ORACLE
PEOPLESOFT ENTERPRISE

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft MultiChannel Framework

March 2007

ORACLE

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft MultiChannel Framework
SKU PT849MCF-B 0307
Copyright © 1988-2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Open Source Disclosure

Oracle takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation. The following open source software may be used in Oracle's PeopleSoft products and the following disclaimers are provided.

Apache Software Foundation

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 2000-2003. The Apache Software Foundation. All rights reserved. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

OpenSSL

Copyright © 1998-2005 The OpenSSL Project. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Loki Library

Copyright © 2001 by Andrei Alexandrescu. This code accompanies the book: Alexandrescu, Andrei. "Modern C++ Design: Generic Programming and Design Patterns Applied". Copyright © 2001 Addison-Wesley. Permission to use, copy, modify, distribute and sell this software for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

Helma Project

Copyright © 1999-2004 Helma Project. All rights reserved. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HELMA PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Helma includes third party software released under different specific license terms. See the licenses directory in the Helma distribution for a list of these license.

Sarissa

Copyright © 2004 Manos Batsis.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

ICU

ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Sun's JAXB Implementation – JDSDK 1.5 relaxngDatatype.jar 1.0 License

Copyright © 2001, Thai Open Source Software Center Ltd, Sun Microsystems. All rights reserved.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

W3C IPR SOFTWARE NOTICE

Copyright © 2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

Note: The original version of the W3C Software Copyright Notice and License could be found at http://www.w3.org/Consortium/Legal/copyright-software-19980720.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

Contents

General Preface

About This PeopleBook xxxvii

PeopleSoft Enterprise Application Prerequisites xxxvii

Application Fundamentals xxxvii

Documentation Updates and Printed Documentation xxxviii

Obtaining Documentation Updates xxxviii

Downloading and Ordering Printed Documentation xxxviii

Additional Resources xxxix

Typographical Conventions and Visual Cues xl

Typographical Conventions xl

Visual Cues xli

Country, Region, and Industry Identifiers xli

Currency Codes xlii

Comments and Suggestions xlii

Common Elements Used in PeopleBooks xlii

Preface

PeopleSoft MultiChannel Framework Preface xlv

PeopleSoft MultiChannel Framework xlv

Additional Resources xlv

Chapter 1

Getting Started with PeopleSoft MultiChannel Framework 1

PeopleSoft MultiChannel Framework Overview 1

PeopleSoft MultiChannel Framework Implementation 1

Chapter 2

Understanding PeopleSoft MultiChannel Framework 5

PeopleSoft MultiChannel Framework 5

PeopleSoft MultiChannel Framework Elements and Channels 5

PeopleSoft MultiChannel Elements 6

PeopleSoft MultiChannel Framework Channels 6

Copyright © 1988-2007, Oracle. All rights reserved.

PeopleSoft MultiChannel Framework Universal Queue.....7
PeopleSoft MultiChannel Console.....8
PeopleSoft MultiChannel Framework Architecture.....8
PeopleSoft MultiChannel Framework Server Architecture.....8
Chat Architecture.....8
Email Architecture.....10

Chapter 3
Configuring PeopleSoft CTI.....13
Understanding PeopleSoft CTI.....13
PeopleSoft CTI.....13
PeopleSoft CTI Components.....14
Understanding PeopleSoft CTI Components.....15
PeopleSoft CTI Requirements.....16
PeopleSoft MultiChannel API.....16
Required Security for PSMCAPI.....17
JavaScript MultiChannel API.....18
Genesys Components.....18
Cisco Components.....19
Java Runtime Environment.....19
Configuring PeopleSoft CTI Using CTI Server or an Integrated Third-Party Routing System.....19
Pages Used to Configure CTI Using CTI Server or an Integrated Third-Party Routing System.....20
Installing PeopleSoft CTI.....21
Enabling PeopleSoft CTI.....22
Creating CTI Configurations.....22
Configuring the CTI Non-Applet Console.....23
Configuring the CTI Applet Console.....24
Using the Shared Phone Book Page.....25
Using the CTI Genesys Page.....26
Using the CTI Cisco Page.....27
Using the Miscellaneous Page.....30
Using the Reason Code Page.....31
Configuring CTI Queues and CTI Agents.....32
Pages Used to Configure CTI Queues and CTI Agents.....32
Configuring CTI Queues.....32
Configuring CTI Agents.....33
Using the Phone Book Page.....34
Personalizing Agent Console.....35
Viewing Information on the Agent Information Page.....37

Copyright © 1988-2007, Oracle. All rights reserved.

Using Other CTI Options.....37
Configuring Pop-Up Windows.....38
Supporting Single Signon.....40
CTI Event Logging.....40
Implementing Free Seating.....40
Using the CTI Sample Pages.....41
Page Used to Demonstrate Outbound Calls.....41
Using the Outbound Call Page.....41

Chapter 4
Using PeopleSoft CTI.....43
Understanding PeopleSoft CTI.....43
PeopleSoft CTI.....43
Cisco Switch Considerations.....44
Using PeopleSoft CTI.....44
Getting Started.....45
Using the CTI Console.....46
Selecting Call Actions.....47
Answering a Call.....48
Transferring a Caller.....48
Initiating Conference Calls.....50
Working with the Hold Status.....50
Disconnecting a Caller.....51
Switching Agent Ready Status.....51
Dialing an Outbound Call.....52
Completing a Call.....52
Using Hot Keys.....53

Chapter 5
Configuring REN Servers.....55
Understanding REN Servers.....55
REN Server Failover, Scalability, and Security Configuration.....55
REN Server Failover.....56
REN Server Clusters.....57
Understanding SSL-Enabled REN Servers.....57
Installing Digital Certificates.....58
Authenticating Server and Client.....58
Performance and Scalability for SSL-Enabled REN Servers.....58

Copyright © 1988-2007, Oracle. All rights reserved.

Configuring REN Server Security....58
Understanding REN Server Security Configuration....58
Defining Permission Lists for REN Server Access....59
Configuring REN Servers....60
Understanding REN Server Configuration Options....60
Page Used to Configure REN Servers....63
Configuring REN Servers and SSL-Enabled REN Servers....63
Defining REN Servers....66
Configuring REN Server and SSL-Enabled REN Server Clusters....68
Pages Used to Configure REN Server and SSL-Enabled REN Server Clusters....69
Clustering REN Servers and SSL-Enabled REN Servers....69
Specifying REN Server Ownership....71
Specifying REN Server Cluster Members....72
Configuring a Reverse Proxy Server with a REN Server....73
Understanding RPS Configuration....73
Example: Configuring a WebLogic 8.1 RPS for a REN Server on Another Host Machine....73
Example: Configuring Oracle Application Server (OAS) to Proxy for a REN Server....75

Chapter 6
Configuring PeopleSoft MCF Servers and Clusters....77
Understanding PeopleSoft MCF Server and Cluster Architecture....77
PeopleSoft MCF Server Configuration....77
PeopleSoft MCF Cluster Architecture....78
Queue Server and Queue Server Failover....79
Logical Queues and Physical Queues....80
PeopleSoft MCF Log Server and Log Server Failover....81
Queue Server Scalability....81
Recommended Configurations....81
Configuring PeopleSoft MCF Clusters....82
Understanding PeopleSoft MCF Cluster Configuration....82
Page Used to Configure PeopleSoft MCF Clusters....83
Configuring PeopleSoft MCF Clusters....83

Chapter 7
Configuring PeopleSoft MCF Queues and Tasks....85
Defining Queues....85
Pages Used to Define Queues....86
Defining Queues....86

Copyright © 1988-2007, Oracle. All rights reserved.

Defining Chat Responses....87
Defining Static Push URLs....88
Configuring Tasks....89
Page Used to Configure Tasks....89
Configuring Tasks....89
Viewing the Cluster Summary....93
Page Used to View the Cluster Summary....93
Viewing the Cluster Summary....93
Tuning Cluster Parameters....93
Page Used to Tune Clusters....94
Tuning Cluster Parameters....94
Notifying Clusters of Changed Parameters....101
Page Used to Notify Clusters of Changed Parameters....102
Notifying Clusters of Changed Parameters....102

Chapter 8
Configuring PeopleSoft MCF Agents....103
Defining Agents....103
Pages Used to Define Agents....103
Creating Agents....103
Specifying Languages That an Agent Supports....105
Personalizing an Agent's Presence....105
Defining Optional Agent Characteristics....106
Pages Used to Define Optional Agent Characteristics....107
Setting Up Buddy Lists....107
Configuring Windows....108
Personalizing Chat....109
Specifying Agent-Specific URLs....110
Specifying Miscellaneous Parameters....111

Chapter 9
Administering Queues, Logs, and Tasks....113
Administering Physical Queues....113
Pages Used to Administer Physical Queues....113
Moving Agents Between Physical Queues....114
Moving Queues....114
Balancing Queues....116
Viewing Queue Server, Queue, and Agent States....116

Copyright © 1988-2007, Oracle. All rights reserved.

Pages Used to View Queue Server, Queue, and Agent States....117
Viewing the Queue Server State....117
Viewing the Queue State Summary....117
Viewing the Agent State Summary....118
Viewing Broadcast, Chat, and Event Logs....119
Pages Used to View Broadcast, Chat, and Event Logs....120
Viewing Broadcast Logs....120
Viewing Chat Logs....121
Viewing Event Logs....123
Viewing PeopleSoft MCF Logs....126
Administering Overflow and Escalated Tasks....127
Pages Used to Administer Overflow and Escalated Tasks....127
Administering Overflow Tasks....127
Administering Escalated Tasks....128

Chapter 10
Managing Tasks and Using Chat in PeopleSoft MultiChannel Framework....129
Managing Tasks with the MultiChannel Console....129
Using the MultiChannel Console to Work with Tasks....129
Communicating with Customers and Agents Using Chat....131
Using the Agent Chat Window....131
Using the Customer Chat Window....133

Chapter 11
Using PeopleSoft MCF Broadcast and Working with Sample Pages....137
Using PeopleSoft MCF Broadcast....137
Pages Used to Configure PeopleSoft MCF Broadcast....138
Understanding JSMCAPI Broadcast....138
Implementing MCF Broadcast....139
Using JSMCAPI Broadcast with MCF Supervisor Console....139
Using PeopleCode Broadcast....144
Viewing Broadcast Logs....147
Working with Sample Pages....147
Pages Used to Work with Sample Pages....148
Using the Customer Chat Sample Page....148
Using the URL Wizard....149
Using the Generic Event Sample Page....151
Using the Generic Event Window....152

Copyright © 1988-2007, Oracle. All rights reserved.

Using the Email Sample Page.....153
Using the Email Window.....154
Using and Demonstrating JSMCAPI.....155
Understanding JSMCAPI.....155
Common Elements Used in This Section.....159
Pages Used to Use and Demonstrate JSMCAPI.....160
Using the CTI Sample Console.....160
Using the Agent Console Page.....167
Using the Monitor Agents Page and Sample Monitor - Agent States Page.....170
Using the Monitor Queues Page and Sample Monitor - Queue Statistics Page.....172
Using PeopleCode Built-in Functions with PeopleSoft MultiChannel Framework.....175
Using Universal Queue Classes.....175

Chapter 12
Configuring PeopleSoft MCF for Third-Party Routing Systems.....177
Understanding Third-Party Routing Systems.....177
Defining Third-Party Routing System Requirements.....180
Defining Third-Party Routing Rules.....180
Configuring PeopleSoft MCF for a Third Party.....180
Defining the Third-Party Flag.....181
Page Used to Define the Third-Party Flag.....182
Defining the Third-Party Flag.....182
Defining the PeopleSoft MCF Cluster Page for a Third Party.....182
Page Used to Define the PeopleSoft MCF Cluster Page for a Third Party.....182
Defining the PeopleSoft MCF Cluster Page for a Third Party.....183
Tuning PeopleSoft MCF Cluster Parameters for a Third Party.....184
Page Used to Tune PeopleSoft MCF Cluster Parameters for a Third Party.....184
Tuning PeopleSoft MCF Cluster Parameters for a Third Party.....184
Notifying PeopleSoft MCF Clusters of Changed Parameters for a Third Party.....187
Page Used to Notify Third-Party Clusters of Changed Parameters.....188
Notifying Third-Party Clusters of Changed Parameters.....188
Defining PeopleSoft MCF Queues for a Third Party.....188
Pages Used to Define PeopleSoft MCF Queues for a Third Party.....189
Defining PeopleSoft MCF Queues For Third Party.....189
Defining Canned Queue Messages.....190
Defining Canned Queue URL.....190
Defining PeopleSoft MCF Agents for a Third Party.....191
Pages Used to Define PeopleSoft MCF Agents for a Third Party.....192
Creating PeopleSoft MCF Agents for a Third Party.....192

Copyright © 1988-2007, Oracle. All rights reserved.

Setting Up Buddy Lists....193
Customizing Windows....194
Defining Messages....195
Specifying Agent-Specific URLs....197
Defining Agent's Presence....197
Specifying the Media....198
Specifying Languages That an Agent Supports....199
Specifying Miscellaneous Parameters....200
Configuring PeopleSoft MCF Task for a Third Party....200
Page Used to Configure PeopleSoft MCF Task for a Third Party....200
Configuring PeopleSoft MCF Task for a Third Party....200
Configuring CTI for a Third Party....201
Viewing Event Logs for a Third Party....201
Page Used to View Event Logs for a Third Party....201
Viewing Event Logs for a Third Party....201
Viewing Broadcast Logs for a Third Party....203
Page Used to View Broadcast Logs For Third Party....203
Viewing Broadcast Logs for a Third Party....203
Working with Third-Party Sample Pages....203
Pages Used to Work with Third-Party Sample Pages....204
Using the Customer Chat Sample Page....204
Using the Generic Event Sample Page....204
Using the Email Sample Page....204
Using the Sample Console Page....204
Using the MCF Broadcast Page....211
Using the PCodeBroadcast Page....212

Chapter 13
Understanding JSMCAPI Classes....213
Understanding JSMCAPI....213
Understanding JSMCAPI Classes....213
PSMC....217
Server....220
RENServer....220
Session....220
_Address....220
Line....221
Connection....221
Group....221

Copyright © 1988-2007, Oracle. All rights reserved.

Task....221
_User....222
MediaType....222
Reason....222
Statistics....222
Data....223
Globals....223
MCEvent....223
Caps....224
ForwardMode....224
_Address Class Hierarchy....224
_Address Class Constructor....225
_Address....226
_Address Class Fields....226
caps....226
id....226
_Address Class Callback Event Method....226
onError....226
_UQAddress Class Constructor....227
_UQAddress....227
_UQAddress Class Fields....227
Tasks....227
_UQAddress Methods....227
acceptTask....228
dequeueTask....228
_UQAddress Class Callback Event Methods....228
onAccepted....229
onAcceptingTask....229
onDequeueingTask....229
onNotify....229
onTaskAdded....229
onTaskRemoved....229
onUnassigned....230
_User Class Hierarchy....230
_User Class Constructor....230
_User....230
_User Class Fields....231
agentID....231
caps....231
id....231

Copyright © 1988-2007, Oracle. All rights reserved.

name....231
presences....232
states....232
ST_LOGGEDIN....232
ST_LOGGEDOUT....232
ST_NOTREADY....232
ST_READY....232
ST_UNKNOWN....232
ST_WORKNOTREADY....232
ST_WORKREADY....233
statistics....233
statistics1....233
statistics2....233
A2AChat Class Constructor....233
A2AChat....233
A2AChat Class Fields....234
address....234
agentID....234
agentName....234
appData....235
chatType....235
customerName....235
id....235
isConference....236
jr....236
question....236
subject....236
type....236
uniqueueId....236
A2AChat Class Method....236
getURL....237
A2AChatAddress Class Constructor....237
A2AChatAddress....237
A2AChatAddress Class Fields....237
id....238
tasks....238
A2AChatAddress Class Method....238
initiateChat....238
A2AChatAddress Class Callback Event Methods....239
onChatEnded....239

Copyright © 1988-2007, Oracle. All rights reserved.

onInitiatingChat....239
onNotify....239
AgentStatistics Class Constructor....239
AgentStatistics....239
AgentStatistics Class Fields....240
averageCallDuration....240
averageHoldDuration....240
callsHandled....240
data....240
percentIdleTime....240
percentTimeAvailable....241
percentTimeInCurrentState....241
percentTimeUnavailable....241
timeCurrentLogin....241
timeWorking....241
totalTaskAcceptedLogin....241
totalTaskDoneLogin....241
totalTaskUnassignedLogin....242
unavailableDuration....242
waitDuration....242
AppData Class Constructor....242
AppData....242
AppData Class Fields....242
data....243
groupId....243
jr....243
question....243
strData....243
subject....243
uniqueId....243
url....244
userId....244
username....244
wizUrl....244
AppData Class Method....244
addKeyValue....244
Buddy Class Constructor....245
Buddy....245
Buddy Class Fields....245
Buddy Class Callback Event Methods....246

Copyright © 1988-2007, Oracle. All rights reserved.

onStat1....246
onStat2....246
onState....246
Call Class Constructor....246
Call....247
Call Class Fields....247
line....247
statistics....248
CallData Class Constructor....248
CallData....248
CallData Class Fields....248
ani....248
callId....249
callType....249
data....249
dnis....249
CallData Class Method....249
addKeyValue....249
CallStatistics Class Constructor....250
CallStatistics....250
CallStatistics Class Fields....250
data....250
holdTime....251
queueTime....251
talkTime....251
Chat Class Constructor....251
Chat....251
Chat Class Fields....252
address....252
agentId....252
appData....252
chatconnection....253
chatType....253
customerName....253
groupId....253
question....253
subject....253
statistics....254
userData....254
Chat Class Method....254

Copyright © 1988-2007, Oracle. All rights reserved.

gettpUrl....254
getUrl....254
ChatAddress Class Constructor....255
ChatAddress....255
ChatAddress Class Fields....255
chatconnections....256
ChatAddress Class Methods....256
chat....256
getChatconnectionByConnectionId....256
getChatconnectionindexByConnectionId....257
getFreeChatconnection....257
getFreeChatconnectionIndex....257
ChatAddress Callback Event Methods....258
onCapabilitiesChanged....258
ChatConnection Class Constructor....258
ChatConnection....258
ChatConnection Class Fields....259
caps....259
chat....259
connectionId....259
id....259
state....259
ChatConnection Class Methods....260
answer....260
conference....260
forward....261
gethistory....261
getUrl....262
message....262
pushURL....263
reject....263
release....263
typing....264
ChatConnection Class Callback Event Methods....264
onAccepted....264
onAnswering....264
onCapabilitiesChanged....265
onChatdataChanged....265
onConferencing....265
onDialing....265

Copyright © 1988-2007, Oracle. All rights reserved.

onDropped....265
onError....266
onForwarded....266
onForwardError....266
onForwarding....266
onHistory....266
onIncomingChat....266
onMessage....267
onPartyAdded....267
onPartyChanged....267
onPartyRemoved....267
onProperties....267
onPushURL....268
onRejected....268
onReleased....268
onReleasing....268
onRevoked....268
onTalking....268
onTyping....269
onUserdataChanged....269
ChatConnectionCaps Class Constructor....269
ChatConnectionCaps....269
ChatConnectionCaps Class Fields....270
canAnswer....270
canConference....270
canConferenceSingle....270
canForward....270
canIndicateTyping....270
canPushURL....270
canReject....271
canSendMessage....271
ChatData Class Constructor....271
ChatData....271
ChatData Class Fields....271
data....271
ChatData Class Methods....271
addKeyValue....272
Email Class Constructor....272
Email....272
Email Class Fields....272

Copyright © 1988-2007, Oracle. All rights reserved.

address....273
agentId....273
appData....273
customerName....273
emailconnection....274
emailId....274
groupId....274
question....274
statistics....274
subject....274
userData....274
Email Class Method....275
gettpUrl....275
getUrl....275
EmailAddress Class Constructor....275
EmailAddress....276
EmailAddress Class Fields....276
agent....276
emailconnections....276
EmailAddress Class Methods....277
getEmailconnectionByConnectionId....277
getEmailconnectionindexByConnectionId....277
getFreeEmailconnection....277
EmailAddress Callback Event Methods....278
EmailConnection Class Constructor....278
EmailConnection....278
EmailConnection Class Fields....278
caps....279
connectionId....279
email....279
id....279
state....279
EmailConnection Class Methods....280
abandon....280
answer....280
complete....280
forward....281
reject....281
withdraw_RES....282
EmailConnection Class Callback Event Methods....282

Copyright © 1988-2007, Oracle. All rights reserved.

onAnswering....282
onCapabilitiesChanged....282
onCompleted....283
onDropped....283
onEmaildataChanged....283
onError....283
onForwarded....283
onForwardError....283
onForwarding....284
onIncoming....284
onProcessing....284
onRejected....284
onRevoked....284
onUserdataChanged....285
onWithdraw_REQ....285
EmailConnectionCaps Class Constructor....285
EmailConnectionCaps....285
EmailConnectionCaps Class Fields....285
canAnswer....286
canComplete....286
canForward....286
canReject....286
EmailData Class Constructor....286
EmailData....286
EmailData Class Fields....287
data....287
EmailData Class Methods....287
addKeyValue....287
Extension Class Constructor....287
Extension....287
Extension Class Fields....288
forwardMode....288
isDnd....288
lines....288
numOfLines....289
Extension Class Methods....289
cancelDnd....289
cancelForwardSet....289
forwardSet....290
getDialingLine....290

Copyright © 1988-2007, Oracle. All rights reserved.

getFreeLine....290
getLineByConnectionId....291
getOffHookLine....291
setDnd....292
Extension Class Callback Event Methods....292
onCancelingDnd....292
onCancelingForward....292
onDnd....292
onDndCanceled....293
onForwardCanceled....293
onForwarded....293
onForwarding....293
onSettingDnd....293
ExtensionCaps Class Constructor....294
ExtensionCaps....294
ExtensionCaps Class Fields....294
canCancelDnd....294
canDial....294
canFwdBusy....295
canFwdBusyNoAnswer....295
canFwdCancelForward....295
canFwdDefault....295
canFwdNoAnswer....295
canFwdUnconditional....295
canRefreshState....295
canSetDnd....296
ForwardMode Class Constructor....296
ForwardMode....296
ForwardMode Class Field....296
mode....296
GenericAddress Class Constructor....297
GenericAddress....297
GenericAddress Class Fields....297
agent....298
genericconnections....298
GenericAddress Class Methods....298
getFreeGenericconnection....298
getGenericconnectionByConnectionId....298
getGenericconnectionindexByConnectionId....299
GenericAddress Class Callback Event Methods....299

Copyright © 1988-2007, Oracle. All rights reserved.

