do so is:

1. Launch the CC Console on the ECC server.
2. View the Chat LED. A green color indicates that a valid connection between the ECC server and the Tomcat server.

# Upgrading Enterprise Contact Center Chat

If you are upgrading from a previous version of ShoreTel Contact Center to version 5.1, you also need to upgrade Enterprise Contact Center Chat.

To upgrade to Enterprise Contact Center Chat 5.1:

**Step 1** Backup Chat Toolkit 5.0 by copying the **ECCChat** folder in the webapps directory on the chat server.

**Step 2** Copy the 5.1 **ECCChat.war** file, from the Chat Toolkit folder on the ShoreTel Contact Center 5.1 Solution Installation CD, to the Tomcat webapps directory on the chat server.

The file is automatically decompressed by the Tomcat server and creates the ECC Chat folder in the webapps directory.

**Step 3** Update the **web.xml** file (using a text editor) by changing the parameter value for the parameter name **ECCServerName** to the valid host name or IP address of the Contact Center server. The host name must be a fully qualified domain name (FQDN).

The web.xml file controls web application connectivity with Contact Center and displays the Java log window on the server for troubleshooting. It can be found in the root:\Apache Software Foundation\Tomcat\webapps\ECCChat\WEB-INF directory.

**Step 4** Copy the 5.0 version of the **chat.css** file from your backup location to the ECC Chat folder in the webapps directory.

This file defines the visual layout of the chat.

**Step 5** Copy the 5.0 version of the HTML templates (**chat.html** and any other custom html pages) from your backup location to the ECCChat folder in the webapps directory.

These files specify the look and feel of the chat browser.

**Step 6** Verify that the chat server is communicating with the Contact Center server by launching the CC Console on the ECC server.

If the Chat LED has a green color, a valid connection between the ECC server and the Tomcat server exists.

# Upgrading Enterprise Contact Center Web Callback

When upgrading from a previous version of ShoreTel Contact Center to version 5.1, you also need to upgrade Enterprise Contact Center Web Callback.

To upgrade to Enterprise Contact Center Web Callback 5.1:

**Step 1** Backup Chat Toolkit 5.0 by copying the **WebCallback** folder in the webapps directory on the chat server.

**Step 2** Copy the 5.1 **WebCallback.war** file, from the Chat Toolkit folder on the ShoreTel Contact Center 5.1 Solution Installation CD, to the Tomcat webapps directory on the chat server.

The file is automatically decompressed by the Tomcat server and creates the WebCallback folder in the webapps directory.

**Step 3** Update the **web.xml** file (using a text editor) by changing the parameter value for the parameter name **ECCServerName**, to the valid host name or IP address of the Contact Center server. The host name must be a fully qualified domain name (FQDN).

The web.xml file controls web application connectivity with Contact Center and displays the Java log window on the server for troubleshooting. It can be found in the root:\Apache Software Foundation\Tomcat\webapps\WebCallbck\WEB-INF directory.

**Step 4** Copy the 5.0 version of the **chat.css** file from your backup location to the WebCallback folder in the webapps directory.

This file defines the visual layout of the chat.

**Step 5** Copy the 5.0 version of the HTML templates (**webcallback.html** and any other custom html pages) from your backup location to the WebCallback folder in the webapps directory.

These files specify the look and feel of the chat browser.

**Step 6** Verify that the chat server is communicating with the Contact Center server by launching the CC Console on the ECC server.

If the Chat LED has a green color, a valid connection between the ECC server and the Tomcat server exists.

CHAPTER 3

# Implementing Chat Toolkit

To begin a chat conversation, the customer navigates into a specific page from which the chat process can begin. This page resides in the organization's web site.

In this page, a form exists that should be filled with the necessary information for the chat process to begin. The information can be filled any way the web developer considers for the specific requirements of the organization. After all the details are filled in, the customer submits the form.

The responsibility for filling in the form and validating the data resides with the web developer. The Chat Toolkit performs no tests on the data that it receives.

## Enabling a Site with Chat

The form that begins the chat should contain information that can be divided into the following categories:

- **Operational fields (mandatory)**: These fields are mandatory and configure the responses, services, and so on, that the Contact Center uses to route the call.
- **Service Required fields (mandatory)**: These fields define the service that is required in the Contact Center as defined in the Contact Center Admin.
- **Customer Information fields (Optional but with a default)**: Theses fields define the Customers email address and the name that will be used in the Chat window.
- **Call Profile fields (optional)**: These fields define values to the Call Profile fields defined in the Contact Center Admin. These values should be strings and are passed as is to the Contact Center.
- **Skill fields (optional)**: These fields define values to the skills defined in the Contact Center Admin. The values should be numeric and are passed as is to the Contact Center so it can route the call.