GenericConnection Class Constructor....300
GenericConnection....300
GenericConnection Class Fields....300
caps....300
connectionId....300
generic....300
id....301
state....301
GenericConnection Class Methods....301
abandon....301
answer....302
complete....302
forward....302
reject....303
withdraw_RES....303
GenericConnection Class Callback Event Methods....304
onCapabilitiesChanged....304
onCompleted....304
onDropped....304
onError....304
onForwarded....305
onForwardError....305
onForwarding....305
onGenericdataChanged....305
onIncoming....305
onProcessing....305
onRejected....306
onRevoked....306
onUserdataChanged....306
onWithdraw_REQ....306
GenericConnectionCaps Class Constructor....306
GenericConnectionCaps....306
GenericConnectionCaps Class Fields....307
canAnswer....307
canComplete....307
canForward....307
canReject....307
GenericData Class Constructor....308
GenericData....308
GenericData Class Fields....308

Copyright © 1988-2007, Oracle. All rights reserved.

data....308
GenericData Class Methods....308
addKeyValue....308
GenericTask Class Constructor....309
GenericTask....309
GenericTask Class Fields....309
address....310
agentId....310
appData....310
customerName....310
genericconnection....310
genericId....311
groupId....311
question....311
statistics....311
subject....311
userdata....311
GenericTask Class Method....311
gettpUrl....312
getUrl....312
GLOBALS Class Fields....312
A2AChat.PS_JR....312
A2AChat.TYPE_ANSWER....313
A2AChat.TYPE_CONSULT....313
Server.TYPE_CTI....313
Server.TYPE_UQ....313
Task.TYPE_A2ACHAT....313
Task.TYPE_CHAT....314
Task.TYPE_CTI....314
Task.TYPE_EMAIL....314
Task.TYPE_GENERIC....314
GLOBALS Class Methods....314
initJSMCAPI....315
isValid....315
MCFBroadcast....315
Group Class Constructor....316
Group....316
Group Class Fields....317
id....317
name....317

Copyright © 1988-2007, Oracle. All rights reserved.

registered....317
statistics....317
statistics1....317
statistics2....318
Group Class Callback Event Methods....318
onStat....318
onStat1....318
onStat2....318
onTaskAdded....318
onTaskRemoved....319
GroupStatistics Constructor....319
GroupStatistics....319
GroupStatistics Fields....319
data....319
listOfTasksInTheQueueByTaskType....320
maxTaskCompletionTime....320
newestTask....320
newestTaskCompletionTime....320
numberOfAbandoned....320
numberOfLoggedIn....320
numberOfQueued....320
numUnassignedTasks....320
queuedWaitTime....321
queueUpTime....321
relativeQueueLoad....321
timeElapsedOldestTask....321
GroupStatistics1 Class Constructor....321
GroupStatistics1....321
GroupStatistics1 Class Fields....322
mostRecentTaskDone....322
mostRecentTaskDoneData....322
mostRecentTaskEnqueued....322
mostRecentTaskEnqueuedData....322
numAgentsAvailable....322
numAgentsLoggedIn....322
numEscalation....323
numOverflow....323
numTaskAccepted....323
numTaskDone....323
numTaskQueued....323

Copyright © 1988-2007, Oracle. All rights reserved.

reasonFlag....323
taskTotalTimeInSystem....323
timeSinceStart....324
GroupStatistics2 Class Constructor....324
GroupStatistics2....324
GroupStatistics2 Class Fields....324
averageTaskDuration....324
averageWaitTime....324
oldestTask....325
recentTask....325
timeElapsedOldestTask....325
timeElapsedRecentTask....325
Line Class Constructor....325
Line....325
Line Class Fields....326
call....326
caps....326
connectionid....326
id....326
isMuted....327
state....327
Line Class Methods....327
alternate....327
answer....328
attachUserData....328
clear....328
complete....329
conference....329
conferenceSingle....330
dial....330
dropParty....330
getAni....331
getDescr....331
getDnis....331
getPadvalue....332
getReferenceId....332
getUrl....332
grabCall....333
hold....333
join....334

Copyright © 1988-2007, Oracle. All rights reserved.

mute....334
park....334
reconnect....335
reject....335
release....336
retrieve....336
sendDTMF....336
setcallresult....337
setcallresultDNC....337
setcallresultReschedule....338
transfer....338
transferMute....338
unmute....339
updateCallData....339
Line Class Callback Event Methods....340
onAlternating....340
onAnswering....340
onAttachingUD....340
onCallDataChanged....340
onCapabilitiesChanged....341
onClearing....341
onCompleting....341
onConferencing....341
onDialing....341
onDropped....342
onError....342
onGrabbing....342
onHeld....342
onHolding....342
onJoining....342
onMuted....343
onOffHook....343
onOnHook....343
onParking....343
onPartyAdded....343
onPartyChanged....344
onPartyRemoved....344
onReconnecting....344
onRejected....344
onRejecting....344

Copyright © 1988-2007, Oracle. All rights reserved.

onReleasing....344
onRetrieving....345
onRinging....345
onSetcallresult....345
onSetcallresultDNC....345
onSetcallresultReschedule....345
onTalking....346
onTransfering....346
onUnmuted....346
onUpdatingCD....346
onUserDataChanged....346
LineCaps Class Constructor....346
LineCaps....347
LineCaps Class Fields....347
canAlternate....347
canAnswer....347
canAttachUserDate....347
canClear....348
canComplete....348
canConference....348
canConferenceSingle....348
canDropParty....348
canHold....348
canMute....348
canPark....349
canReconnect....349
canReject....349
canRelease....349
canRetrieve....349
canSendDTMF....349
canSetcallresult....349
canSetcallresultDNC....350
canSetcallresultReschedule....350
canTransfer....350
canTransferMute....350
canUnmute....350
canUpdateCallData....350
MCEvent Class Constructor....350
MCEvent....351
MCEvent Class Fields....351

Copyright © 1988-2007, Oracle. All rights reserved.

extension....351
group....351
reason....351
user....352
MediaType Class Constructor....352
MediaType....352
MediaType Class Field....352
type....352
PSMC Class Constructor....353
PSMC....353
PSMC Class Fields....353
renserver....353
servers....354
sessions....354
PSMC Class Methods....354
closeSession....354
getCallById....354
getChatById....355
getEmailById....355
getGenericTaskById....356
getLineById....356
openSession....356
start....357
stop....357
Reason Class Constructor....357
Reason....358
Reason Class Fields....358
code....358
desc....358
reasonData1....358
reasonData2....359
reasonData3....359
RenServer Class Constructor....359
RenServer....359
RenServer Class Fields....359
isRunning....359
url....360
RenServer Class Callback Event Methods....360
onDown....360
onUp....360

Copyright © 1988-2007, Oracle. All rights reserved.

Server Class Constructor……360
Server……360
Server Class Fields……361
id……361
info……361
state……361
type……361
Server Class Callback Event Methods……362
onBroadcast……362
onHbLost……362
onHbRecovered……362
onInService……363
onOutOfService……363
onRestart……363
Session Class Constructor……363
Session……363
Session Class Fields……364
addresses……364
buddies……364
groups……364
id……364
serverId……364
state……364
user……365
Session Class Methods……365
broadcastSubscribe……365
broadcastUnsubscribe……365
close……366
open……366
registerAddress……366
registerBuddy……367
registerGroup……367
registerUser……367
setAutoRecovery……368
statPublish……368
unregisterAddress……368
unregisterBuddy……369
unregisterGroup……369
unregisterUser……369
Session Class Callback Event Methods……370

Copyright © 1988-2007, Oracle. All rights reserved.

onAddressRegistered....370
onAddressUnregistered....370
onBuddyRegistered....370
onBuddyUnregistered....370
onClosed....371
onError....371
onGroupRegistered....371
onGroupUnregistered....371
onInfo....371
onOpened....372
onUserRegistered....372
onUserUnregistered....372
Task Class Hierarchy....372
Task Class Constructor....373
Task....373
Task Class Fields....374
caseid....374
cost....374
customerid....374
group....374
id....374
onStat....374
priority....375
type....375
urlAbs....375
urlRel....375
TaskStatistics Class Constructor....376
TaskStatistics....376
TaskStatistics Class Fields....376
data....376
holdTime....376
queueTime....376
talkTime....377
User Class Constructor....377
User....377
User Class Fields....377
addresses....378
agentPassword....378
language....378
registeredAddresses....378

Copyright © 1988-2007, Oracle. All rights reserved.

statesUq....379
User Class Methods....379
ctiBusyUq....379
disableMedia....379
enableMedia....380
isMediaEnabled....380
login....381
loginUq....381
logout....382
logoutUq....382
register....382
setNotReady....383
setPresence....383
setPresenceUq....384
setReady....384
setWorkNotReady....385
setWorkReady....385
unregister....386
User Class Callback Event Methods....386
onCapabilitiesChanged....386
onCtiBusy....387
onCTIBUSYUq....387
onCtiClear....387
onDropped....387
onError....387
onInfo....387
onLoggedIn....388
onLoggedOut....388
onLoggingIn....388
onLoggingOut....388
onMediaDisabled....388
onMediaEnabled....389
onNotReady....389
onPresenceChanged....389
onReady....389
onRegistered....389
onRegistering....389
onSettingNotReady....390
onSettingPresence....390
onSettingReady....390

Copyright © 1988-2007, Oracle. All rights reserved.

onSettingWorkNotReady....390
onSettingWorkReady....390
onStat....391
onStat1....391
onStat2....391
onUnknown....391
onUnregistered....391
onUnregistering....391
onWorkNotReady....392
onWorkReady....392
UserCaps Class Constructor....392
UserCaps....392
UserCaps Class Fields....393
canHandleChat....393
canHandleEmail....393
canHandleGeneric....393
canHandleVoice....393
canLogin....393
canLogout....393
canRefreshState....394
canSetNotReady....394
canSetPresence....394
canSetReady....394
canSetWorkNotReady....394
canSetWorkReady....394
UserData Class Constructor....394
UserData....395
UserData Class Fields....395
UserData Class Field Constants....395
UserData Class Method....396
addKeyValue....396
UserStatistics1 Class Constructor....396
UserStatistics1....396
UserStatistics1 Class Fields....397
availableCost....397
ctiBusy....397
currentQueue....397
mostRecentTaskData....397
mostRecentTaskId....397
numTaskAccepted....397

Copyright © 1988-2007, Oracle. All rights reserved.

numTasksDone....398
numTasksUnassigned....398
presenceText....398
reasonFlag....398
state....398
timeInCurrentState....398
timeSinceLoggedIn....398
UserStatistics2 Class Constructor....399
UserStatistics2....399
UserStatistics2 Class Fields....399
currentQueue....399
timeIdle....399
timeInCurrentState....399
timeNotReady....400
timeSinceLogin....400
totalTimeAvailable....400
totalTimeUnavailable....400

Chapter 14
Configuring the Email Channel....401
Understanding the Email Channel....401
Handling Email....402
Configuring PeopleSoft Integration Broker for the Email Channel....403
Pages Used to Configure Integration Broker for Email....403
Configuring the Gateway....403
Configuring GETMAILTARGET Properties....403
Demonstrating the Email Channel....410
Pages Used to Demonstrate the Email Channel....410
Using the GetMail - Server Page....410
Using the MailStore - DB Page....412

Chapter 15
Configuring Instant Messaging....415
Understanding Instant Messaging....415
Configuring Instant Messaging Servers....415
Configuring Instant Messaging....416
Page Used to Configure Instant Messaging....416
Configuring Instant Messaging....416

Copyright © 1988-2007, Oracle. All rights reserved.

Using the Instant Messaging Sample Pages.....417
Pages Used to Test Instant Messaging.....417
Using the PeopleCode Sample Page.....417
Using the Button Sample Page.....418

Appendix A
JSMCAPI Quick Reference.....421
JSMCAPI Classes.....421
_Address.....421
_UQAddress.....421
_User.....422
A2AChat.....423
A2AChatAddress.....424
AgentStatistics.....425
AppData.....426
Buddy.....427
Call.....428
CallData.....429
CallStatistics.....429
Chat.....430
ChatAddress.....431
ChatConnection.....432
ChatConnectionCaps.....434
ChatData.....435
Connection.....435
ConnectionListener.....436
ConnectionRequest.....436
Email.....436
EmailAddress.....438
EmailConnection.....439
EmailConnectionCaps.....440
EmailData.....441
Extension.....441
ExtensionCaps.....443
ForwardMode.....444
GenericAddress.....444
GenericConnection.....445
GenericConnectionCaps.....447
GenericData.....447

Copyright © 1988-2007, Oracle. All rights reserved.

GenericTask....448
GLOBALS....449
Group....450
GroupStatistics....451
GroupStatistics1....451
GroupStatistics2....452
Line....453
LineCaps....456
MCEvent....457
MediaType....458
PSMC....458
Reason....459
RenServer....459
Server....460
Session....461
Task....463
TaskStatistics....464
User....464
UserCaps....467
UserData....468
UserStatistics1....468
UserStatistics2....469

Appendix B
Installing Digital Certificates for REN SSL....471
Installing Digital Certificates....471
Installing the CA Server Certificate....472
Installing REN Server Certificate....472
Configuring Digital Certificates....472
Importing Certificates in Java Key Store....474
Configuring REN Server....474
Configuring REN Clusters....474
Installing Certificates for Local Node....475
Generating Client Certificate....476
Installing PSMCAPI Certificates....476

Copyright © 1988-2007, Oracle. All rights reserved.

Glossary of PeopleSoft Enterprise Terms.....479

Index 505

Copyright © 1988-2007, Oracle. All rights reserved.

About This PeopleBook

PeopleSoft Enterprise PeopleBooks provide you with the information that you need to implement and use PeopleSoft Enterprise applications from Oracle.

This preface discusses:

- PeopleSoft Enterprise application prerequisites.
- Application fundamentals.
- Documentation updates and printed documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common elements in PeopleBooks.

Note. PeopleBooks document only elements, such as fields and check boxes, that require additional explanation. If an element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common elements for the section, chapter, PeopleBook, or product line. Elements that are common to all PeopleSoft Enterprise applications are defined in this preface.

PeopleSoft Enterprise Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use PeopleSoft Enterprise applications.

You might also want to complete at least one introductory training course, if applicable.

You should be familiar with navigating the system and adding, updating, and deleting information by using PeopleSoft Enterprise menus, pages, or windows. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your PeopleSoft Enterprise applications most effectively.

Application Fundamentals

Each application PeopleBook provides implementation and processing information for your PeopleSoft Enterprise applications.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals PeopleBook. Most product lines have a version of the application fundamentals PeopleBook. The preface of each PeopleBook identifies the application fundamentals PeopleBooks that are associated with that PeopleBook.

Copyright © 1988-2007, Oracle. All rights reserved.

The application fundamentals PeopleBook consists of important topics that apply to many or all PeopleSoft Enterprise applications. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals PeopleBooks. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Download and order printed documentation.

Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on Oracle's PeopleSoft Customer Connection website. Through the Documentation section of Oracle's PeopleSoft Customer Connection, you can download files to add to your PeopleBooks Library. You'll find a variety of useful and timely materials, including updates to the full line of PeopleSoft Enterprise documentation that is delivered on your PeopleBooks CD-ROM.

Important! Before you upgrade, you must check Oracle's PeopleSoft Customer Connection for updates to the upgrade instructions. Oracle continually posts updates as the upgrade process is refined.

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Downloading and Ordering Printed Documentation

In addition to the complete line of documentation that is delivered on your PeopleBook CD-ROM, Oracle makes PeopleSoft Enterprise documentation available to you via Oracle's website. You can:

- Download PDF files.
- Order printed, bound volumes.

Downloading PDF Files

You can download PDF versions of PeopleSoft Enterprise documentation online via the Oracle Technology Network. Oracle makes these PDF files available online for each major release shortly after the software is shipped.

See Oracle Technology Network, http://www.oracle.com/technology/documentation/psftent.html.

Ordering Printed, Bound Volumes

You can order printed, bound volumes of selected documentation via the Oracle Store.

See Oracle Store, http://oraclestore.oracle.com/OA_HTML/ibeCCtpSctDspRte.jsp?section=14021

Copyright © 1988-2007, Oracle. All rights reserved.

Additional Resources

The following resources are located on Oracle's PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps
Interactive Services Repository	Support, Documentation, Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Pre-Built Integrations for PeopleSoft Enterprise and JD Edwards EnterpriseOne Applications
Minimum technical requirements (MTRs)	Implement, Optimize + Upgrade; Implementation Guide; Supported Platforms
Documentation updates	Support, Documentation, Documentation Updates
PeopleBooks support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Release Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Copyright © 1988-2007, Oracle. All rights reserved.

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in PeopleBooks:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Indicates field values, emphasis, and PeopleSoft Enterprise or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter *O*.
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press the W key.
`Monospace font`	Indicates a PeopleCode program or other code example.
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).

Copyright © 1988-2007, Oracle. All rights reserved.

Typographical Convention or Visual Cue	Description
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Visual Cues

PeopleBooks contain the following visual cues.

Notes

Notes indicate information that you should pay particular attention to as you work with the PeopleSoft Enterprise system.

Note. Example of a note.

If the note is preceded by *Important!,* the note is crucial and includes information that concerns what you must do for the system to function properly.

Important! Example of an important note.

Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

Warning! Example of a warning.

Cross-References

PeopleBooks provide cross-references either under the heading "See Also" or on a separate line preceded by the word *See.* Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, region, or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a country-specific heading: "(FRA) Hiring an Employee"

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Copyright © 1988-2007, Oracle. All rights reserved.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in PeopleBooks:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in PeopleBooks:

- USF (U.S. Federal)
- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about PeopleBooks and other Oracle reference and training materials. Please send your suggestions to your product line documentation manager at Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. Or email us at appsdoc@us.oracle.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

Common Elements Used in PeopleBooks

As of Date	The last date for which a report or process includes data.
Business Unit	An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.
Description	Enter up to 30 characters of text.
Effective Date	The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row.

Copyright © 1988-2007, Oracle. All rights reserved.

Once, Always, and **Don't Run**	Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run. Select Always to run the request every time the batch process runs. Select Don't Run to ignore the request when the batch process runs.
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
Report Manager	Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Request ID	An ID that represents a set of selection criteria for a report or process.
Run	Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
SetID	An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group.
Short Description	Enter up to 15 characters of text.
User ID	An ID that represents the person who generates a transaction.

Copyright © 1988-2007, Oracle. All rights reserved.

Copyright © 1988-2007, Oracle. All rights reserved.

PeopleSoft MultiChannel Framework Preface

This preface introduces the PeopleSoft MultiChannel Framework.

PeopleSoft MultiChannel Framework

PeopleSoft MultiChannel Framework delivers an integrated infrastructure to support multiple interaction channels for call center agents or other PeopleSoft users who must respond to incoming requests and notifications on these channels.

PeopleSoft MultiChannel Framework supports voice, email, web-based chat, instant messaging, and generic event channels.

Additional Resources

The following resources are located on the PeopleSoft Customer Connection web site:

Resource	Navigation
Hardware and software requirements	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation and Software, Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation and Software, Installation Guides and Notes
PeopleBook documentation updates	Support, Documentation, Documentation Updates
Troubleshooting information	Support, Troubleshooting

Copyright © 1988-2007, Oracle. All rights reserved.

Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 1

Getting Started with PeopleSoft MultiChannel Framework

This chapter discusses:

- PeopleSoft MultiChannel Framework overview.
- PeopleSoft MultiChannel Framework implementation.

PeopleSoft MultiChannel Framework Overview

PeopleSoft MultiChannel Framework provides the tools that are required to support multiple channels of communication between customers (users) and agents. Some PeopleSoft applications use PeopleSoft MultiChannel Framework, such as an email response management system (ERMS) from PeopleSoft CRM, or you can develop your own applications on the provided framework.

PeopleSoft MultiChannel Framework delivers an integrated infrastructure to support multiple interaction channels for call center agents or other PeopleSoft users who must respond to incoming requests and notifications on these channels. The channels supported include email, web-based chat, voice, instant messaging, and generic events. PeopleSoft MultiChannel Framework can be used from any PeopleSoft application.

PeopleSoft MultiChannel Framework also supports a broadcast function, which allows a user, such as a supervisor, to broadcast a notification message to a group of agents.

PeopleSoft MultiChannel Framework enables third-party routing systems that enable applications, such as PeopleSoft CRM, to provide embedded multi-channel functionality. You may choose either the PeopleSoft queue server or the third-party routing server to route voice, email, chat and generic events.

In the planning phase of your implementation, take advantage of all PeopleSoft sources of information, including the installation guides, PeopleTools documentation, and the PeopleBooks that are specific to your applications.

PeopleSoft MultiChannel Framework Implementation

This section describes the steps that are required to implement PeopleSoft MultiChannel Framework (MCF).

Depending on your business use of PeopleSoft MultiChannel Framework, several activities are necessary for implementation:

- At a minimum you must configure a real-time event notification (REN) server.

Copyright © 1988-2007, Oracle. All rights reserved.

- If you are using a PeopleSoft-supplied application, such as ERMS, you must also configure MCF servers, clusters, queues, and agents.
- You can also develop your own applications built on the PeopleSoft MultiChannel Framework.
- If you are using PeopleSoft CTI or other third party MCF integrations, additional configuration is required.

Configuring REN Servers and Clusters

The REN server routes event notifications through the PeopleSoft MultiChannel Framework. Certain PeopleSoft applications, such as Reporting and Optimization, use the REN server to route notifications without using any of the rest of the PeopleSoft MultiChannel Framework. Therefore, a minimal configuration of PeopleSoft MultiChannel Framework includes at least one REN server and REN server cluster.

REN-server specific security setup is required, including configuration of Real-time Event Notification Permissions for each role using a REN server alone or as part of PeopleSoft MultiChannel Framework.

To provide a secure channel of communication between the clients and the REN servers, the REN servers may be SSL-enabled. SSL-enabled REN servers enable secure communication by providing client and server authentication

Typically, a system administrator configures and manages REN servers and clusters.

See Chapter 5, “Configuring REN Servers,” page 55.

Configuring MCF Servers, Clusters, Queues, and Tasks

MCF servers (queue servers and log servers) work with REN servers to queue and route task notifications to agents. An implementation of PeopleSoft MultiChannel Framework requires configuration of MCF servers, MCF clusters, queues, and tasks.

Typically, a system administrator configures and manages MCF servers and clusters, and either a system administrator or an agent supervisor configures and manages queues and tasks.

See Chapter 6, “Configuring PeopleSoft MCF Servers and Clusters,” page 77 and Chapter 7, “Configuring PeopleSoft MCF Queues and Tasks,” page 85.

Creating and Defining Agents

Agents are PeopleSoft users that are further defined as PeopleSoft MultiChannel Framework agents. Agents manage tasks that are assigned to them. Agents can log on to and accept tasks from the MultiChannel Console.

Typically, an agent supervisor defines agents and assigns agents to appropriate queues.

See Chapter 8, “Configuring PeopleSoft MCF Agents,” page 103.

Using MCF Sample Pages

PeopleSoft MultiChannel Framework is delivered with sample pages that demonstrate the use of email, web-based chat, generic, and instant messaging channels. An application developer can use these sample page definitions as the basis for application pages. The *Enterprise PeopleTools 8.48 PeopleBook: PeopleCode API Reference* includes detailed descriptions of PeopleSoft’s Mail Classes, MCFIMInfo Classes, and the Universal Queue Classes.

Additional sample pages demonstrate how to use the JavaScript MultiChannel Application Programming Interface (JSMCAPI) to customize the CTI console and to develop supervisor desktops using the available monitoring functions.

See Chapter 9, “Administering Queues, Logs, and Tasks,” page 113.

 Copyright © 1988-2007, Oracle. All rights reserved.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Mail Classes"; *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "MCFIMInfo Class" and *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Universal Queue Classes".

Configuring PeopleSoft CTI

PeopleSoft MultiChannel Framework includes support for PeopleSoft CTI. PeopleSoft CTI requires supporting computer-telephony middleware and additional configuration separate from other channels of PeopleSoft MultiChannel Framework.

See Chapter 3, "Configuring PeopleSoft CTI," page 13.

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 2

Understanding PeopleSoft MultiChannel Framework

This chapter discusses:

- PeopleSoft MultiChannel Framework.
- PeopleSoft MultiChannel Framework elements and channels.
- PeopleSoft MultiChannel Framework architecture.

PeopleSoft MultiChannel Framework

PeopleSoft MultiChannel Framework delivers an integrated infrastructure to support multiple interaction channels for call center agents or other PeopleSoft users who must respond to incoming requests and notifications on these channels.

PeopleSoft MultiChannel Framework can be used from any PeopleSoft application.

In this context, the word *channel* refers to the technology used to communicate during an interaction. PeopleSoft MultiChannel Framework supports the following channels:

- Voice (telephone).
- Web collaboration (chat).
- Email.
- Generic tasks.
- Instant messaging.

The PeopleSoft MultiChannel Framework includes an HTML agent console, universal queueing, real-time task routing, customer-to-agent and collaborative chat, and centralized event logging.

PeopleSoft MultiChannel Framework Elements and Channels

This section discusses:

- PeopleSoft MultiChannel Framework elements.
- PeopleSoft MultiChannel Framework channels.
- PeopleSoft MultiChannel Framework universal queue.

Copyright © 1988-2007, Oracle. All rights reserved.

- PeopleSoft MultiChannel Console.

PeopleSoft MultiChannel Elements

PeopleSoft MultiChannel Framework comprises the following services and elements:

- Universal queue, running on the Universal Queue server process (PSUQSRV).
- Real-time event notification (REN) server, running on the REN server process (PSRENSRV).
- MultiChannel framework (MCF) log, running on the MCFLOG server process (PSMCFLOG).
- MultiChannel console, the HTML interface through which users manage the channel interactions assigned to them.
- Chat windows, the HTML interfaces used for customer-to-agent and collaborative chat sessions.
- Agents, identified by their expertise and responsibilities.
- PeopleCode built-in functions and an email application package.
- GETMAILTARGET connector running under PeopleSoft Integration Broker.
- PeopleSoft MultiChannel Application Programming Interface (PSMCAPI) to enable server-side CTI integration.
- JavaScript MultiChannel Application Programming Interface (JSMCAPI) to enable a customizable CTI console and monitoring functions.

Each of these services and elements requires configuration.

In addition, each communication channel handled by PeopleSoft MultiChannel Framework requires supporting elements:

- Computer-telephony integration (CTI) middleware to notify the system of telephone calls.
- An email server to store and serve email.
- Application pages to request customer-to-agent chat sessions and to provide context data and resolution logic for all interactions.
- Application pages or batch processes to enqueue generic events.

PeopleSoft MultiChannel Framework Channels

This section discusses support for communications channels offered by PeopleSoft MultiChannel Framework.

Voice

The agent console offers a softphone and full CTI support with Genesys and Cisco CTI middleware, or any middleware implementing the PSMCAPI. Relevant application pages appear based on data attached to the call by the Interactive Voice Response (IVR) and CTI middleware.

Web Collaboration

PeopleSoft application pages can include Live Help buttons that initiate customer-to-agent chat sessions. The customer and agent chat windows are browser-based and do not require a client installation or applet download. The universal queue routes chat requests to the first available agent with the skills required to handle that request. The agent chat window displays relevant customer information and enables the agent to push web content to the customer. The agent can manage multiple chat sessions from the agent console.

 Copyright © 1988-2007, Oracle. All rights reserved.

Agents can also conference peers and supervisors into a chat and transfer chat sessions to other agents or queues. Agents can also initiate collaborative chats with other agents on their buddy lists.

Email

PeopleSoft MultiChannel Framework enables applications to fetch Multipart Internet Mail Extensions emails from Post Office Protocol 3 (POP3) and Internet Message Access Protocol 4 (IMAP4) mail servers, store their parts in a database, and route the email to call center agents by either adding the email to worklists or enqueueing them on the universal queue. Large emails and binary attachments are not inserted into the database, but are instead stored in an attachment repository, accessible by URLs from a browser. The repository checks user and role-based security before retrieving an attachment. The email framework is built on PeopleSoft Integration Broker technology.

PeopleSoft MultiChannel Framework supports emails conforming to the SMTP specifications including both inbound and outbound HTML email.

Generic Channel

Channels that are not provided by PeopleSoft MultiChannel Framework can be integrated by using the generic channel to enqueue tasks onto the universal queue.

Instant Messaging

PeopleSoft MultiChannel Framework includes the capability to use instant messaging called from an application page and using one of several instant messaging clients, like AOL, Yahoo, or Sametime.

PeopleSoft MultiChannel Framework Universal Queue

The universal queue accepts, evaluates, and distributes incoming task requests from multiple communication channels: email, web chat, and generic notifications.

The universal queue handles email, chat, and generic tasks. It distributes workload across the call center, or any other pool of qualified users, based on the priority of the task and the availability of agents possessing the required skill level and language skills. Availability is based on agent presence and the cost of the new task (a measure of the task's impact on agent capacity) against the current workload of each agent.

Agents can forward tasks to other agents or to another queue. The task is removed from the transferring agent's workload and added to the accepting agent's workload.

Email and generic tasks that are not closed before the agent signs out persist in the database. Persisted tasks are reassigned to the same agent that accepted the tasks when the agent signs in again. A task that is not accepted, within configurable time limits, by the agent to whom it was assigned is reassigned to another qualified agent, if one is available. Tasks that are not resolved within configurable time limits are automatically escalated. Tasks that cannot be assigned to or are not accepted by any agent within configurable time limits are moved to an overflow table.

Voice tasks (CTI) are not queued or routed by the universal queue. They take precedence over all other tasks. However, the queue server adds the cost of voice tasks to the agent workload calculations it uses to queue and assign incoming tasks.

Copyright © 1988-2007, Oracle. All rights reserved.

PeopleSoft MultiChannel Console

The agent console is the web-browser-based desktop from which the user manages all tasks, irrespective of channel. The console combines CTI, chat, email, and generic notice response tools into one configurable window. Agents use the console to sign in, to select their current queue, to accept tasks, and to initiate and accept collaborative chat requests with buddy users. After accepting a task, additional browser-based windows appear to enable the agent's response. These windows are task-dependent and include elements developed specifically for supporting applications, such as email response management systems.

A custom CTI console can be created using the JSMCAPI.

PeopleSoft MultiChannel Framework Architecture

This section discusses:

- PeopleSoft MultiChannel Framework server architecture.
- Chat architecture.
- Email architecture.

PeopleSoft MultiChannel Framework Server Architecture

The REN server (the PSRENSRV process) is essential to the framework architecture. MCF events are sent to REN servers, which then deliver them to recipients of those topics. The REN server is a modified web server using the HTTP 1.0 or HTTP 1.1 communications protocol. Communication with server processes and MCF browser windows is bidirectional, as the browser windows maintain persistent connections to the REN server. Events can be sent proactively to browser windows without polling or page refreshes. To provide a secure channel of communication to overcome concerns from PeopleSoft customers about sensitive data and its security, the REN server can be SSL-enabled. An SSL-enabled REN server provides secure communication that encrypts and provides client and server authentication.

Applications send interaction and action requests (tasks) received from the supported communication channels to logical queues. The REN server notifies the universal queue server (the PSUQSRV process) responsible for that queue that a new task has arrived. Tasks are queued in order of priority until they can be assigned to an available agent qualified to respond to the task, at which time the queue server sends an assignment notification to the user's MultiChannel Console through the REN server.

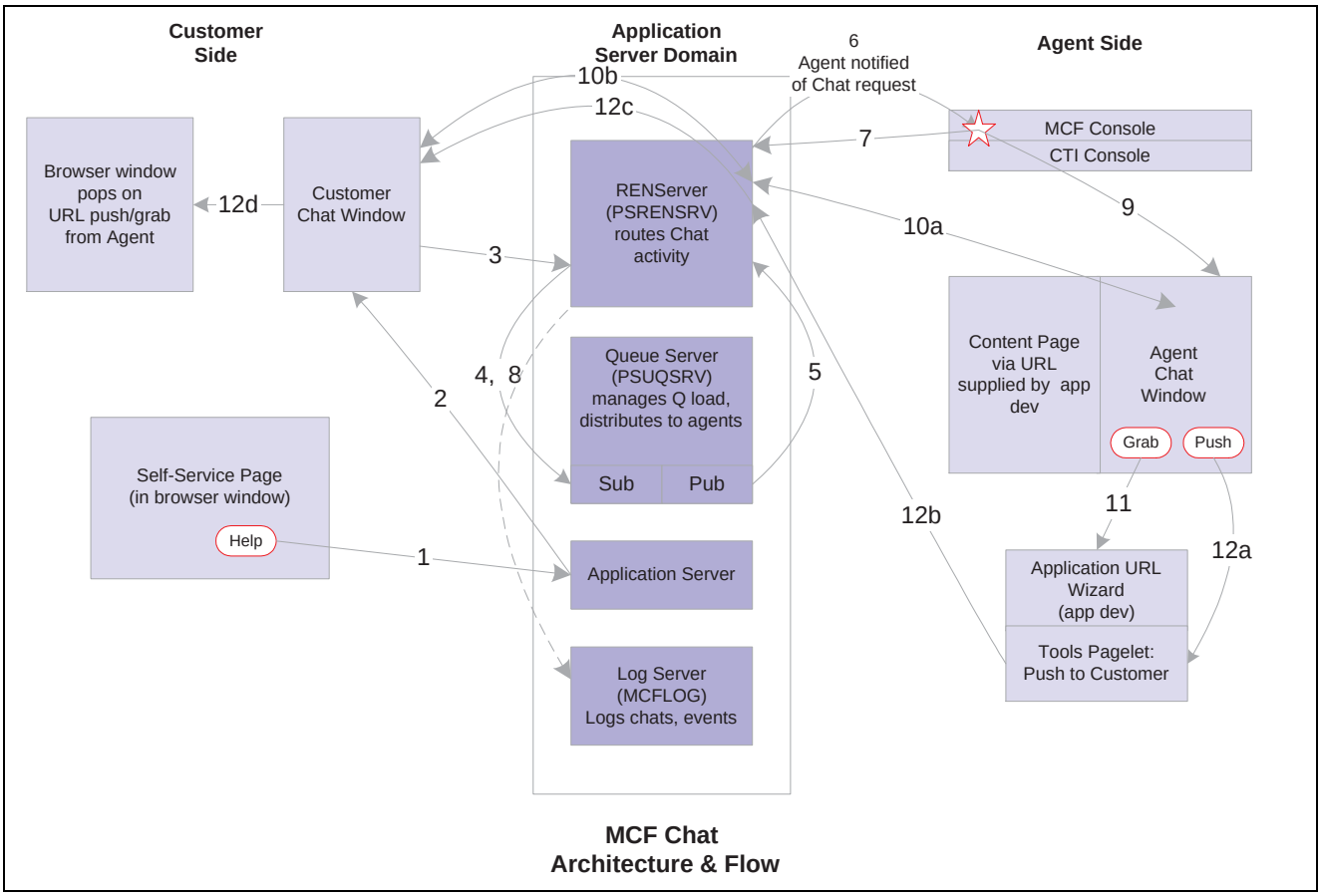
The queue server routes work requests (tasks) to users based upon a set of configurable policy properties that define which agents can handle what types of tasks and when the tasks can be assigned. The queue server manages state information about the current status of active agents and active tasks.

The MCF log server logs MCF events and chat content to the database. Configure logging levels on the MCF administration pages.

Chat Architecture

This diagram illustrates the architecture of a chat session:

 Copyright © 1988-2007, Oracle. All rights reserved.



Chat architecture and flow

When a customer clicks the Help button on an application page:

1. InitChat() passes the following parameters to the application server:
   - Queue number.
   - Priority override.
   - Context page URL.
   - Query.

   See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions".
2. The application server posts to the REN server to notify the queue server that a customer chat is waiting.

   The application server then returns the name and port of the REN server and the iScript to build the customer chat window.
3. The customer chat window appears and communicates with the REN server to receive all events on that chat topic.
4. The REN server notifies the queue server that there is a customer chat waiting.

   The queue server determines the appropriate agent according to workload, cost, agent availability, skill level, and language.
5. The queue server tells the REN server to notify the agent that the agent has been assigned a chat.

Copyright © 1988-2007, Oracle. All rights reserved.

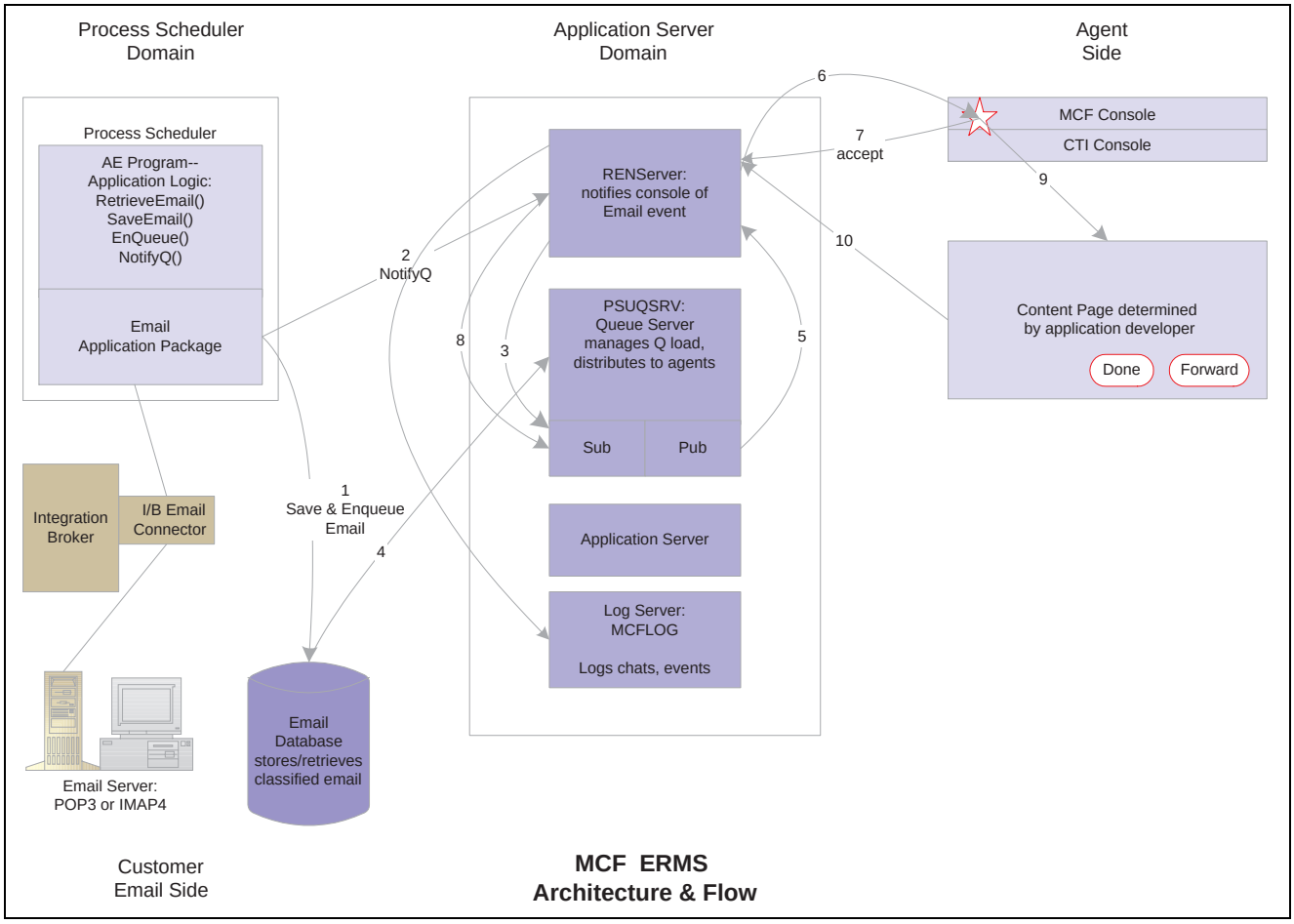
6. The REN server notifies the agent from the MultiChannel Console that the agent has been assigned a chat.
7. The agent, from the MultiChannel Console, notifies the REN server to notify the queue server that the agent has accepted the task.
8. The REN server notifies the queue that the agent has accepted the task.
9. The MultiChannel Console displays an agent chat window.

   The agent responds to inquiry.
10. Agent to customer two-way communication is mediated by the REN server.
11. If the agent chooses to grab a URL, the Grab button invokes the application URL wizard.
12. If the agent chooses to push a URL, the Push button invokes a pagelet (labeled with an *A* in the diagram) to push a selected URL through the REN server (labeled with a *B* in the diagram) and customer chat window (labeled with a *C* in the diagram) to a new browser window.

The entire chat session can be logged through the MCF log server (labeled with a dotted line in the preceding diagram).

Email Architecture

This diagram illustrates the architecture of email processing:



Email architecture and flow

Copyright © 1988-2007, Oracle. All rights reserved.

When a customer sends an email:

1. A PeopleSoft Application Engine program uses the MCF email application package classes and PeopleCode built-in functions to save and enqueue email in a database.
2. The PeopleSoft Application Engine program notifies the REN server that email has been enqueued.
3. The REN server notifies the queue server of the waiting email.
4. The queue server retrieves email information required to determine appropriate routing from the database.

   The queue server determines the appropriate agent to handle each email according to workload, cost, agent availability, skill level, and language.
5. The queue server tells the REN server to notify the agent that the agent has been assigned an email.
6. The REN server notifies the agent from the MultiChannel Console that the agent has been assigned an email.
7. The agent, from the MultiChannel Console, notifies the REN server to notify the queue server that the agent has accepted the task.
8. The REN server notifies the queue that the agent has accepted the task.
9. The MultiChannel Console displays an agent email window, as determined by the application developer.

   The agent responds to the email.
10. The agent's resolution of the task is communicated back to the REN server by either the Done or Forward button.

Email events can be logged to a database by the MCF log server.

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 3

Configuring PeopleSoft CTI

This chapter provides an overview of the PeopleSoft Computer Telephony Integration (CTI) console and PeopleSoft CTI components and discusses how to:

- Configure PeopleSoft CTI.
- Configure queues and agents.
- Use other CTI options.
- Use the CTI sample pages.

Understanding PeopleSoft CTI

This section discusses:

- PeopleSoft CTI.
- PeopleSoft CTI components.

PeopleSoft CTI

PeopleSoft CTI enables you to integrate your PeopleSoft applications with your call center. PeopleSoft CTI offers the following benefits:

- Seamlessly integrates your PeopleSoft application with the Genesys CTI Framework, Cisco Intelligent Contact Management (ICM) software, or other CTI software that implements the PeopleSoft MultiChannel Application Programming Interface (PSMCAPI), to improve agent productivity.

  Agents refers to the individuals who interact with your customers using CTI.
- Requires only that you install a supported web browser on the agent's workstation.
- Enables agents to take advantage of browser-based call management and automatic population of PeopleSoft transaction pages with the relevant customer data associated with an incoming call.
- Transmits DTMF (Dual-Tone Multi Frequency) data.
- Handles outbound calls from automated system.

PeopleSoft CTI is an optional component that you can integrate with the PeopleSoft MultiChannel Framework. This means that you have the option to incorporate a CTI channel within the MultiChannel Framework. PeopleSoft CTI requires third-party middleware in the form of the Genesys CTI Framework, a Cisco ICM system, or CTI middleware that implements PSMCAPI.

In PeopleSoft CTI, the CTI middleware performs the call routing. The universal queue is not involved in routing calls. For an incoming call, the CTI middleware notifies the MultiChannel Console, which then notifies the queue server so that the agent's workload can be updated with the cost of a call.

Copyright © 1988-2007, Oracle. All rights reserved.

PeopleSoft CTI Components

The PeopleSoft CTI console works together with the IVR (Interactive Voice Response) system, the CTI middleware, an Automatic Call Distributor (ACD), and your PeopleSoft application.

When a customer calls, the caller enters their information (for example, an account number) using the IVR system. Using this information, the CTI middleware routes the call to an appropriate ACD queue. The ACD sends the call to the next available agent on that queue and notifies the CTI middleware that there is an incoming call on that Directory Number (DN). The CTI middleware, in turn, notifies the CTI Console, and passes the customer's information as attached data. The CTI Console uses the attached data to determine what PeopleSoft transaction page to open (pop-up) for the agent and what application data to retrieve from the database. The agent can manage the call using the CTI Console, which in turn communicates with the PBX using the CTI middleware.

For PeopleSoft CTI systems using Enterprise PeopleTools 8.45.05 or later, the CTI console uses a JavaScript MultiChannel Application Programming Interface (JSMCAPI) to communicate with CTI middleware that implements the PeopleSoft MultiChannel Application Programming Interface (PSMCAPI).