Whether any fields are hidden, filled in by the customer, or hard coded inside the form is irrelevant to the Contact Center. However, it is important for the field names and their values to match the field names and values as defined in Contact Center Director.

After the form is submitted, Contact Center starts the chat session.

# Web Developer Checklist

The following checklist defines the necessary steps that a web developer must follow in order to integrate a web site with the chat feature:

1. Create a page where the Chat process will begin. In the page embed the Chat Begin Form.

   If you are using the chat.html page, open the file using Notepad or any other text editor.

2. Change the input element **routing_device** to the IRN of the chat routing created in Contact Center Director.

3. Change the input element **service_req** to the service to handle this request. This must be the service pointed to by the IRN configured above.

4. Save the file.

## Detailed Explanation of the Chat Begin Form

```
<!—1 -->
 <!-- MANDATORY
      service_req: defines the service that the
      call will be routed to.
 -->

 <input type="hidden" name="service_req", value="food">

<!—2 -->
 <!-- MANDATORY: defines the name that will be used by the
      cusomer in the chat window
  -->

<!—3 -->
 <input type="text" name="Cust_name" size="30">
 <input type="hidden" name="Priority" value="0">

 <!-- Optional: Customers email address to send the trancript to
  -->

<!—4 -->
 <input type="text" name="Cust_email" size="50">

<!—5 -->
 <!-- Optional: Call profile fields follow -->

<!—6 -->
 <input type="hidden" name="LanguageText" value="ENGLISH">

<!—7 -->
 <!-- Optional: Skill values follow -->
 <input type="hidden" name="English" value="100">
 <input type="hidden" name="Spanish" value="0">
 <input type="hidden" name="Russian" value="0">
```

<!—1 --> Field “**service_req**": This field defines the Service that will be used in the Contact Center to handle the Chat Call. A Service with this value must be defined in Contact Center Director. The service should be configured as a Chat service.

<!—2 --> Customer Information fields start here

<!—3 --> Field "**Cust_name**"

<!—4 -->Field “**Cust_email**"

<!—5 --> Call profile Fields

<!—6 --> The call profile fields follow the following format:

- name=”name as in Contact Center Director for the Call Profile field”
- value=”value for the field”

  In this page we use the call profile fields “Priority” and “LanguageText” as an example.

<!—7 --> Skill fields: The skill fields follow the following format::

- name=”name as in Contact Center Director for the Skill”
- value=”value for the field”

### Visual Customization

The visual layout can be changed by modifying the **chat.css** file using standard CSS formating.

# Enabling Web Callback

Using the Web Callback feature, the customer can use the web to schedule the time the Contact Center will establish a callback call to the customer at the customer’s phone number.

In order to use this feature, the callback feature in Contact Center has to be enabled.

## Web Developer Checklist

The process to enable the Web Call Back feature in a site is as simple as submitting a form to the WebCallBackServlet located on the Tomcat server.

The parameters submitted by the form should contain all the necessary information, so that Contact Center can return the call.

The Toolkit provides a sample page WebCallBackRequest.htm where all the relevant information is filled.

## The webcallback.html

The following is a sample of a web callback page that allows the customer to enter a callback number and time.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Web Callback</title>
        <link rel="stylesheet" type="text/css" href="chat.css">
        <script language="javascript" type="text/javascript" src="parameters.js"></script>


</head>
<body bgcolor="#FFFFFF" >



        <FORM METHOD="POST" ACTION="/WebCallback/WebCallbackServlet" >

                <div>
                                First Name <INPUT name="CallBack_Name" >
                                Surname <INPUT name="CallBack_Surname">

                </div>
                <div>

                                Phone Number:(
                                <INPUT name="CallBack_AreaCode" size="3">)
                                <INPUT name="CallBack_Phone" size="7"> (Area-Code) Number
                </div>
                <div>
                                Call back Time
                                <select name="CallBack_Hour" size="1" >
                                        <option value="1">01
                                        <option value="2">02
                                        <option value="3">03
                                        <option value="4">04
                                        <option value="5">05
                                        <option value="6">06
                                        <option value="7">07
                                        <option value="8">08
                                        <option value="9">09
                                        <option value="10">10
                                        <option value="11">11
                                        <option value="12">12