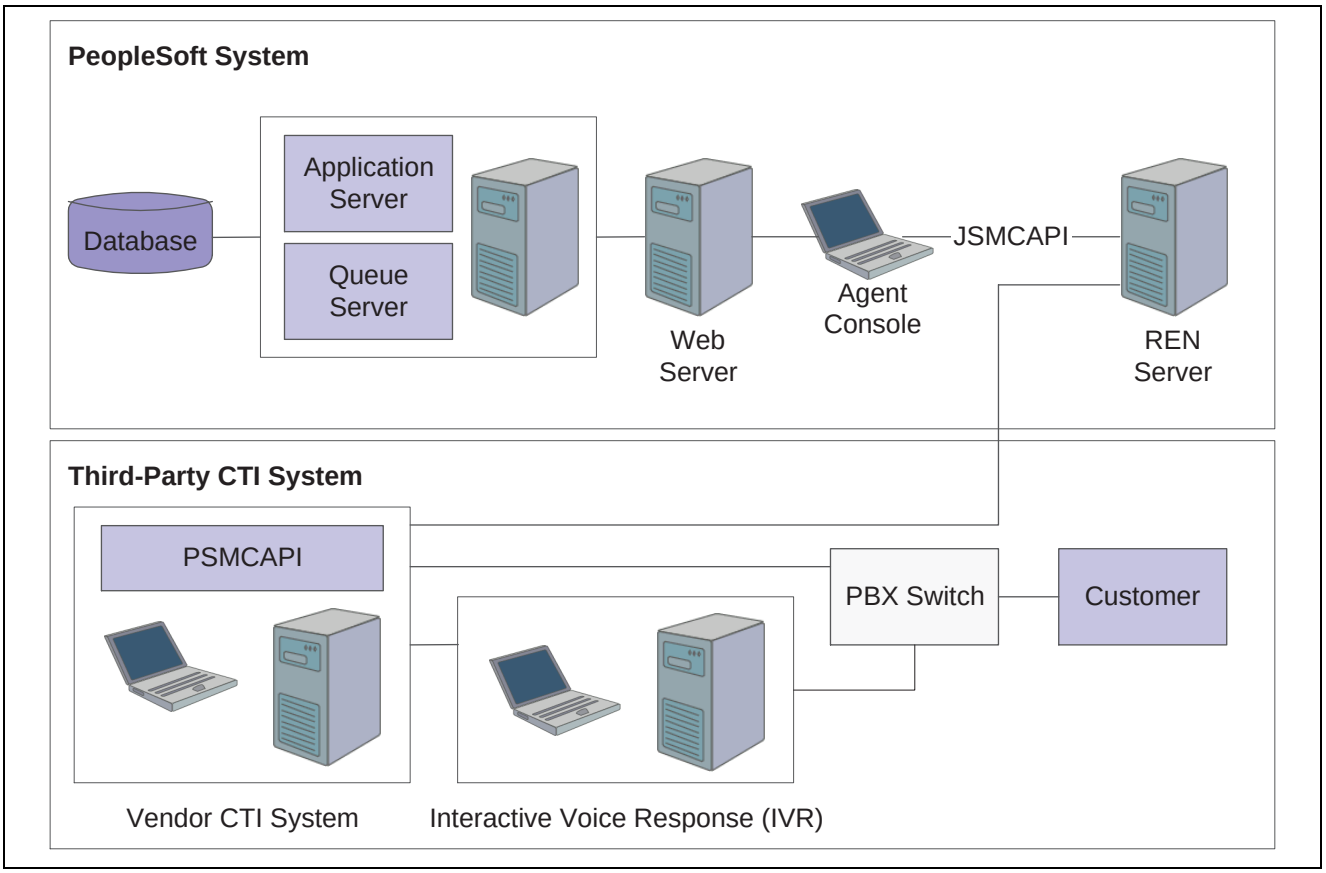
For PeopleSoft CTI systems using Enterprise PeopleTools 8.44 or earlier, the CTI console uses a Java applet that runs within the web browser to communicate directly with your CTI middleware. The CTI console communicates to the Java applet using JavaScript. The applet is delivered in a file called pCti.cab that is approximately 500 KB. The applet resides in the browser cache to reduce network traffic and improve response time.

Note. The CTI console Java applet was deprecated in PeopleTools 8.46.

Configure PeopleSoft CTI to use either the applet or the non-applet (JSMCAPI) console on the CTI Type tab on the Configure CTI page.

This diagram illustrates PeopleSoft CTI architecture using PSMCAPI and JSMCAPI:

 Copyright © 1988-2007, Oracle. All rights reserved.



PeopleSoft CTI integration architecture

Note. PeopleSoft supports both Cisco and Genesys CTI solutions using the CTI Java applet, and any CTI solution that implements the PSMCAPI. In this document, when the phrase *your CTI vendor* or *CTI middleware* appears, assume that it refers to either Cisco or Genesys products or to other CTI vendors that support the PSMCAPI, as appropriate to your installation.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference, "Internet Script Classes (iScript)"

Understanding PeopleSoft CTI Components

This section describes components required for PeopleSoft CTI. It discusses:

- PeopleSoft CTI requirements.
- PeopleSoft MultiChannel API.
- JavaScript MultiChannel API.
- Required security for PSMCAPI.
- Genesys components.
- Cisco components.

Copyright © 1988-2007, Oracle. All rights reserved.

- Java Runtime Environment.

PeopleSoft CTI Requirements

PeopleSoft CTI requires one of the following:

- CTI middleware implementing the PSMCAPI.
- Genesys CTI and the Java Runtime Environment.
- Cisco CTI and the Java Runtime Environment.

PeopleSoft MultiChannel API

The PeopleSoft MultiChannel API (PSMCAPI) is a Java API and software development kit (SDK) that provides server-side connectivity with the PeopleSoft CTI system. PSMCAPI enables third-party telephony vendors and system integrators to integrate with PeopleSoft applications.

No additional installation is required to use PSMCAPI or JSMCAPI; they are installed during PeopleTools installation.

This table provides the locations of PSMCAPI and JSMCAPI components:

Item	Location
PSMCAPI Java archive	<PS_HOME>\sdk\psmcapi\dist\lib
PSMCAPI configuration files: • psmcapilog.properties • renclient.properties	<PS_HOME>\sdk\psmcapi\dist\config
JSMCAPI JavaScript file	<PS_HOME>\webserv\<domain>\applications\peoplesoft\PORTAL\ps\pMCF

Configuring PSMCAPI

Two files, psmcapilog.properties and renclient.properties, include parameters to configure PSMCAPI. Configure the logging characteristics of PSMCAPI in psmcapi.properties. To have PSMCAPI generate full debug logs, set com.peoplesoft.pt.mcf.level to FINEST.

The following table lists all the parameters in renclient.properties:

Parameter	Default Value	Description
interval_heartbeat_server	30000	Heartbeat interval in milliseconds from PSMCAPI to console.
interval_heartbeat_client	30000	Heartbeat interval in milliseconds from console to PSMCAPI.
maxsize_eventqueue	100000	Maximum number of events in the event queue.

 Copyright © 1988-2007, Oracle. All rights reserved.

Parameter	Default Value	Description
interval_event_expired	300000	Expiration interval in milliseconds for an event in the event queue. An expired event is never published to the client/console and is discarded from the event queue.
interval_topic_reaper	3600000	Topic reaper interval in milliseconds.
mtu_size	0	The Maximum Transmission Unit size of your computer or network. Set mtu_size to zero for no TCP packet padding, or to the MTU size for your network or computer to remove TCP acknowledgement delays.
heartbeats_to_miss	2	The number of heartbeat intervals to wait before removing a non-responsive client
psmcapi_heartbeats_to_miss	5	The number of PSMCAPI heartbeat intervals to wait before removing a non-responsive client.
tcp_nodelay	True	TCP no delay. Set to true to turn off TCP Nagle algorithm
disable_session	False	This parameter is used when multiple PSMCAPI implementations subscribe to same REN server. If this parameter is set to True, it turns off the listeners for internal session and heartbeat.

Required Security for PSMCAPI

PeopleSoft CTI solutions that use the PeopleSoft MultiChannel Application Programming Interface (PSMCAPI) require CTI-specific configuration of REN server security to authenticate the CTI server. The CTI server is not part of a PeopleSoft application server domain, but it can generate an authentication token (PSTOKEN) to be authenticated by the application server. To enable this authentication, PeopleSoft delivers a default CTI server permission list and role for use exclusively by the CTI server.

You must create a CTI server user profile (such as CTISRV) and assign the CTI server role, PeopleTools CTI Server, to it. Be sure to specify the user profile password and symbolic ID, and set the ID type as *None.*

This table lists CTI server security objects for use with PSMCAPI:

Security Object	Name
CTI server permission list	PTPT1700

Copyright © 1988-2007, Oracle. All rights reserved.

Security Object	Name
CTI server security object (set for full access on PeopleTools, Realtime Event Notification Permissions page for specified permission list)	MCF CTI Server
CTI server role	PeopleTools CTI Server

CTI agents only require that the MCF Agent security object be enabled on the REN Permissions page of their associated permission list. The MCF Agent security object includes security for both CTI and other MCF channels.

To enable CTI configuration pages, authorize the CTI Type, CTI Applet, and CTI Non-Applet panel items on the PT_CTI page permissions for the appropriate permission list.

Also enable WEBLIB_MCF weblib permissions for the appropriate permission list.

See Also

Chapter 5, "Configuring REN Servers," Configuring REN Servers, page 60

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, "Getting Started with Security Administration"

JavaScript MultiChannel API

The JavaScript MultiChannel Application Programming Interface (JSMCAPI) is an interface that can be used by application developers to generate the CTI console or to enable CTI functionality on a PeopleSoft Pure Internet Architecture page. The JSMCAPI builds on the real-time event notification (REN) JavaScript client. JSMCAPI uses standard JavaScript.

See Also

Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using and Demonstrating JSMCAPI, page 155

Genesys Components

If you are using a Genesys CTI system with the PeopleSoft CTI Java applet console, the assumption is that you already have a functioning Genesys system configured at your site. Genesys Java API is shipped as part of the CTI applet; however, no other Genesys products are shipped with the PeopleSoft system.

To interact with the PeopleSoft CTI system, you need a Genesys T-Server installed and configured before you begin installing your PeopleSoft CTI system. Refer to your Genesys documentation for installation information.

In addition, your PeopleSoft application may require the following CTI components to attach data to incoming calls:

- Genesys Strategy Builder or Interaction Router or equivalent.
- An IVR supported by Genesys and capable of passing call data to Genesys.

For detailed information regarding specific versions that PeopleSoft supports, refer to the PeopleSoft Platforms database on the Customer Connection web site.

 Copyright © 1988-2007, Oracle. All rights reserved.

Note. The Genesys Configuration Server is an optional but supported component.

Cisco Components

If you are using a Cisco CTI system with the PeopleSoft CTI Java applet console, the assumption is that you already have a functioning and configured Cisco system at your site. The Cisco Java API is shipped as part of the CTI applet; however, no other Cisco products are shipped with the PeopleSoft system.

You need a Cisco ICM Central Controller Server installed and configured before you begin installing your PeopleSoft CTI system.

In addition, your PeopleSoft application may require an IVR supported by Cisco and capable of passing call data to Cisco. This enables CTI components to attach data to incoming calls.

Note. For detailed information about specific PeopleSoft-supported versions, refer to the PeopleSoft Platforms database on the Customer Connection web site.

Java Runtime Environment

Java applets, such as the PeopleSoft CTI Java applet console, run within a Java Virtual Machine (JVM) that uses a *sandbox* (a controlled environment in which remote programs run) to restrict access to what the applets can do. The Java Runtime Environment is required for those PeopleSoft CTI systems using the Java applet console. The JRE is not required if you are using the JSMCAPI console.

If you are using the Java applet console, you should install and use the Sun Java Plug-in 1.4.2 or later. You can download the plug-in from Sun Microsystems at www.java.com.

Applet Protection

The CTI applet has an attached digital signature, which enables the PeopleSoft CTI applet to open a network connection to your CTI system as opposed to being able to only send data back to the web server that downloaded it.

The digital signature protects the applet against tampering, and it requires that the agent grant permission to run the applet. When doing so, agents need to indicate whether such permission should be granted to any PeopleSoft code for all subsequent sessions or just for the current session only. Instruct your end users to select the appropriate option for your site.

See Also

Supported web browsers on the Customer Connection web site.

Browser limitations on the Customer Connection web site.

Configuring PeopleSoft CTI Using CTI Server or an Integrated Third-Party Routing System

This section describes how to install and configure the PeopleSoft CTI system using the CTI server or the third-party routing system. This information assumes that you already have a functioning CTI system installed and configured at your site.

Copyright © 1988-2007, Oracle. All rights reserved.

To configure PeopleSoft CTI, use the CTI Configuration (PT_CTI_CONFIG) and Miscellaneous (PT_CTI_MISC) components for the queue server and CTI Third-Party (MCF_TP_CTI_CONFIG) component for the third-party routing system

This section discusses how to:

- Install PeopleSoft CTI.
- Enable PeopleSoft CTI.
- Create CTI configurations.
- Configure the CTI non-applet console.
- Configure the CTI applet console.
- Use the Shared Phone Book page.
- Use the CTI Genesys page.
- Use the CTI Cisco page.
- Use the Miscellaneous page.
- Use the Reason Code page.

Pages Used to Configure CTI Using CTI Server or an Integrated Third-Party Routing System

Page Name	Object Name	Navigation	Usage
CTI Type	CTICONSOLETYPE	• PeopleTools, MultiChannel Framework, CTI, CTI Type • PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, CTI Type	Enter whether you are using the CTI Java applet console or the CTI JavaScript (non-applet) console.
CTI Non-Applet	CTIREN	• PeopleTools, MultiChannel Framework, CTI, CTI Non-Applet • PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, CTI Non-Applet	Configure CTI for use with the JavaScript console.
CTI Applet	PT_CTI_CONFIG	• PeopleTools, MultiChannel Framework, CTI, CTI Applet • PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, CTI Applet	Configure CTI for use with the Java applet console.

 Copyright © 1988-2007, Oracle. All rights reserved.

Page Name	Object Name	Navigation	Usage
Shared Phone Book	PT_CTI_PNCTI	• PeopleTools, MultiChannel Framework, CTI, Shared Phone Book • PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, Shared Phone Book	Create a list of frequently dialed phone numbers for use with a specified configuration.
CTI Cisco	PT_CTI_CSCO	• PeopleTools, MultiChannel Framework, CTI, CTI Cisco • PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, CTI Genesys	Configure parameters specific to Cisco CTI middleware for use with the Java applet console.
CTI Genesys	PT_CTI_GENESYS	• PeopleTools, MultiChannel Framework, CTI, CTI Genesys • PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, CTI Genesys	Configure parameters specific to Genesys CTI middleware for use with the Java applet console.
Miscellaneous	PT_CTI_MISC	PeopleTools, MultiChannel Framework, CTI Configuration, Miscellaneous	Enter default screen pop-up window URL. **Note.** This page is available for CTI server only.
Reason Code	CTI_SYSREASON	PeopleTools, MultiChannel Framework, Third-Party Configuration, CTI Configuration, Reason Code	Define your own customized codes when using third-party routing system. **Note.** This page is available for third-party routing system only.

Installing PeopleSoft CTI

When you run the PeopleSoft Pure Internet Architecture setup program, the PeopleSoft CTI files are installed automatically to your web server.

Note. You do not need to select any additional options from the install program dialogs. The CTI files are installed by default.

After you have run the PeopleSoft Pure Internet Architecture setup program, enable the PeopleSoft CTI console and configure the system as discussed in the following sections.

Copyright © 1988-2007, Oracle. All rights reserved.

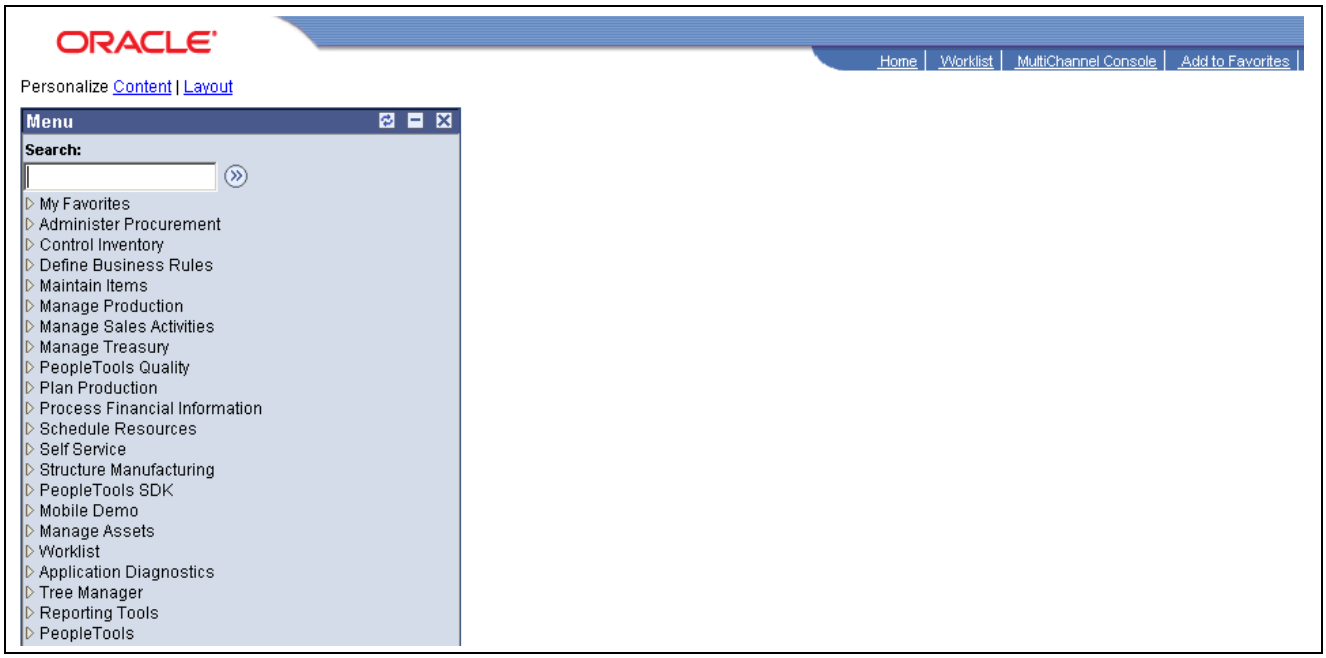
See Also

PeopleTools Installation Guide for your platform

CTI vendor product documentation

Enabling PeopleSoft CTI

If a user is set up as a CTI agent or a MultiChannel Framework agent, the Multichannel Console link appears in the universal navigation header (on the upper-right portion of the screen) when they access the PeopleSoft Pure Internet Architecture. The following example shows the Multichannel Console link displaying in the PeopleSoft Pure Internet Architecture:



MultiChannel Console link displaying

Users who do not have full access to the MCF Agent object do not see this link.

Note. Blended agents—agents who use both the CTI and MCF consoles—must log on to queues supported by the same REN server cluster.

See Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using the CTI Sample Console, page 160.

See Also

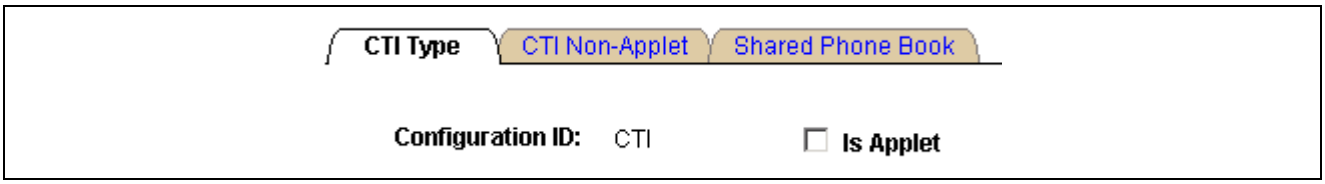
Enterprise PeopleTools 8.49 PeopleBook: Internet Technology, “Understanding Portal Technology”

Chapter 3, “Configuring PeopleSoft CTI,” Configuring CTI Agents, page 33

Creating CTI Configurations

Access the CTI Type page.

 Copyright © 1988-2007, Oracle. All rights reserved.



CTI Type page

Select the Is Applet if you are using the CTI Java applet console.

Clear the Is Applet check box if you are using the CTI JavaScript (JSMCAPI) console.

A CTI configuration contains all the information required for a user to be able to connect to one of the following:

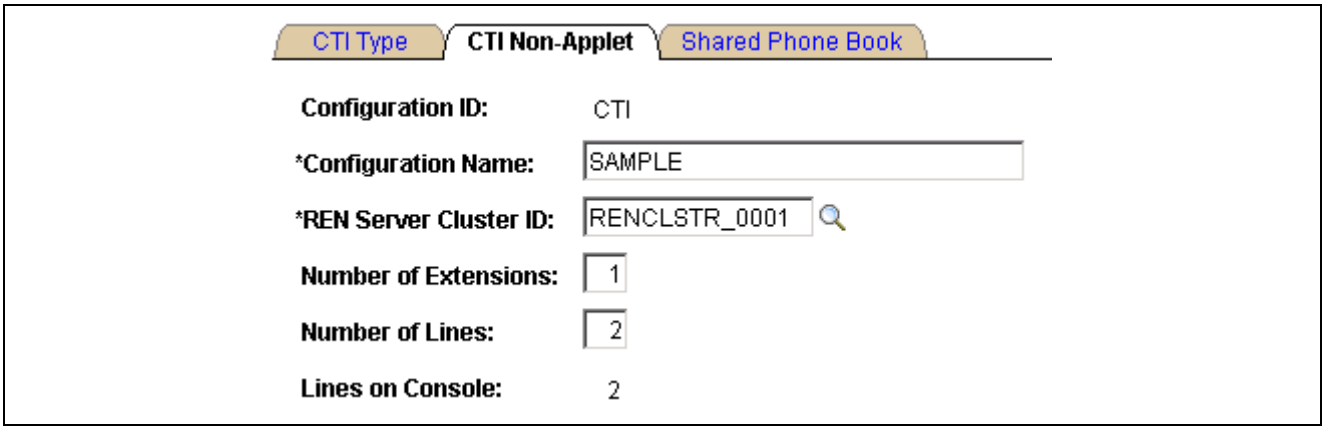
- Any CTI server that implements PSMCAPI.
- Genesys T-Server or Configuration Server.
- Cisco ICM Server.

Note. The Configuration ID is created when CTI is configured for the first time.

To create a CTI configuration, use the CTI Type page, the CTI Non-Applet or CTI Applet page, the Shared Phone Book page. If you are using the CTI applet, also configure the CTI Cisco or CTI Genesys page, depending on your CTI middleware.

Configuring the CTI Non-Applet Console

Access the CTI Non-Applet page to configure the CTI non-applet (JSMCAPI) console.



CTI Non-Applet page

Configuration ID Displays the name that you gave the configuration when you created it. The name cannot be modified once created.

Configuration Name Add a descriptive name to help identify the configuration.

Number of Extensions Enter the number of extensions or directory numbers associated with the telephone.

For Genesys, *1* or *2* can be used.

For Cisco, only a value of *1* can be used.

Copyright © 1988-2007, Oracle. All rights reserved.

Number of Lines Enter the number of lines associated with each extension. Depending on your configuration, you can enter up to two lines.

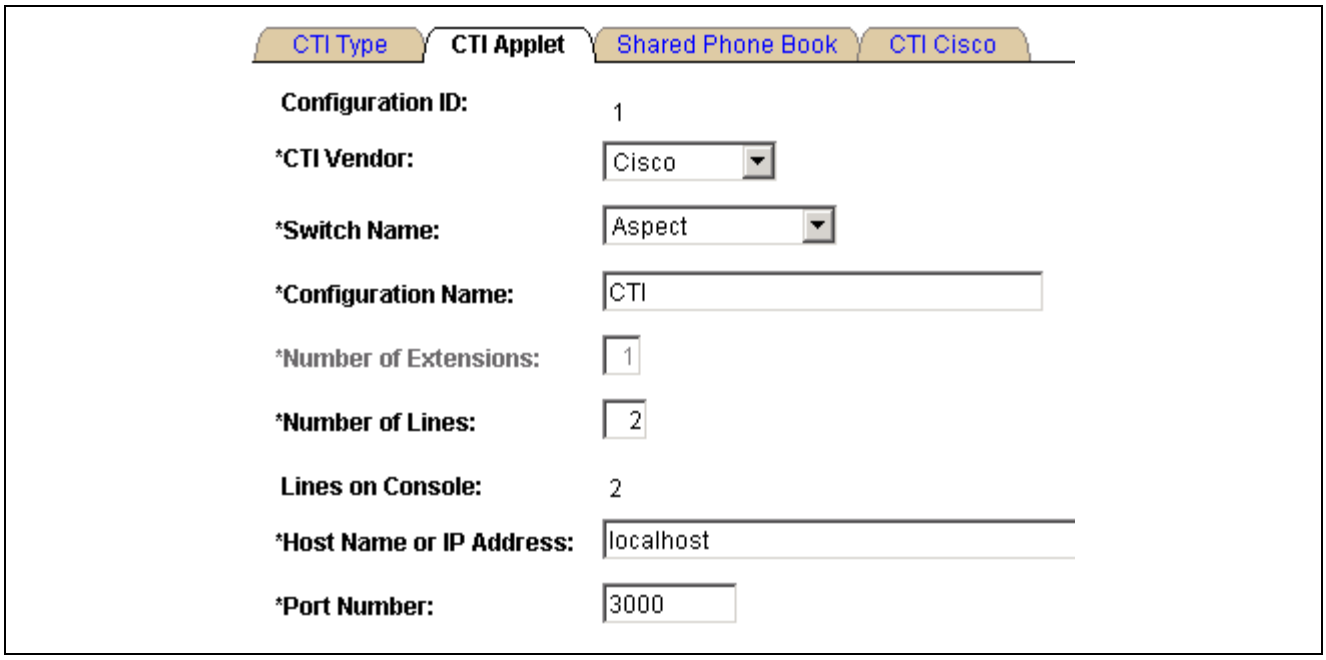
If Number of Extensions is *1,* this value should be *2.*

If Number of Extensions is *2,* this value should be *1.*

Lines on Console The CTI console supports up to two lines. The only supported configurations are two extensions with one line each, one extension with two lines, or one extension with one line.

Configuring the CTI Applet Console

Access the CTI Applet page to configure PeopleSoft CTI for use with the CTI applet console.



CTI Applet page

Configuration ID Displays the name that you gave the configuration when you created it. The name cannot be modified once created.

CTI Vendor Select the vendor of your CTI solution. The options are:

- *Genesys*
- *Cisco*

This value controls whether the CTI Cisco page or the CTI Genesys page appears in the component. For example, if you select *Cisco,* the CTI Cisco page appears.

Switch Name Select from one of the supported switches.

Currently, the supported switches for Genesys are:

- *Siemens Hicom 300*
- *Nortel Meridian*
- *Avaya Definity G3*

Copyright © 1988-2007, Oracle. All rights reserved.

- *Aspect*

Consult the Genesys Framework T-Server Reference Manuals to check if features offered by the CTI Console are supported on your switch. For example, mute transfers are not supported on Aspect switches.

Currently, the supported switches for Cisco are:

- *IPCC*
- *Avaya Definity G3*
- *Aspect*

Consult the Cisco documentation to determine if features offered by the CTI Console are supported on your switch. For example, DND and mute transfers may not be supported for all switches.

Configuration Name Add a descriptive name to help identify the configuration.

Number of Extensions Enter the number of extensions or directory numbers associated with the telephone.

For Genesys, *1* or *2* can be used.

For Cisco, only a value of *1* can be used.

Number of Lines Enter the number of lines associated with each extension. Depending on your configuration, you can specify up to two lines.

If Number of Extensions is *1,* this value should be *2*.

If Number of Extensions is *2,* this value should be *1*.

Lines on Console The CTI console supports up to two lines. The only supported configurations are two extensions with one line each, one extension with two lines, or one extension with one line.

Host Name or IP Address Enter the host name or IP address for one of the following:

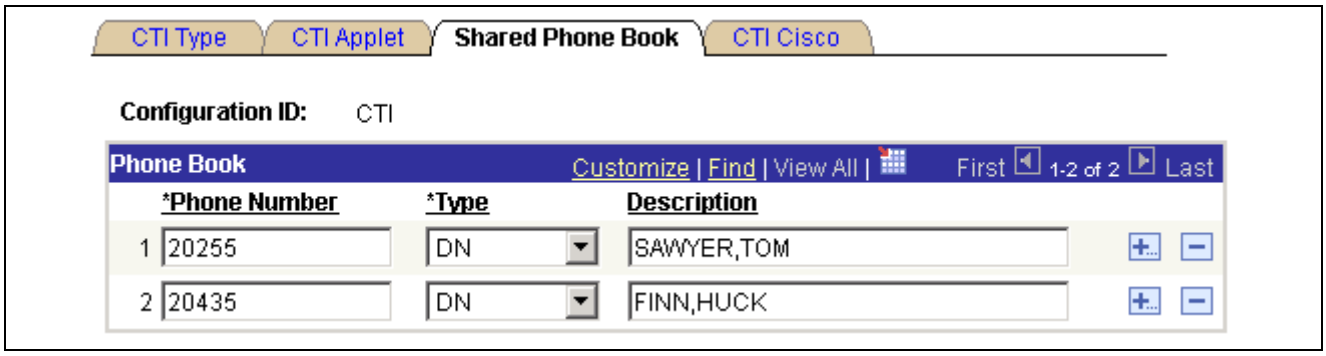
- Genesys T-Server or Configuration Server.
- Cisco ICM Server.

Port Number Enter the port number on which the T-Server or Configuration Server or Cisco ICM listens.

Using the Shared Phone Book Page

Access the Shared Phone Book page.

Copyright © 1988-2007, Oracle. All rights reserved.



Shared Phone Book page

On this page, you can manage a list of frequently dialed phone numbers for a specific CTI configuration. These numbers appear when an agent connected to that CTI configuration selects the drop-down list box when dialing a number from the CTI console. This saves the agents from manually entering any dialed numbers present in that CTI configuration when making outbound calls.

Note. Phone lists are updated on the CTI console only after the CTI console launches. To refresh phone lists, refresh the browser and reactivate the console.

Phone Number Enter a frequently dialed phone number to associate with this configuration.

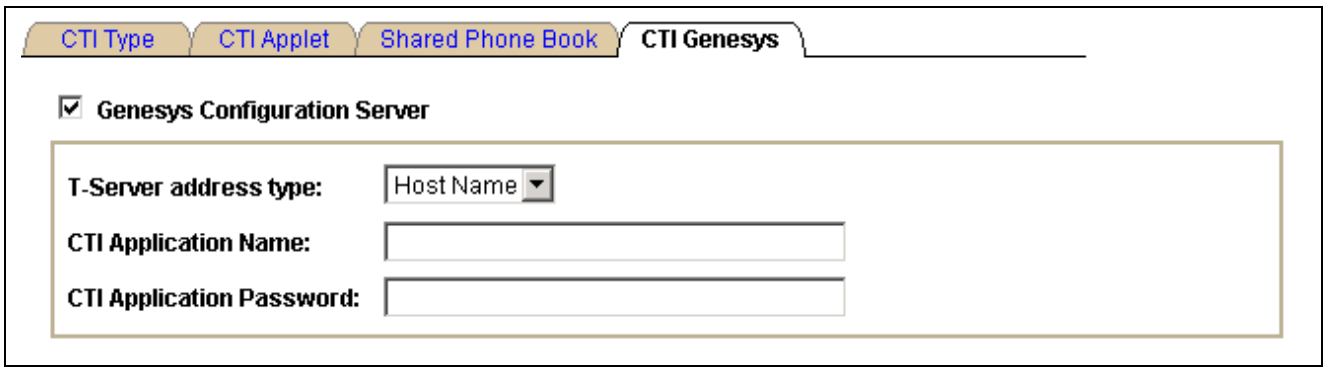
Type Select either:

- DN (directory number): This number identifies a telephone set on a PBX or in the public network. The caller dials this number to establish a connection to the addressed party. The DN can be a local PBX extension (a local DN) or a public network telephone number
- Queue: Directory number that identifies an ACD queue or group. Calls to a group are distributed to agents belonging to the group, according to ACD algorithms.

Description Add a description for the telephone number.

Using the CTI Genesys Page

Access the CTI Genesys page.



CTI Genesys page

The CTI Genesys page provides additional options for Genesys implementations. This page only appears when Genesys is selected as the CTI Vendor on the General Configuration page and the chosen CTI type is Applet.

Copyright © 1988-2007, Oracle. All rights reserved.

Genesys Configuration Server

Select to direct the CTI console to get data required for connecting to the T-Server from a configuration server instead of from the PeopleSoft database. Using a configuration server is transparent to the user or agent. When the agent activates the console, it requests a list of available T-Servers from the Configuration Server, and then the console sequentially attempts to connect to each T-Server in that list until it establishes a connection. If it reaches the end of the list without connecting to a T-Server, an error is returned to the agent.

Note. If you select this option, you enable the Application User Information group box on this page. Enter the appropriate application user name and password for the Genesys system.

T-Server address type

If you are using a configuration server to indicate what T-Server to connect to (instead of connecting directly to a T-Server), indicate whether the configuration server is returning the host name or the IP address of the T-server.

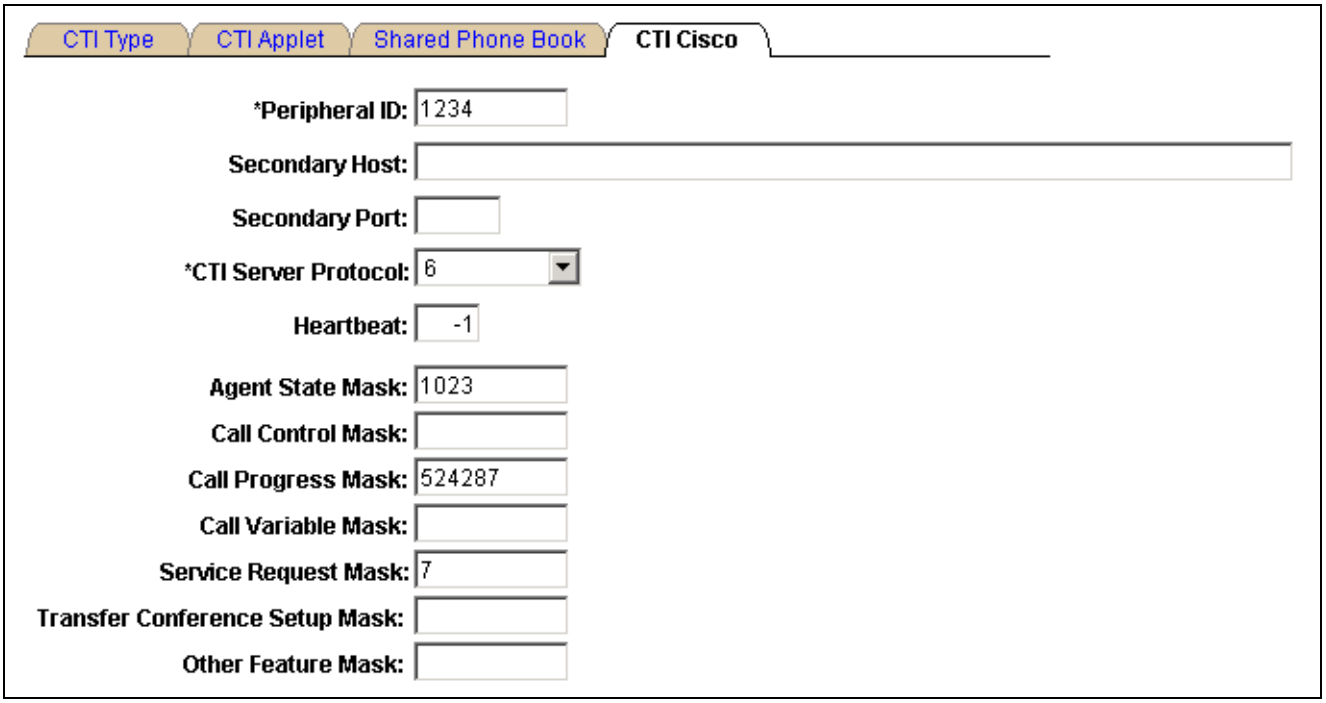
Enter the type of address that should be used when sending information to the T-Server. The options are *Host Name* and *IP Address*. Your selection should correspond to the host name or IP address that you entered on the CTI Applet page.

CTI Application Name and **CTI Application Password**

Enter the Genesys application name and password used to sign on to the Genesys Configuration Server. Using this option, each user connects to the configuration server with the same application name.

Using the CTI Cisco Page

Access the CTI Cisco page.



CTI Cisco page

The CTI Cisco page provides additional options for Cisco implementations. This page appears only when Cisco is selected as the CTI Vendor on the General Configuration page and the chosen CTI type is Applet.

Copyright © 1988-2007, Oracle. All rights reserved.

Peripheral ID Enter the switch (ACD) that the current configuration uses. The peripheral ID is the alias used to identify a switch within the system.

Secondary Host and **Secondary Port** Enter a second ICM server. In most cases, the secondary server is used for fail over and load balancing.

CTI Server Protocol Enables you to set the server protocol for a configuration. Currently, only protocol 6 is supported.

Heartbeat Enter an interval in seconds for the system to send a message to the CTI Server to make sure it is up and running. To disable the feature, enter −1. To enable it, enter a value in seconds. The minimum value is 5 seconds. Refer to the Cisco documentation for any recommendations for this option.

Agent State Mask Enter a combination of agent state masks that the CTI agent wants to receive. For instance, you may set the AGENT_AVAILABLE_MASK, when you want the application to receive available AGENT_STATE_EVENT messages. PeopleSoft supports the following masks:

- AGENT_LOGIN_MASK
- AGENT_LOGOUT_MASK
- AGENT_NOT_READY_MASK
- AGENT_AVAILABLE_MASK
- AGENT_TALKING_MASK
- AGENT_WORK_NOT_READY_MASK
- AGENT_WORK_READY_MASK
- AGENT_BUSY_OTHER_MASK
- AGENT_HOLD_MASK

Note. This value may be overridden by a Cisco system-level mask.

Refer to your Cisco documentation for more information.

Call Control Mask The PeopleSoft system does not currently use this mask in the CTI application. This option is reserved for future use.

Refer to your Cisco documentation for information about this mask.

Note. This value may be overridden by a Cisco system-level mask.

Call Progress Mask Displays any unsolicited call event messages that you want your application to receive. For instance, you can enter the events that a particular application depends on, such as Begin_Call, CALL_DELIVERED_EVENT, and so on. You can opt to skip events that the application does not depend on, and this can reduce network traffic.

The PeopleSoft system uses the following masks:

- BEGIN_CALL_MASK
- END_CALL_MASK
- CALL_DATA_UPDATE_MASK

 Copyright © 1988-2007, Oracle. All rights reserved.

- CALL_FAILED_MASK
- CALL_DELIVERED_MASK
- CALL_ESTABLISHED_MASK
- CALL_HELD_MASK
- CALL_RETRIEVED_MASK
- CALL_CLEARED_MASK
- CALL_CONNECTION_CLEARED_MASK
- CALL_ORIGINATED_MASK
- CALL_CONFERENCED_MASK
- CALL_TRANSFERRED_MASK
- CALL_DIVERTED_MASK
- CALL_SERVICE_INITIATED_MASK

Note. Modify this value with caution. If you mistakenly elect to skip an event on which an application depends, the application can fail.

This value may be overridden by a Cisco system-level mask.

Call Variable Mask

This option is intended for future use. It is not currently implemented for use with PeopleSoft applications.

Refer to your Cisco documentation for information on this mask.

Note. This value may be overridden by a Cisco system-level mask.

Service Request Mask

Enables you to adjust CTI service masks. The PeopleSoft system uses the following masks:

- CTI_SERVICE_CLIENT_EVENTS
- CTI_SERVICE_CALL_DATA_UPDATE
- CTI_SERVICE_CLIENT_CONTROL

If you modify this option, do not enter a value lower than 7.

Refer to your Cisco documentation for more information on this option.

Note. This value may be overridden by a Cisco system-level mask.

Transfer Conference Setup Mask

Enter the valid ways the application may be configured for a transfer or conference call.

Refer to your Cisco documentation for more information on this option.

Other Feature Mask

Enables you to select the other features supported by an application.

Refer to your Cisco documentation for more information about this option.

Two additional parameters are available to configure the Cisco expanded call context variable. Configure these expanded call context variables in the Cisco middleware:

Copyright © 1988-2007, Oracle. All rights reserved.

- user.PS.referenceID: CTI internal use only.

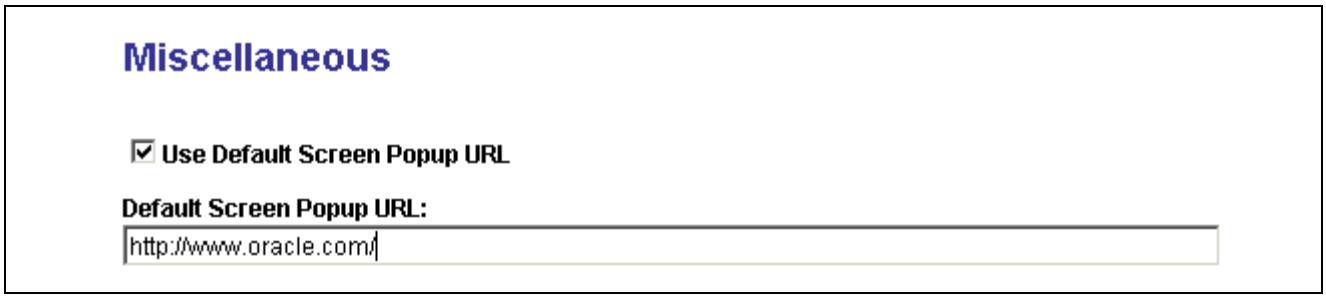
  The console uses this parameter to identify a call.
- user.PS.OutboundContext: This variable, a string type, is attached to an outbound call.

  This variable can be used to associate an outbound call with the desired context.

Using the Miscellaneous Page

Access the Miscellaneous page.

Note. This page is available using the CTI server only.



Miscellaneous page

Set default screen pop-up URL parameters on the Miscellaneous page.

Use Default Screen Popup URL Select this option if you want to set a default URL for the pop-up screen.

The system uses the default method to determine what URL to use for the page launched for incoming calls. The system uses that default URL to examine the user data attached to that call. However, for customers that do not want to attach this user data to incoming calls, this option enables you to use the same URL for all pop-up screens.

After selecting this option, enter the URL in the Default Screen Popup URL edit box.

Default Screen Popup URL If the Use Default Screen Popup URL is selected, enter the value for the default URL.

Note. To ensure that the user does not have to sign on to the pop-up window, the domain of the default URL must exactly match the domain of the signon URL. If the signon URL has no domain, the machine name of the default URL should be the same as the machine name of the signon URL. If either the domain or the machine name do not match, the system prompts the user for the user ID and password for the first pop-up screen.

With Matching Domains:

Within the same domain, such as example.com, the machine names can be different. For example:

Signon URL:

http://ntserver1.example.com/peoplesoft8/signon.html

Default URL:

Copyright © 1988-2007, Oracle. All rights reserved.

http://uxserver2.example.com/servlets/iclientservlet/peoplesoft8/?ICType=PANEL&Menu=UTILITIES&Market=GBL&Component=MESSAGE_CATALOG1&Target=Main1&LANGUAGE_CD=ENG&MESSAGE_SET_NBR=1

Without Matching Domains:

Without the matching domains, the machine name should be the same in both URLs. For example:

Signon URL:

http://ntserver1/peoplesoft8/signon.html

Default URL:

http://ntserver1/servlets/iclientservlet/peoplesoft8/?ICType=PANEL&Menu=UTILITIES&Market=GBL&Component=MESSAGE_CATALOG1&Target=Main1&LANGUAGE_CD=ENG&MESSAGE_SET_NBR=1

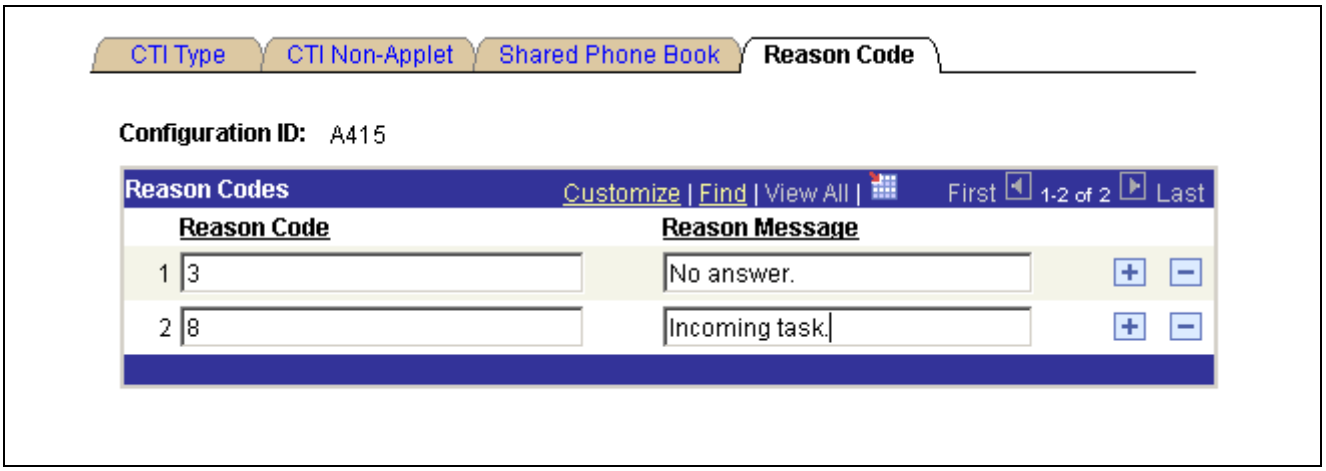
See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, "Implementing Single Signon"

Using the Reason Code Page

Access the Reason Code page.