                                </select>
                                <select name="CallBack_Min" size="1" >
                                        <option value="0">00
                                        <option value="5">05
                                        <option value="10">10
                                        <option value="15">15
                                        <option value="20">20
                                        <option value="25">25
                                        <option value="30">30
                                        <option value="35">35
                                        <option value="40">40
                                        <option value="45">45
                                        <option value="50">50
                                        <option value="55">55

                                </select>
```

```
                                <select name="CallBack_AM_PM" size="1" >
                                        <option value="AM">AM
                                        <option value="PM">PM
                                </select>

                        </div>
                        <INPUT type="HIDDEN" name="service_req" , value="WebCallback">
                        <INPUT type="HIDDEN" name="CallBack_ResponsePage"
                                value="webcallback_response.html">

                        <div>
                        <br>
                                Please press "Submit" button after completing the form.
                        <br>
                                Press "Reset" to clear and start again.
                        </div>
                        <div>

                                <INPUT name=Submit type="submit" value=Submit>
                                <INPUT type="reset">
                        </div>
                </FORM>

</BODY>
</HTML>
```

The following table explains the parameters used by the Web Callback form.

| Field Name | Description | Legal Values |
|---|---|---|
| FORM method=POST | Method should be post | POST |
| action="http://tomcat.cs.com:8080 /EpicChatService /WebCallbackServlet | This should point to the WebCallBackServlet on the Tomcat server | |
| CallBack_Name | Name of the person to call | String |
| CallBack_Surname | Surename of the person to call | String |
| CallBack_AreaCode | Area code of the destination phone | String with numbers |
| CallBack_Phone | Phone number to dial | String with numbers |
| CallBack_Hour | Time of the return call (Hours) | 1-12 |
| CallBack_Min | Time of the return call (Minutes) | 00-59 |
| CallBack_AM_PM | Call back time is AM or PM? | AM \| PM |
| CallBack_ResponsePage | Page that will be sent back to the customer after submitting the call back page | |

# Servlet Configuration

The deployment descriptor presented here is for the Tomcat Server. If you customize the descriptor to modify the names and locations of the servlets, all the pages should be modified accordingly; therefore you should keep the default mappings if possible.

## WAR Deployment Descriptor

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app
   PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
   "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
   <servlet>
               <servlet-name>ECCChat</servlet-name>
               <display-name>ECCChat</display-name>
               <servlet-class>eccChatService.ECCChat</servlet-class>
               <init-param>
                        <param-name>ECCVersion</param-name>
                        <param-value>5.00.00</param-value>
               </init-param>
               <init-param>
                        <param-name>FC_CPBI</param-name>
                        <param-value>760</param-value>
               </init-param>
               <init-param>
                        <param-name>CPBI_SERVICE</param-name>
                        <param-value>2</param-value>
               </init-param>
               <init-param>
                        <param-name>ECCServerName</param-name>
                        <param-value>10.28.0.103</param-value>
               </init-param>
               <init-param>
                        <param-name>ECCServerPort</param-name>
                        <param-value>31452</param-value>
               </init-param>
               <init-param>
                        <param-name>ClientTimeoutInterval</param-name>
                        <param-value>2</param-value>
               </init-param>
               <load-on-startup>2</load-on-startup>
   </servlet>

<servlet>
      <servlet-name>WebCallbackServlet</servlet-name>
          <servlet-class>WebCallback.WebCallbackServlet</servlet-class>
          <init-param>
          <param-name>FC_CPBI</param-name>
                  <param-value>760</param-value> </init-param>
      <init-param>
                  <param-name>CPBI_SERVICE</param-name>
          <param-value>2</param-value>
```

```
            </init-param>
         <init-param>
            <param-name>EpicServerName</param-name>
                    <param-value>10.28.0.103</param-value>
        </init-param>
        <init-param>
                    <param-name>EpicServerPort</param-name>
                    <param-value>31452</param-value>
        </init-param>
        <init-param>
                    <param-name>ConsoleDebugStatus</param-name>
                    <param-value>False</param-value>
        </init-param>
              <load-on-startup>2</load-on-startup>
</servlet>

   <servlet>
            <servlet-name>ECCAgentStartPage</servlet-name>
            <display-name>ECCAgentStartPage</display-name>
            <servlet-class>eccChatService.ECCAgentStartPage</servlet-class>
   </servlet>
   <servlet-mapping>
            <servlet-name>ECCChat</servlet-name>
            <url-pattern>/ECCChat</url-pattern>
   </servlet-mapping>

            <servlet-mapping> <servlet-name>WebCallbackServlet</servlet-name>
            <url-pattern>/WebCallbackServlet</url-pattern> </servlet-mapping>

</web-app>
```

# Index