Note. This page is available only to third-party vendors.



Reason Code page

This page is used by the third-party vendors to define their customized codes.

See Also

Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Understanding JSMCAPI, page 155

Copyright © 1988-2007, Oracle. All rights reserved.

Configuring CTI Queues and CTI Agents

To configure CTI queues and agents, use the Queue Configuration (PT_CTI_QUEUE) and CTI Agent Configuration (PT_CTI_AGENT) components.

This section discusses how to:

- Configure CTI queues.
- Configure CTI agents.
- Use the Phone Book page.
- Use the Personalization page.
- View information on the Agent Info page.

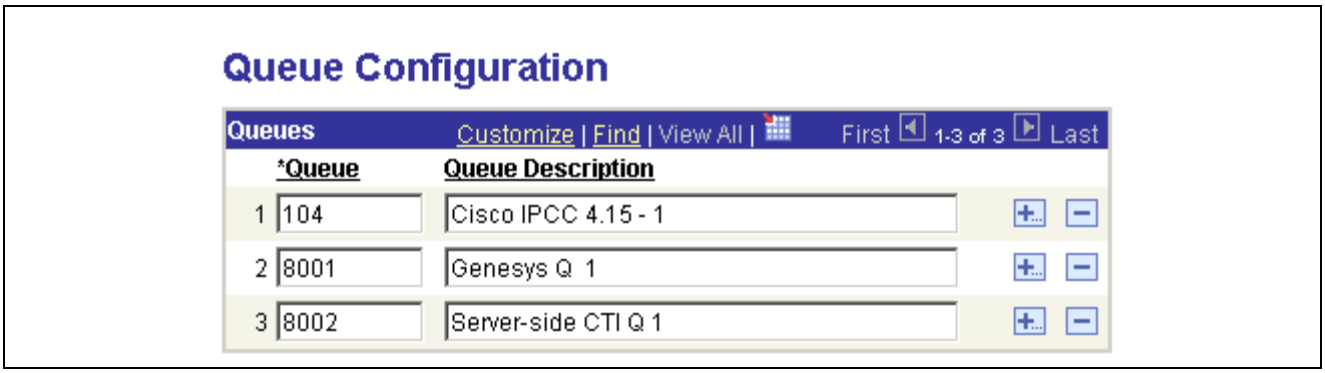
Pages Used to Configure CTI Queues and CTI Agents

Page Name	Object Name	Navigation	Usage
Queue Configuration	PT_CTI_QUEUE	PeopleTools, MultiChannel Framework, CTI Configuration, Queue	Configure CTI queues.
CTI Agent Configuration	PT_CTI_AGENT	PeopleTools, MultiChannel Framework, CTI Configuration, Agent, CTI Agent Configuration	Configure CTI agents.
Phone Book	PT_CTI_PNAGENT	PeopleTools, MultiChannel Framework, CTI Configuration, Agent, Phone Book	Use an agent phone book.
Personalization	PT_CTI_AGENTGUI	PeopleTools, MultiChannel Framework, CTI Configuration, Agent, Personalization	Personalize agent console and pop-up windows.
Agent Information	PT_CTI_AGENTINFO	PeopleTools, MultiChannel Framework, CTI Configuration, Agent Information	View information about the current agent.

Configuring CTI Queues

Access the Queue Configuration page.

Copyright © 1988-2007, Oracle. All rights reserved.



Queue Configuration page

Use this page to add queues for agents.

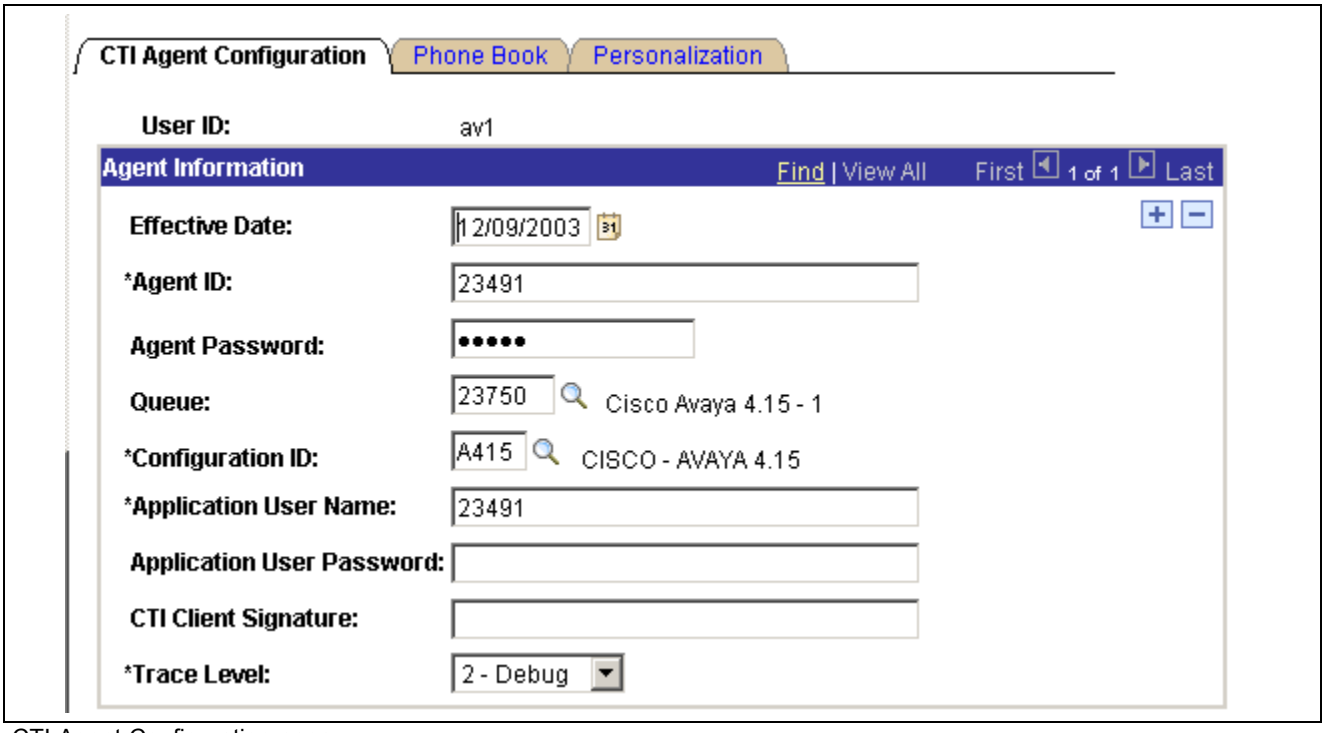
Queue Enter the directory number identifying an ACD group. Calls to a group are distributed to ACD agents belonging to that group, according to ACD algorithms.

Note. Queues names can be alphanumeric.

Queue Description Enter a brief description of the queue.

Configuring CTI Agents

Access the CTI Agent Configuration page.



CTI Agent Configuration page

Configuring CTI agents involves the CTI Agent Configuration page, the Phone Book page, and the Personalization page.

Copyright © 1988-2007, Oracle. All rights reserved.

User ID Displays the PeopleSoft User ID of the agent.

Agent Information

Effective Date Enter the date on which the current configuration should become active.

Agent ID Enter a user ID for the agent within the switch.

Agent Password Enter the password the agent uses to sign on to the phone, if any.

Queue Enter the name of the queue you want to assign to an agent. Use the Queue Configuration page to associate a queue with a directory number identifying an ACD group.

Configuration ID Select the name of the configuration that you want to associate with the agent. The configuration ID is the name of the configuration you created using the CTI Applet or CTI Non-Applet page.

Application User Name/Password Enter the CTI user name and password of the agent.

CTI Client Signature

Note. This option only appears for Cisco configurations.

This control shows the signature of a particular agent. The signature uniquely identifies an agent if you have implemented call monitoring. Typically, this value appears as an email address, such as john.doe@example.com.

Trace Level For Trace Level, you have the following options:

- 0 - None: Disables tracing.
- 1 - Info (informational): Traces agent actions, such as dialing out, transfers, and so on.
- 2 - Debug: Used to troubleshoot crashes and other major errors.

  If you ever need to open a PeopleSoft GSC case about a CTI issue, include a level 2 - Debug trace.

If you are using the JSMCAPI (non-applet) console, two tracers (a JSMCAPI tracer and a console tracer) appear if trace level 2, Debug, is selected.

If you are using the Java applet console, trace information is written to the browser's Java Console, which must be enabled. In Microsoft Internet Explorer, enable the Java Console by selecting Tools, Internet Options, Advanced, Java, Java Console. View the Java Console by selecting View, Java Console. To clear the console, press C on your keyboard.

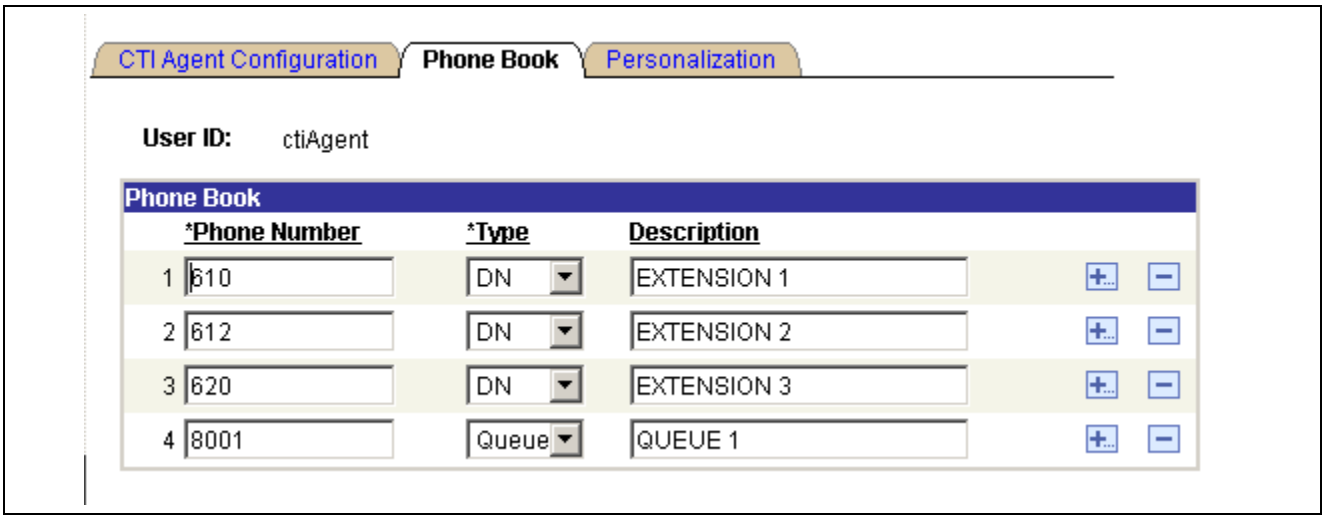
In Sun Java, enable Java Console by selecting Tools, Sun Java.

Note. Setting the trace level to *2 - Debug* can degrade performance. Unless you are troubleshooting the system, set the trace level to *None*..

Using the Phone Book Page

Access the Phone Book page.

Copyright © 1988-2007, Oracle. All rights reserved.



Phone Book page

On this page, you can manage a list of frequently dialed telephone numbers for a *specific* CTI agent. These numbers appear when that agent selects the drop-down list box for a number to dial in the CTI console. This listing saves the agents from having to manually enter frequently dialed numbers when making outbound calls.

Note. Phone lists are updated on the CTI console only after the CTI console launches. To refresh phone lists, refresh the browser and reactivate the console.

Phone Number Enter frequently dialed telephone numbers associated with a particular agent.

Type Select either:

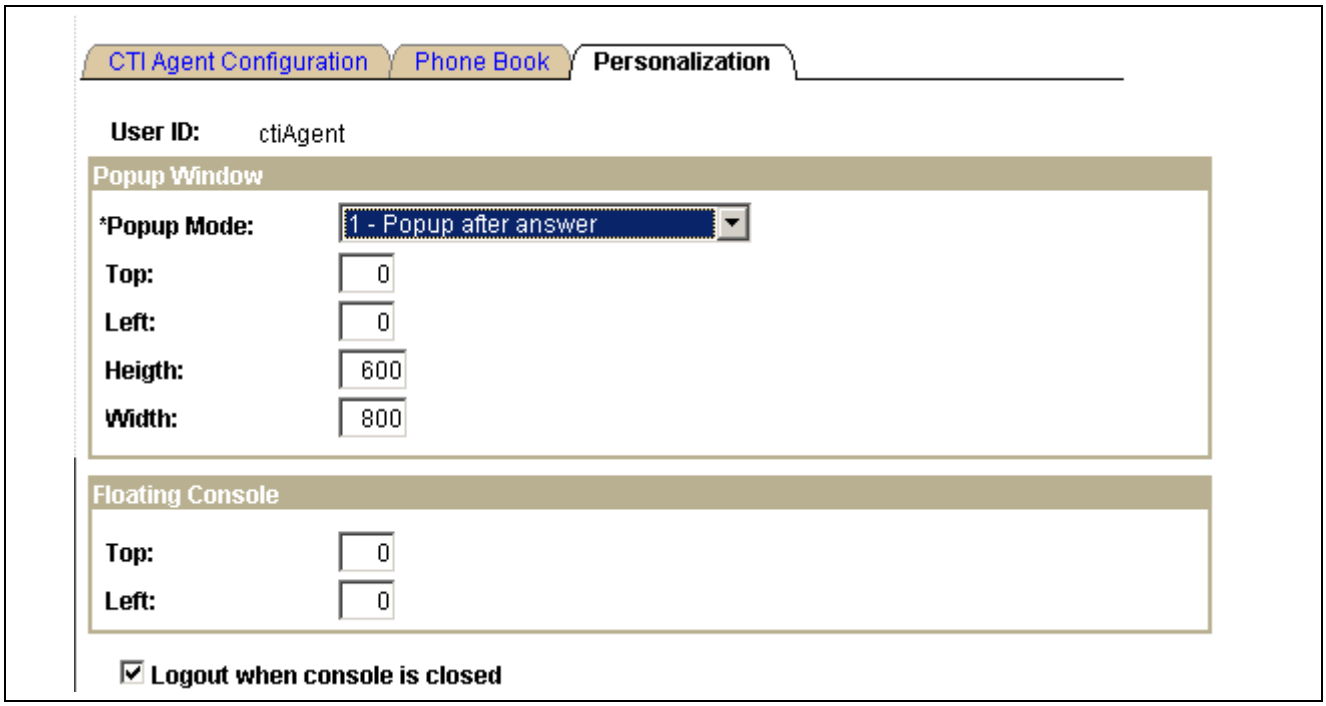
- DN (directory number): This number identifies a telephone set on a PBX or in the public network. The caller dials this number to establish a connection to the addressed party. The DN can be a local PBX extension (a local DN) or a public network telephone number.
- Queue: Directory number identifying an ACD queue or group. Calls to a group are distributed to agents belonging to the group, according to ACD algorithms.

Description Add a description for the telephone number.

Personalizing Agent Console

Access the Personalization page.

Copyright © 1988-2007, Oracle. All rights reserved.



Personalization page

Use this page to personalize the pop-up windows' timing, size, and position on the desktop as well as the position of the floating console.

Popup Window

Popup Mode

Enables you to configure when the pop-up window appears. You can have it appear after the call is answered, or you can have it appear when a call comes in. If it appears before the call is answered, the agent can determine if she or he wants to answer the call based on the information that appears in the pop-up window.

The default pop-up mode is: *1- Popup after answer.*

In some cases, for example accepting a transfer, although you set the value as *0- Popup when incoming,* you still get the pop-up screen after you answer. This is because no call data was attached by the CTI vendor to the incoming event, and consequently the PeopleSoft CTI console couldn't build the URL for the pop-up screen.

For example, assume party A is transferring a call to party B. Party B gets the incoming event, but party B doesn't get a screen pop-up window. This is because the console did not receive the user data to generate the URL despite the fact that the mode is set to *0.* But, after B answers this call, A completes the transfer. Then, B gets the event *partychanged*, receives the user data, and generates the URL and pop-up window.

Top

Enter the top position, in pixels. This value is relative to the top of the screen.

Left

Enter the left position, in pixels. This value is relative to the left side of the screen.

Height

Enter the height of the window, in pixels. The minimum value is 100.

Copyright © 1988-2007, Oracle. All rights reserved.

Width Enter the width of the window, in pixels. The minimum value is 100.

Logout when console is closed Select to automatically log the agent out when the CTI console is closed. Only applies on a Genesys switch system; Cisco always logs out the CTI agent when the CTI console is closed.

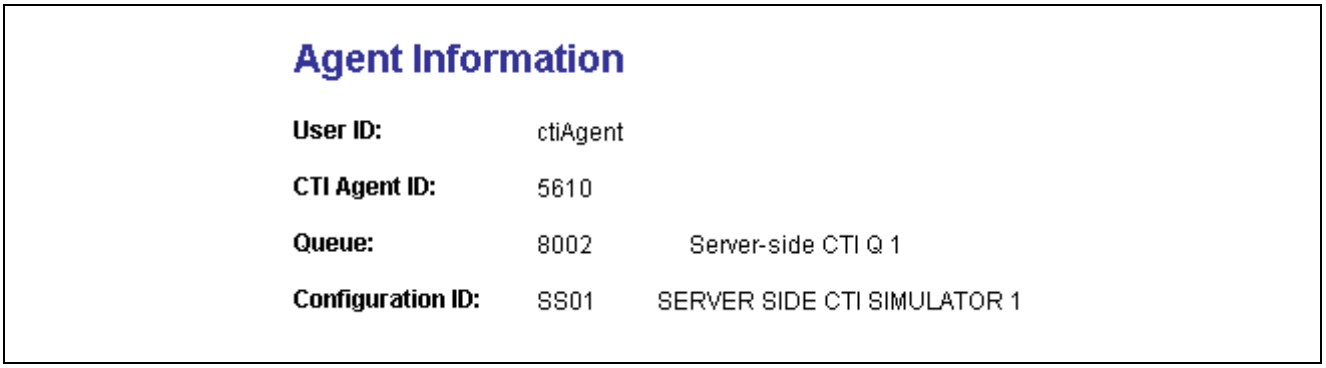
Floating Console

Top Enter the top position in pixels. This value is relative to the top of the screen.

Left Enter the left position in pixels. This value is relative to the left side of the screen.

Viewing Information on the Agent Information Page

Access the Agent Information page.



Agent Information page

The Agent Information page is a read-only page that displays the following information about a CTI agent.

User ID Displays the agent's PeopleSoft User ID.

Agent ID Displays the agent's CTI middleware ID.

Queue Displays the queue to which an agent is assigned.

Configuration ID Displays the configuration ID associated with the agent.

Using Other CTI Options

You can configure other optional CTI parameters.

This section discusses how to:

- Configure pop-up windows.
- Support single signon.
- Log CTI events.
- Implement "free seating."

Copyright © 1988-2007, Oracle. All rights reserved.

Configuring Pop-Up Windows

PeopleSoft CTI can launch and populate transaction pages in the following ways:

- Default URL.

  The same URL is used for all calls. The Automatic Number Identification (ANI) is passed in as a parameter to that URL. The ANI identifies the telephone number from which the incoming call originated and it may be useful in determining what application data to retrieve. For example, it may be the home phone number of a customer.

- Build the URL from attached Call Data.

  PeopleSoft CTI formats a URL for the browser with a specific target PeopleSoft menu, market, and component. However, this method cannot be used if you are accessing multiple databases through the PeopleSoft Portal.

- iScripts.

  PeopleSoft CTI opens the transaction page using an iScript. This method must be used if you are accessing multiple databases through the PeopleSoft Portal. The iScript communicates with the targeted database to populate the appropriate transaction page with the caller's data.

Note. The call ID is passed in the URL string as a variable named *callID*.

Note. Browser settings and add-on tools that block pop-up windows can prevent CTI inbound call pop-up windows from appearing. If possible, configure such tools to allow pop-up windows from your site or domain.

Setting Up Genesys for Pop-Up Windows

Unless you choose to use the default URL for your pop-up windows, you must create certain user-defined variables that are attached to each incoming call. These variables are then sent to the PeopleSoft CTI application providing instructions on which PeopleSoft Pure Internet Architecture page displays for the agent. This data is typically created and attached to the call using the Genesys Strategy Builder.

Note. The following variables are case sensitive. Match the case of the variables as shown in this section.

ICType

Represents the type of PeopleSoft service being called, either a panel (page) or a script.

If the ICTYPE is set to Panel, then the following three attached data variables must be set:

- Menu: The name of the PeopleSoft menu containing the destination component.
- Market: The market property of the target component.
- Component: The target component name in the PeopleSoft Application.

If the ICTYPE is set to Script, then the following attached data variable must be set:

ICScriptProgramName. This represents the location of the iScript, which is executed through the pop-up screen.

Descr (description)

Optional data for descriptive purposes only. You can add any descriptive information that might be useful to the agent. This information appears in

Copyright © 1988-2007, Oracle. All rights reserved.

the CTI console. For example, you may want the agent to be aware of the customer's priority status, as in "Gold Customer."

Application Specific Data Any other attached data keys that are sent to the CTI console by the Genesys telephony server, such as customer number, are passed to the target application page as parameters.

The parameters are separated from the PeopleSoft URL by a question mark (?). Parameters are separated from other parameters by an ampersand (&). If you are using iScripts, you can use the GetParameter methods to read the parameters in your PeopleCode. If you choose to use ICPanels instead, the parameters are used as the key list.

Refer to your PeopleSoft application documentation for specific instructions on any other attached data keys that must be passed to the CTI console by Genesys.

Setting up Cisco for Pop-Up Windows

Unless you choose to use the default URL for your pop-ups windows, you must create certain expanded call context (ECC) variables. Expanded call context (ECC) variables are variables that you define and enable in the Cisco ICM Configuration Manager to store values associated with the call. These variables are sent to the PeopleSoft CTI console when ICM notifies it of an incoming call, and the console uses the variables to determine which PeopleSoft page to launch for the agent.

Every call data key used in your implementation must be registered in the Cisco ICM, before the PeopleSoft system can receive the attached call data from the IVR.

The PeopleSoft system only recognizes data keys with *user.PS.* in front of all the PeopleSoft key names. For example,

```
user.PS.Descr
```

The following table contains the call data variables that must be created and registered in the ICM.

Note. The following variables are case sensitive. Match the case of the variables as shown in this section.

ICType user.PS.ICType. 6 chars. The valid values are: Panel (page) or Script. It represents the type of PeopleSoft service being called, either a panel (page) or a script.

If ICType is set to panel, the following variables apply:

- user.PS.Component. 20 chars. This variable is required to specify the name of the destination component.
- user.PS.Menu. 10 chars. This variable is required to specify the name of the menu within PeopleSoft containing the destination component.
- user.PS.Market. 3 chars. This variable is required to specify the name of the Market of the destination component.

If the ICType is set to Script, then the following variable applies:

user.PS.ICScriptProgramName. 70 chars. If the ICTYPE is set to Script, then this variable is required to specify the location of the iScript, which is executed through the pop-up screen.

Descr user.PS.Descr. 30 chars.

Copyright © 1988-2007, Oracle. All rights reserved.

	You can add any descriptive information that might be useful to the agent. This information displays in the control bar. For example, you may want the agent to be aware of the customer's priority status, as in "Gold Customer."
Application Specific Data	Any other variables that the application requires must be prefixed with *user.PS.* PeopleSoft truncates the *user.PS.* before passing the parameter to the application. For example, user.PS.App1(3 chars) user.PS.App2(1 chars)

Note. Typical character sizes for these variables appear in the previous table. The actual size of these variables depends on the components specified by the PeopleSoft application you are using. Refer to your PeopleSoft application documentation for correct size information.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference, "Internet Script Classes (iScript)"

Supporting Single Signon

PeopleSoft CTI offers single signon. The console connects to the CTI middleware using the CTI user ID and password retrieved from the PeopleSoft database.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, "Implementing Single Signon"

CTI Event Logging

CTI events can be logged in the CTI Event Log. The logged CTI events can be viewed by navigating to PeopleTools, MultiChannel Framework, Universal Queue, Administration, CTI Event Log.

To implement CTI event logging, navigate to PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Cluster Tuning, and set the key *log_cti* to *yes.* Use the Notify Cluster page to notify the affected queue to refresh logging parameters.

The CTI event log logs all events and requests related to a session, user, or call connection. It also logs all events related to call and group information.

Note. Applet-based CTI events are not logged.

See Also

Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Tuning Cluster Parameters, page 93

Implementing Free Seating

When users sign on, they do not need to reenter telephone extensions and other user information if they have used the workstation before and the relevant information for the telephone associated with that workstation has not changed.

The PeopleSoft system enables free seating by maintaining a cookie on the workstation.

 Copyright © 1988-2007, Oracle. All rights reserved.

Using the CTI Sample Pages

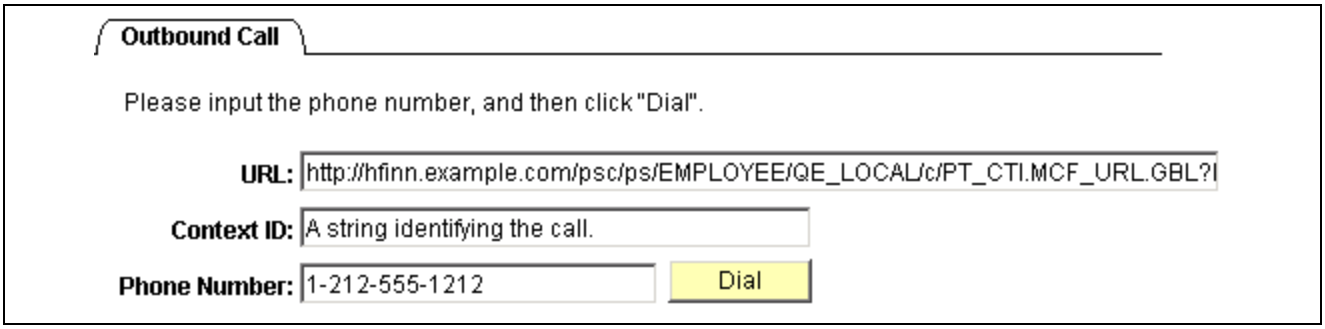
To demonstrate an outbound call, use the Sample Pages component (PT_CTI_DEMOOUTB). The CTI sample pages are intended for demonstration purposes only and should not be used in production.

Page Used to Demonstrate Outbound Calls

Page Name	Object Name	Navigation	Usage
Outbound Call	PT_CTI_DEMOOUTBOUN	PeopleTools, MultiChannel Framework, CTI Configuration, Sample Pages, Outbound Call	Demonstrate making an outbound call.

Using the Outbound Call Page

Access the Outbound Call page.



Outbound Call page

The Outbound Call page is an example of how you can customize an application page to enable a user to direct the CTI console to dial a telephone number displayed on that page. The outbound calling demonstration works only when the CTI console is enabled and the user has registered with the CTI vendor.

URL Enter the URL of a page to display when dialing out.

Context ID Enter a string to attach to the call as outbound context.

Phone Number Enter the telephone number you want to dial. This field accepts numeric digits only. Do not enter special characters, such as - (hyphen), . (period), or other separators.

Dial Click to dial the telephone number you entered.

Note. The CTI Outbound Call sample page may fail when used with the CTI applet console and a Netscape browser if you have specified an authentication token domain for PeopleSoft Pure Internet Architecture.

See Also

Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the CTI Sample Console, page 160

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 4

Using PeopleSoft CTI

This chapter provides an overview of PeopleSoft CTI and discusses how to use PeopleSoft CTI using a CTI server or an integrated third-party routing system.

Understanding PeopleSoft CTI

This section discusses:

- PeopleSoft CTI.
- Cisco switch considerations.

PeopleSoft CTI

PeopleSoft CTI is a browser-based call management system that helps call agents work more efficiently with customers. PeopleSoft CTI integrates your Genesys or Cisco CTI middleware and your PeopleSoft applications. It exchanges data between the CTI system and your PeopleSoft applications so that the system automatically fills PeopleSoft transaction pages with the appropriate customer information—the information related to the caller.

With PeopleSoft CTI, you can perform the following tasks:

- Operate two lines or two extensions.
- Answer incoming calls.
- Release calls.
- Put a caller on hold.
- Monitor call status.
- Access PeopleSoft applications.
- Transfer callers.
- Initiate conference calls.
- Place an outbound call.

Note. PeopleSoft CTI supports both Genesys and Cisco CTI systems, and any CTI system that implements the PeopleSoft MultiChannel Application Programming Interface (PSMCAPI). In the following sections, when the phrase *your CTI middleware* or *CTI vendor* appears, assume that it refers to either Genesys or Cisco products, or another product implementing PSMCAPI, as appropriate for your installation.

Because a CTI console developed with the JavaScript MultiChannel Application Programming Interface (JSMCAPI) can be customized, features described in this chapter may not appear or may behave differently.

Copyright © 1988-2007, Oracle. All rights reserved.

The CTI Interface

The PeopleSoft CTI interface is incorporated *within* the MultiChannel Console.

See Also

Chapter 3, "Configuring PeopleSoft CTI," page 13

Chapter 10, "Managing Tasks and Using Chat in PeopleSoft MultiChannel Framework," page 129

Cisco Switch Considerations

The following list presents the limitations of the Cisco Systems IP Contact Center switch:

- MAKE_CALL is only supported when the agent is in the Not Ready state.
- Consultative and blind transfers are supported.

  Placing a call on hold and making a new call and completing the transfer is not supported due to limitations in the Call Manager.
- Hold and retrieve are supported, but not during a consultative call as it disrupts the consult relationship in Call Manager.
- Only the conference initiator can add parties to the conference.
- Cisco does not support two applications managing the same call.

  Therefore, you cannot run a Cisco softphone and the PeopleSoft CTI console on the same workstation.

Note. Refer to your CTI vendor documentation for further information about any limitations imposed by a particular switch. In some cases, the switch limitations may actually be due to the CTI server, so be sure to refer to any server limitations that may also impose limitations on the switch.

Using PeopleSoft CTI

This section discusses how to:

- Get started.
- Use the CTI console.
- Select call actions.
- Answer a call.
- Transfer a caller.
- Initiate conference calls.
- Work with the hold status.
- Disconnect a caller.
- Switch agent ready status.
- Dial an outbound call.
- Complete a call.
- Use hot keys.

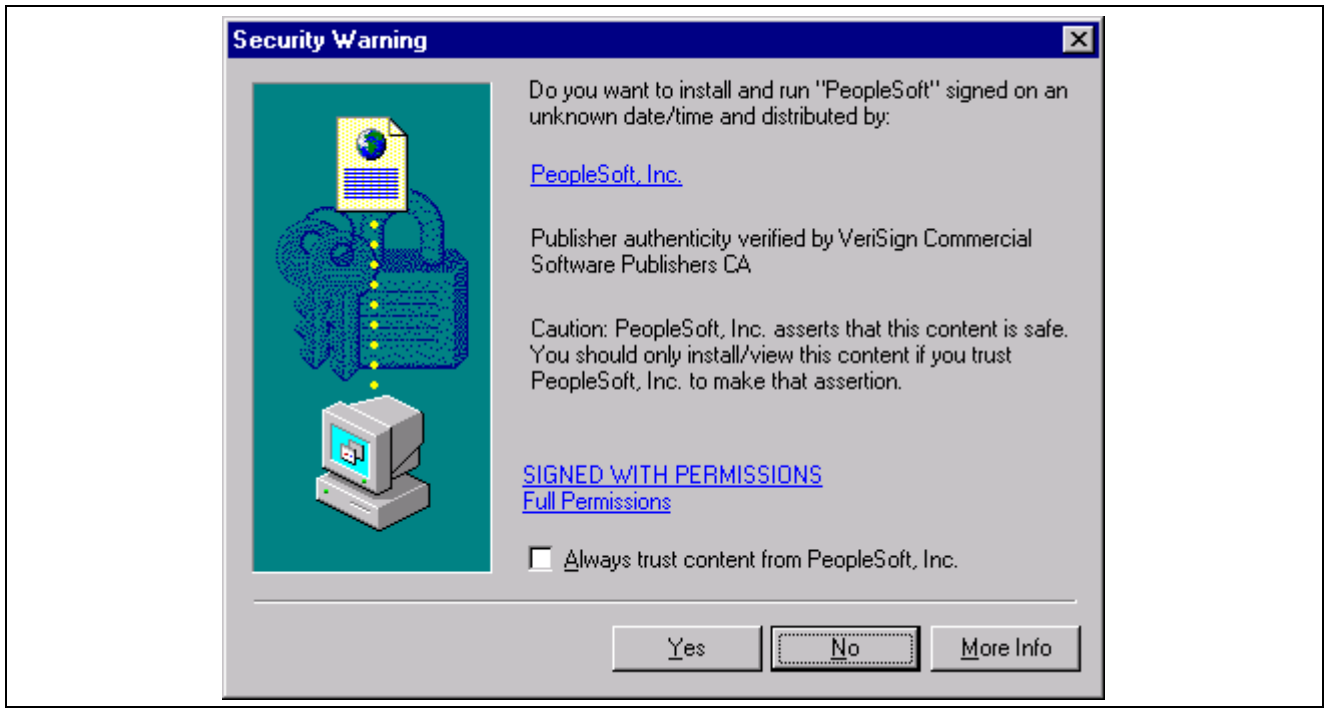
 Copyright © 1988-2007, Oracle. All rights reserved.

Getting Started

Getting started with PeopleSoft CTI is a three-step process that involves:

- Signing in to the PeopleSoft system.
- Specifying your extension information.
- Connecting to your CTI middleware (either Genesys or Cisco).

The first time that you access PeopleSoft CTI, a Security Warning dialog box *may* appear prompting you to trust information from PeopleSoft. Click the Yes button.



Security Warning dialog box

The Security Warning dialog box may not appear if:

- You have previously accepted PeopleSoft applets as trusted.
- The system is not correctly installed.

Note. If you have any questions or concerns about this warning, contact your system administrator.

To sign in to PeopleSoft CTI:

1. On the PeopleSoft sign-in page, enter your PeopleSoft user ID and password as you normally do to sign in to the PeopleSoft system.
2. Click the MultiChannel Console link in the universal navigation header.
3. On the CTI control bar, set your configuration by doing the following:
   - Make sure that your agent ID appears beneath the CTI Agent ID label.
   - Make sure that the appropriate queue name appears beneath the Queue label.

   Queues are discussed in a subsequent section.

Copyright © 1988-2007, Oracle. All rights reserved.

- If you are signing in from a workstation with a different extension, specify the current extensions in the Extension field.
- The Extension Type field indicates whether an agent can receive calls by way of a queue or just from a directory number (DN).

  If this value is set to *Queue,* the console offers options that are queue-specific, such as being able to log on to a queue.
- Click the Activate button to sign in to Genesys or Cisco.

  The CTI console appears.

Using the CTI Console

Change any registration parameters if necessary and click Set or Activate (depending on your version). After doing so, the CTI interface appears, which looks similar to the following:

Note. To launch server side CTI and MCF console using Netscape, go to Tools, Popup manager, Allow popups and add your URL.

Note. Keep in mind that the PeopleSoft CTI interface operates *within* the MultiChannel Framework Console. CTI users will only see the MultiChannel Console if their user ID belongs to a role that has MCF_AGENT or MCF_SUPR real-time event notification (REN) permissions, and WEBLIB_MCF web libraries in their role's permission list.

CTI Console

Note. Do not open two consoles for the same user on a single machine. Doing so can result in an error.

(register)

Select to register and unregister with the CTI middleware. The button acts as a toggle switch. When the green check mark appears, you are registered; when the red *X* appears, you are not registered.

Lines

Select the option to the left of the telephone icon to activate the associated line. The icon you select determines the active line. All call actions you choose apply only to the active line. The color of the icon reveals the activity on the line. The colors are:

- Blue: Inactive.
- Red: Ringing.
- Green: On call.
- Yellow: On hold.

Select Call Action

Select actions related to calls, including *Answer, Hold,* and *Transfer.* Which call action options appear depends on your current status. For instance, the Hold call action is only valid while you are on the line with a caller. Call action options are discussed in detail in the following section.

Status

Displays the agent status, as in Ready or Not Ready (to receive calls). It can also show Ready/DND (do not disturb) if the agent is not using a Queue.

Copyright © 1988-2007, Oracle. All rights reserved.

Q (queue) If the agent belongs to an automatic call distributor (ACD) group, the group appears here.

Messages Informational messages appear in the right corner of the CTI Console. Examples of such messages are “Dialing,” “Connected,” and “Released.” These messages do not persist.

Note. Error messages appear in separate windows.

Call Duration The system tracks the amount of time spent on calls for each line.

Incoming Call Information Displays the string associated with the Descr call variable. For example, “Gold Customer.”

Note. A CTI agent on a Genesys switch system can select on the agent Personalization page whether to log out automatically when the console is closed. Cisco always logs out the CTI agent when the CTI console is closed.

Selecting Call Actions

The Select Call Action drop-down list box contains all of the options you have for handling calls. Depending on the status of the agent or the telephone line, certain selections from the drop-down list box are not available.

After you select a call action there are two buttons, Go and Cancel, that appear to the right of the Select Call Action drop-down list box. Go executes the selected call action, and Cancel stops the call action you executed.

The following list describes the call actions:

Dial When you are in Agent Ready or Agent Not Ready mode, you can call another party.

Answer Call Answers an incoming call. The Answer Call button appears and flashes only when an incoming call is waiting to be answered.

Transfer Mute Transfers the caller to the desired number without speaking to the intended recipient. A dialog box appears enabling you to enter the extension of the person to whom you want to transfer the caller.

Note. This action is not supported on Aspect switches.

Transfer Transfers the caller to the desired number. You will have the opportunity to speak with the recipient before transferring the caller. A dialog box appears enabling you to enter the extension of the person to whom you want to transfer the caller.

Conference Enables you to add one or more individuals to your call.

Hold Places the caller on hold.

Retrieve Takes the caller out of *Hold* status. Appears only when the caller is on hold.

Release Disconnects the caller. This option is only available as a selection once a call is answered.

Change When an agent status is unknown, the system sets the status to *Change* so that the user can set the status manually to match the status for the telephone.

Copyright © 1988-2007, Oracle. All rights reserved.

Note. You can transfer to or initiate a conference call with individuals who are not enabled to access PeopleSoft CTI. Their phone rings, but keep in mind that they do not get the pop-up window showing customer data.

The following call actions are available depending on whether you are using a DN or Queue configuration:

Queue Select from:

Log on: Enables the agent to log an extension on to a queue.

Ready: Indicates that the agent is ready to receive incoming calls. This option is available when the extension is associated with a queue and the status bar reads Agent Not Ready.

Not Ready: Stops incoming calls. This option is available when the extension is associated with a queue and the status bar reads Agent Ready.

Log off: Enables the agent to log an extension off a queue. When logged off, the agent is no longer participating in the queue.

DN This option is available when the extension is not associated with a queue.

Ready: Indicates that the agent is ready to receive incoming calls.

Answering a Call

After you have logged on to PeopleSoft CTI, you are ready to receive calls. For each incoming ACD call to your extension, the telephone extension icon turns red. After you have accepted a call the system does not send you more incoming calls until you have completed the current call.

To answer a call:

1. Select the radio button to the left of the telephone icon that has turned red.

   The *Answer* option is automatically selected as the current option in the drop-down list box when an incoming call arrives.

2. Click Go.

   The popup browser launches with the appropriate PeopleSoft transaction page displayed. The system determines which page to display based on caller information sent by the CTI middleware.

   After you have answered a call, you enter the not available status.

If an agent erroneously clicks Cancel instead of Go when accepting an incoming call, the agent can recover the call by:

1. Selecting a call option.
2. Selecting the line that is receiving the incoming call.

Note. If CTI screen pops are configured, the CTI console will attach a CTI mini console to the screen pop page. This mini console will take a few seconds to initialize before it can be used.

Transferring a Caller

Occasionally, you need to transfer callers to other agents. PeopleSoft CTI supports two types of transfers:

- Transfer Mute: This option enables you to transfer a call without speaking to the target agent before transferring the call.

 Copyright © 1988-2007, Oracle. All rights reserved.

- Transfer: This option is also known as a consultative transfer, which means that you consult with the target agent before transferring the call.

Note. You can always transfer or invite users to a conference call even if the called party is not CTI-enabled. The non-CTI-enabled users do not get pop-up windows but their phones still ring.

When initiating a transfer or conference call on the Cisco system and the contacted party does not answer, wait until the system notifies you of the unsuccessful connection before attempting another action. This wait can be 20 to 30 seconds.

Cisco ICM does not notify client applications about some call events during two-step transfers or conferences. As a result, the state of the PeopleSoft CTI Console may not stay synchronized with that of the teleset, especially if the teleset is used to initiate or complete these call actions. The CTI console should automatically resynchronize with the teleset when the call is completed.

To perform a transfer mute:

1. Select the appropriate telephone line.

   The selected line must be green.

2. From the Select Call Action drop-down list box, select *TransferMute*.

3. In the Phone No. edit box, select the number that you want to dial.

   The drop-down list box contains all of the numbers from the shared phone book and agent phone book. If the number that you want to dial does not appear, click Dial other number and manually enter the number.

4. Click Go.

   This connects the caller to the new agent and releases your line.

   The system prompts the recipient of the transfer that it is transferring a call from your extension. When the recipient accepts the transfer, the PeopleSoft page connected to the caller's case opens as it did for you when you first received the call.

To perform a transfer (consultative):

1. Select the appropriate telephone line.

   The selected line must be green.

2. From the Select Call Action drop-down list box, select *Transfer*.

3. In the Phone No. edit box, select the number that you want to dial.

   The drop-down list box contains all of the numbers from the shared phone book and agent phone book. If the number that you want to dial does not appear, click Dial other number and manually enter the number.

4. Click Go.

   When using one extension with two lines, the outbound call you make to the agent to whom you are transferring the incoming call gets initiated on your second line and the incoming call gets placed on hold. Once the outbound call is established you can consult with the recipient and place that call on hold. To complete the transfer, you need to go back to the first line and select *Complete* and click Go. This action will release the call on the second line and transfer the call on the first line to the recipient of the transfer.

Copyright © 1988-2007, Oracle. All rights reserved.

When using two extensions, each with one line, the CTI Console does not have access to the outbound call to the intended recipient of the transfer. Once the outbound call is established, you can consult with the recipient. You do not have to toggle between the two lines, and you do not have the ability to put the recipient on hold.

5. To complete the transfer, select *Complete* and click Go.

Initiating Conference Calls

If you find that you need the assistance of other agents to answer a caller's questions, you can use the conference feature to include the appropriate agents on a call.

To initiate a conference call:

1. Select the appropriate telephone line.

   The selected line must be green.

2. From the Select Call Action drop-down list box, select *Conference.*
3. From the drop-down list box of all numbers from the shared and agent phone books, select a number to be dialed.

   If the number is not there, select *Dial other number* to get an edit box and enter the number to be dialed.

4. Click Go.

   The system notifies the target agent of the incoming call (conference). The PeopleSoft page associated to the caller's case opens for the target agent as it did for you when you first received the call.

   This feature depends upon two parameters set up by the administrator:

   - Default Screen Pop-up URL.
   - Personalization: Screen Pop-up Mode.

     If the default URL for screen pop-up is set and the screen pop-up mode is 0 (pop up when incoming) for the second agent, the agent gets the screen pop-up as soon as the call is transferred.

     If the screen pop-up is set to 1 (pop up after answer), the screen pops up only after the first agent completes the transfer or conference call. However, if the default URL for the screen pop-up is not set, then it doesn't matter whether the mode is 0 or 1. In that case, the second agent gets the screen pop-up only after the first agent completes the transfer or conference call.

     When using one extension with two lines, the outbound call you make to the agent to whom you are inviting to the conference gets initiated on your second line and the incoming call gets placed on hold. After the outbound call is established you can consult with the third party, and place that call on hold. To complete the conference, you need to go back to the first line and select *Complete* and click Go. This action releases the call on the second line and starts the conference on the first line.

     When using two extensions, each with one line, the CTI Console does not have access to the outbound call to the third party. Once the outbound call is established, you consult with the target agent. You do not have to toggle between the two lines, and you do not have the ability to put the recipient on hold. To start the conference, select *Complete* and click Go.

5. After consulting with the target agent, select *Complete* from the Select Call Action drop-down list box, and click Go.

Working with the Hold Status

Putting calls on hold and retrieving calls on hold is likely to be the call action you perform most.

 Copyright © 1988-2007, Oracle. All rights reserved.

To place a call on hold:

1. Select the appropriate telephone line.

   The selected line must be green.

2. From the Select Call Action drop-down list box, select *Hold.*
3. Click Go.

To retrieve a call on hold:

1. Select the appropriate telephone line.

   The selected line must be green.

   The retrieve option is automatically selected as the current option in the drop-down list box when there is a call on hold.

2. Click Go.

Disconnecting a Caller

After you have finished with a call, follow this procedure to release the call.

To release a call:

1. Select the appropriate telephone line.

   The selected line must be green.

2. On the console control bar, select *Release* from the Lines drop-down list box.
3. Click Go.

   The system automatically places you in wrap-up mode, which enables you to complete any remaining work before accepting more incoming calls. Technically, when in wrap-up mode, your status is Agent Not Ready. When you are ready to accept incoming calls, select *Agent Ready.*

Switching Agent Ready Status

Your agent status determines whether you can receive incoming calls.

Agent statuses include:

- *Agent Ready*

  To activate *Agent Ready* status:

  From the Select Call Action drop-down list box, select *Agent Ready.*

  When you are ready, the system routes incoming calls to your extensions.

- Do Not Disturb

  To activate Do Not Disturb status:

  From the Select Call Action drop-down list box, select *DND.*

  With Do Not Disturb, your extensions will not accept incoming calls.

  Note. This status is not available to agents associated with an ACD queue.

- *Agent Not Ready*

Copyright © 1988-2007, Oracle. All rights reserved.

To activate *Agent Not Ready* status:

From the Select Call Action drop-down list box, select *Agent Not Ready.*

This status is typically used when agents are at their desks, but temporarily unable to receive calls. While you are not ready, the system routes calls to other available agents.

Note. This status applies only to agents associated with an ACD queue.

Dialing an Outbound Call

You can use PeopleSoft CTI to place a call while the agent status is either *Agent Ready* or *Agent Not Ready.*

Note. If you have multiple extensions assigned to you, *do not* call one of your extensions from the other.

To place an outbound call:

1. Select your status from *Agent Ready* or *Agent Not Ready.*
2. Select the option next to the telephone icon representing a free line.

   For example, if you had a customer on hold on one line, select the icon for the second line.
3. From the Select Call Action drop-down list box, select *Dial.*
4. From the drop-down list box showing all numbers from the shared and agent phone books, select a number to be dialed.

   If the number is not there, select Dial other number to get an edit box and enter the number to be dialed.
5. Click Go.

   As with any other call you receive, you have access to all the call actions for calls you initiate. You can transfer the person that you've called, place the line on hold, or initiate a conference with another party.

   Note. The system places you in *Agent Not Ready* status until you release the call.

   When making an outbound call:

   - A URL can be specified to display a page.

     When displaying the page, the outbound call data is also attached to the URL so that an application can collect information such as the ANI, DNIS, and so on.
   - A context ID (such as a customer case number or an invoice number) is attached to the call data.

     This allows PeopleSoft application context to be passed on to Cisco or Genesys middleware where it can be stored. This context can be used to establish a relationship between the outbound call and the application context when the call was made.

See Also

Chapter 3, "Configuring PeopleSoft CTI," Using the Shared Phone Book Page, page 25

Completing a Call

When you disconnect or release a call from the mini console, the system disconnects you from the CTI middleware, and the mini console becomes disabled (unavailable for entry). However, the PeopleSoft page remains active so that you can finish updating information if needed.

 Copyright © 1988-2007, Oracle. All rights reserved.

The availability of an agent to accept incoming calls depends on how the system administrators of the agent have set up the agent's CTI middleware. Genesys has a setting known as *wrap-up time*, which allows an agent a certain amount of time to update the previous caller's information or complete related work before accepting more calls.

Using Hot Keys

Hot keys are combinations of keyboard buttons you can press instead of using a mouse. To help you easily select options, PeopleSoft CTI offers the following hot keys:

Hot Key	Description
ALT + R	Registers your phone extensions with the CTI middleware.
ALT + 1	Makes Extension 1 the active line.
ALT + 2	Makes Extension 2 the active line.
ALT + S	Presents a list of applicable call actions for you to select.
ALT + P	Presents a list of frequently called telephone numbers for you to select.
ALT + G	Enables you to execute a call action.
ALT + C	Enables you to cancel a call action.
ALT + Z	Enables you to check agent status.

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 5

Configuring REN Servers

This chapter provides overviews of real-time event notification (REN) servers and Secure Sockets Layer (SSL) enabled REN servers and discusses how to:

- Configure REN server security.
- Configure REN servers.
- Configure REN server and SSL-enabled REN server clusters.
- Configure a reverse proxy server with a REN server.

Understanding REN Servers

This section discusses:

- REN server failover, scalability, and security configuration.
- REN server failover.
- REN server clusters.

The REN server, an application server domain process, is essential to PeopleSoft MultiChannel Framework (MCF) architecture. MCF events are sent to REN servers, which deliver them to recipients of those topics.

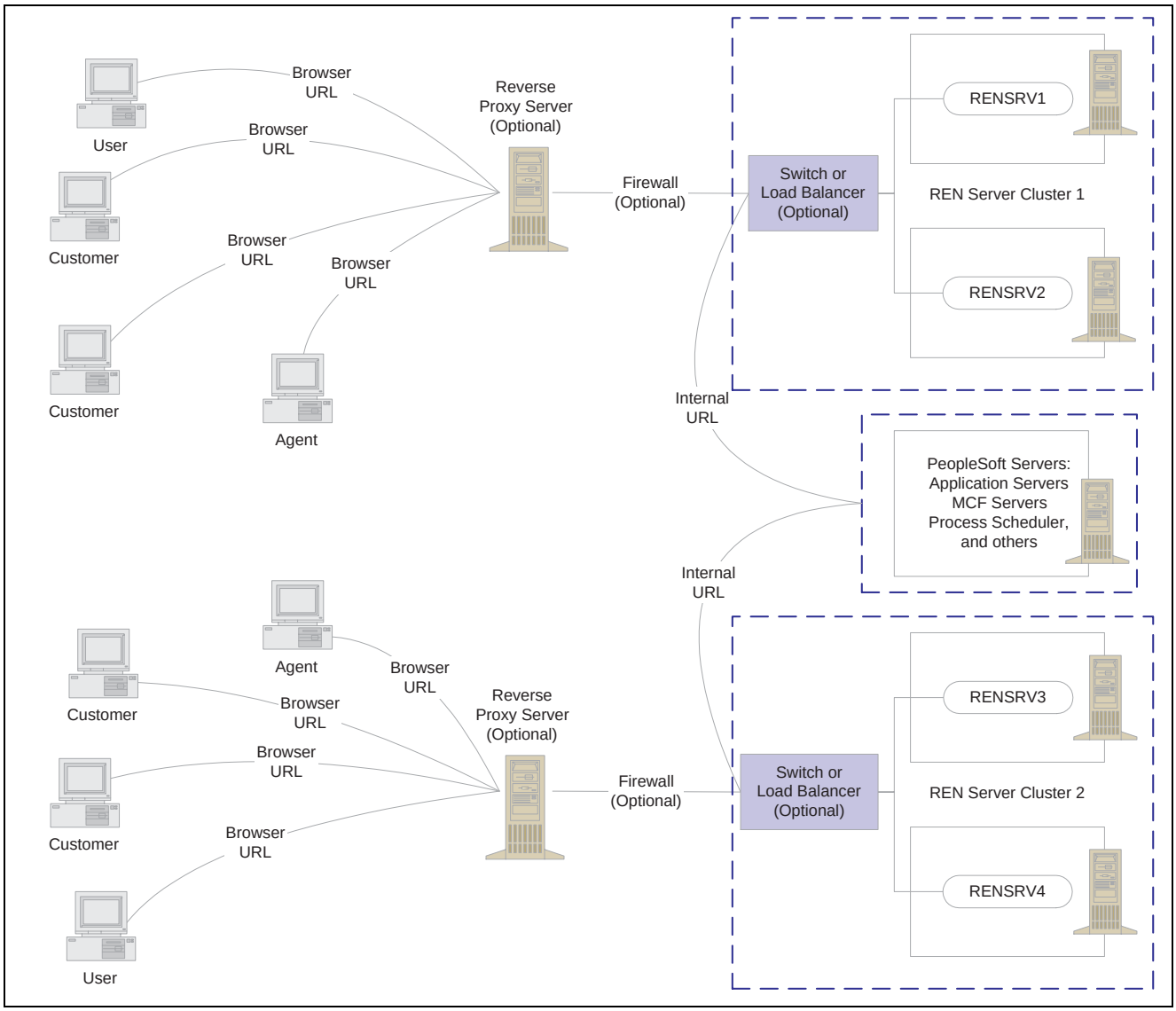
REN servers are also used by other PeopleSoft applications to push event notifications to users, such as the Reporting Window output option and the Optimization Progress Window.

The REN server is a modified web server using the HTTP 1.0 or 1.1 communications protocol. Communication with MCF server processes and browser windows is bidirectional, as they maintain persistent connections to the REN server. Events can be sent proactively to browser windows without polling or page refreshes.

REN Server Failover, Scalability, and Security Configuration

REN servers can be configured to support both failover and scalability, and should be protected with firewalls and appropriate security measures, as illustrated in the following diagram:

Copyright © 1988-2007, Oracle. All rights reserved.



REN server configuration example

REN Server Failover

Although the REN server is integrated into an application server domain, it is not a standard PeopleTools server process (it has no database connection) and therefore has a separate failover mechanism. Two scenarios exist for failure recovery:

- For a standalone REN server, BEA Tuxedo restarts the server if it fails.

  MCF servers and consoles reconnect to the REN server. However, any active browser sessions (such as MCF chat) will be interrupted until a connection can be reestablished between the chat console and the restarted REN server.

- For clustered REN servers, each REN server in the cluster is a peer that mirrors the current state.

  This configuration has two advantages over a standalone REN server:

  - Clustered REN servers guard against hardware failure (provided that the clustered REN servers are on different host machines).

Copyright © 1988-2007, Oracle. All rights reserved.

- Active browser sessions are not lost.

REN Server Clusters

You can configure a REN server cluster with only one REN server member. However, a REN server cluster that is configured with two or more REN servers provides failover.

All REN servers in a cluster mirror each other and appear to external processes as a single URL. The REN server cluster must have an HTTP load balancer or switch as its front end. All connections with browsers and application server processes address the front end's URL. The load balancer should use an active standby content-switching rule to route all traffic to a designated REN server in the cluster. The front end selects an alternate member of the cluster only when the designated REN server fails to respond.

The REN server cluster maintains mirrored state in all members by relaying events with HTTP messages. The REN server cluster therefore does not address scalability issues. Clustering REN servers does not improve performance and may increase processing overhead and internal network traffic. The internal HTTP connections between cluster members should be high speed for best performance. Because of the overhead involved in synchronous cluster members, each member of a cluster can handle less load than a REN server in a cluster with only one REN server.

Note. In an environment where there are multiple REN servers within a single cluster, the primary REN server sends synchronization data to the other members of the cluster. If any of these synchronization messages fail, then the primary REN server will retry up to cluster_retry_count times. The minimum value of this parameter, cluster_retry_count in psrenconfig.txt is 0, which means that the REN server will not retry.

If a REN server crashes, it does not rejoin the cluster, because it would not be synchronized with the other clustered REN servers. The entire cluster must be shut down and rebooted to restore all members back to full participation.

Incoming cluster requests must eventually route to the front end's HTTP address. Queue servers and application servers use the cluster URL, which is typically set to be the URL of the front end. Browser clients make requests using the browser URL, which may be set to the front end, or to a server that proxies to the load balancer. If browser transactions are encrypted with SSL, then the browser URL is an HTTPS address to a reverse proxy server or SSL accelerator.

Note. If you use SSL between the browser and REN server, then you must use a reverse proxy server or SSL accelerator, unless you have configured an SSL-enabled REN server.

Understanding SSL-Enabled REN Servers

You can enable a secure channel of communication between the clients and the REN server by enabling SSL on the REN server using openssl. The SSL protocol runs above TCP/IP and below higher-level protocols, such as HTTP and IMAP4. By using TCP/IP on behalf of higher-level protocols, openssl allows an SSL-enabled server to authenticate itself to an SSL-enabled client, a client to authenticate itself to a server, and both machines to establish an encrypted connection.

This section discusses:

- Installing digital certificates.
- Authenticating sever and client.
- Performance and scalability for SSL-enabled REN servers.

Copyright © 1988-2007, Oracle. All rights reserved.

Installing Digital Certificates

REN servers require digital certificates to work in SSL mode. The servers pick up the certificates from the PeopleTools database. The certificates must be imported into PeopleTools database from PeopleTools, Security, SecurityObjects, Digital Certificates. Certificates that are installed in the database will have a unique combination of certificate type and alias.

The certificate type that is used for the server should be of the type CERT, and the alias is <machine name>.<domain name>. When the certificate is configured with a unique alias name, it should be associated with the REN server that is SSL enabled. The REN server loads its server certificate from the database at the start up.

See Appendix B, "Installing Digital Certificates for REN SSL," page 471.

Authenticating Server and Client

For server authentication, the server sends its certificate to the client as a part of the SSL handshake and the client authenticates by verifying the Certificate Authority (CA) of the certificate against its trusted keystore. When the REN server is configured for SSL, all clients must trust the CA of the server certificate to participate in a successful communication.

Client authentication verifies the clients's authenticity to participate in a communication with the server. When the REN server is configured for client authentication, all clients must supply a valid client certificate to participate in a successful communication.

All clients must use the REN cluster's HTTPS URL to communicate in the SSL mode. If the REN server is SSL only, access will be denied to any client trying to communicate with a HTTP URL port. The browser-based clients, the application server client, and the REN Java clients should be configured appropriately to communicate with an SSL-enabled REN server.

Performance and Scalability for SSL-Enabled REN Servers

During an SSL transaction, the handshake is an added overhead that occurs. However, for every transaction, the handshake is done once to authenticate the server and the client. After authentication, the data is digitally signed, encrypted, and exchanged on an established session. For each console, authentication establishes a session only once, and all subsequent transactions do not inherit any overhead of authentication.

Configuring REN Server Security

This section provides an overview of REN server security configuration and discusses how to define permission lists for REN server access.

Understanding REN Server Security Configuration

Protect the REN server behind firewalls. A reverse proxy server can be used between browser clients and the REN server. Browser sessions can be SSL-encrypted using a reverse proxy server or hardware SSL accelerator.

Note. The security of your PeopleSoft system, and configuration of load balancers, switches, and reverse proxy servers, is beyond the scope of this document. Refer to your PeopleBooks for more information.

 Copyright © 1988-2007, Oracle. All rights reserved.

REN server access from browser clients is restricted to users who are currently logged into PeopleSoft with appropriate REN server permissions. You must enable single signon security to obtain REN server access. Permission to access REN server applications is granted on permission lists, which are in turn associated with security roles and user IDs. Clients lacking access permission receive a 403 Forbidden page from the REN server.

Note. REN server access requires that single signon is enabled.

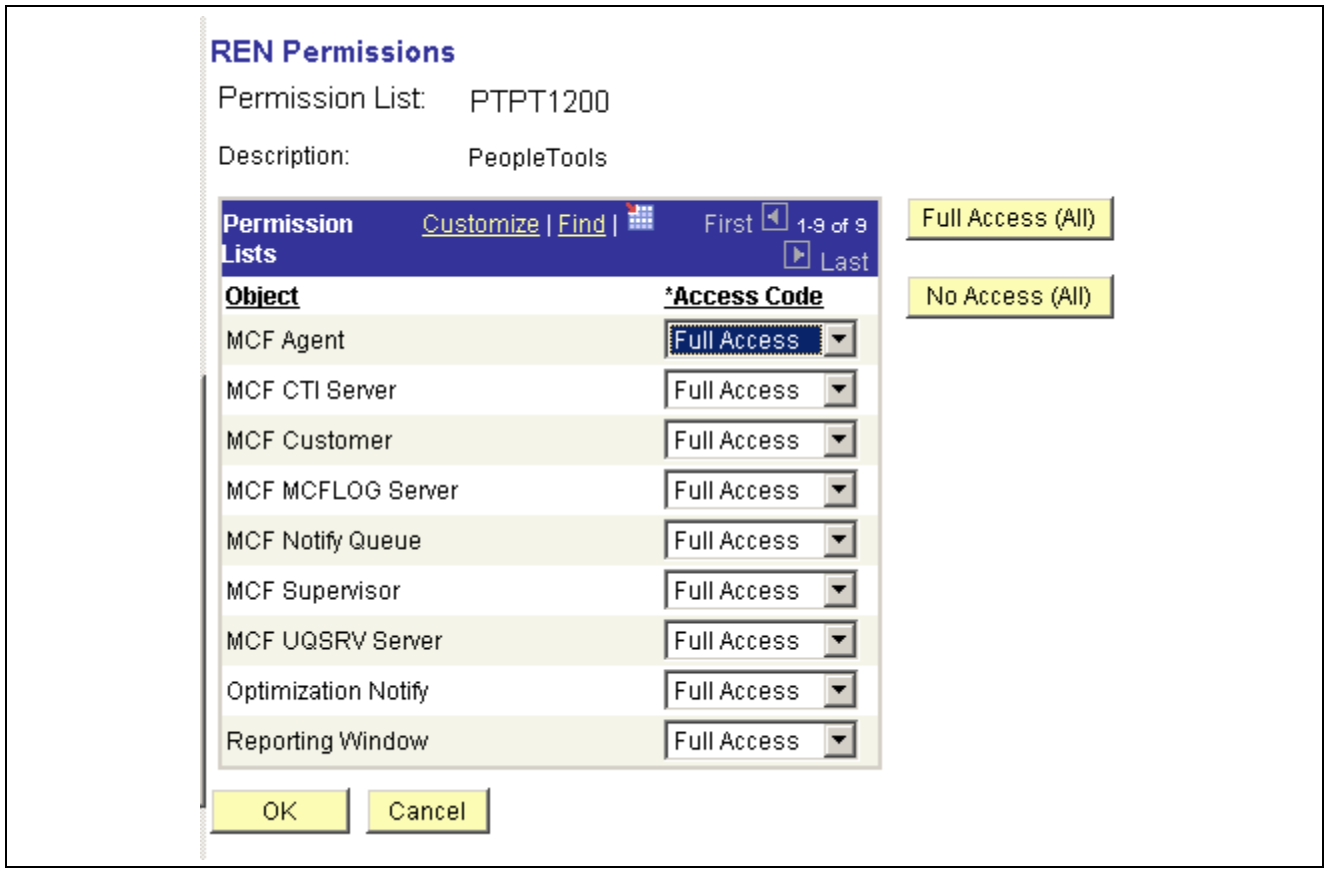
See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, "Getting Started with Security Administration"

Enterprise PeopleTools 8.49 PeopleBook: System and Server Administration, "Getting Started with System and Server Administration"

Defining Permission Lists for REN Server Access

The following REN Permissions page shows the objects and permissions that are defined for permission list PTPT1200. Customers can create their own permission list and define access to REN server in the permission list that they created.



REN Permissions page

To define permission lists for REN server access:

1. Select PeopleTools, Security, Permissions & Roles, Permission Lists.
2. On the search page, search for and select your permission list.

Copyright © 1988-2007, Oracle. All rights reserved.

3. On the Permission List page, select the PeopleTools tab.
4. Click Realtime Event Notification Permissions.
5. On the REN Permissions page, select your permissions.

   To enable REN server access for roles that are defined with the current permission list, select *Full Access* for each object that is required by the role. For example, users who require access to the MultiChannel Console must have Full Access defined for the MCF Agent object.

   The MultiChannel Console link appears in the universal navigation header for any user with full access permissions defined for the MCF Agent object. However, the user must also be configured as an MCF or CTI agent to access the MultiChannel Console or CTI console.

Note. To enable access to the Report-to-Window functionality, add WEBLIB_RPT to the permission list's Web Libraries page, and set Reporting Window to Full Access on the REN Permissions page.

Grant full access to the MCF CTI Server object only on the permission list that is assigned to the CTI server role. No other users should have MCF CTI Server access.

The user ID that is configured to start the Process Scheduler must have full access to the Reporting Window REN permission on at least one permission list for that user ID. If the user ID does not have full access to the Reporting Window, then the pop-up window will stay in a status of queued.

See Also

Chapter 3, "Configuring PeopleSoft CTI," Required Security for PSMCAPI, page 17

Configuring REN Servers

To configure REN servers, use the REN Server (REN_SERVER_CMP) component.

This section provides an overview of selecting REN server configuration options and discusses how to:

- Configure REN servers and SSL-enabled REN servers.
- Define REN servers.

Understanding REN Server Configuration Options

Depending on your requirements, choose one of two REN server creation and configuration options:

- To create a single REN server in a particular database using default configuration parameters, create an application server domain using PSADMIN.

  Event Notification is enabled by default in the quick-configure menu. An associated REN server cluster is also created by default.

- To create additional REN servers in a particular database, configure each REN server as required on the REN Server Definition and REN Server Cluster pages.

  Then create the associated application server domains. Event Notification is enabled by default in the quick-configure menu.

 Copyright © 1988-2007, Oracle. All rights reserved.

When a REN server starts up, it looks for configuration information in the database, using the application server domain name and host name as keys. If the associated configuration information exists in the database, the REN server uses it. If no such configuration information exists, the REN server is configured using defaults, which also configure a REN server cluster for each REN server. You can change the default REN server configuration using the REN Server Configuration pages, but such changes do not take effect until the REN server starts up again.

Note. You can create only one REN server per application server domain.

This section discusses some possible REN server configurations that are dependent on domain server topology.

Simple Configuration: Mycompany.com

In this configuration, the REN server is on the host machine MachA, the REN server uses the default port number 7180, DNS addresses the host machine as MachA.mycompany.com, and no SSL or reverse proxy server is involved:

Parameter	Value
PeopleSoft Pure Internet Architecture Authentication Token Domain	mycompany.com
Authentication Domain in REN Server Cluster Configuration	mycompany.com
REN Server Cluster Root Path	/psren
REN Server Cluster URL	http://MachA:7180
REN Server Browser URL	http://MachA.mycompany.com:7180

Simple Configuration with SSL-Enabled REN Server: Mycompany.com

In this configuration, the REN server is on the host machine MachA, the REN server uses the default port number 7143, and DNS addresses the host machine as MachA.mycompany.com. The REN server is SSL-enabled.

Parameter	Value
PeopleSoft Pure Internet Architecture Authentication Token Domain	mycompany.com
Authentication Domain in REN Server Cluster Configuration	mycompany.com
REN Server Cluster Root Path	/psren

Copyright © 1988-2007, Oracle. All rights reserved.

Parameter	Value
REN Server Cluster URL	https://MachA:7143
REN Server Browser URL	https://MachA.mycompany.com:7143

Reverse Proxy Server with Non-SSL Configuration

This configuration includes a single REN server and a reverse proxy server. The reverse proxy server could be either a dedicated reverse proxy server or a web server with a proxy plug-in configured to redirect both PeopleSoft Pure Internet Architecture and REN server requests. The application server host machine is MachA, and the REN server uses its default port 7180. The reverse proxy server is on MachRPS using port 8080 for HTTP. The DNS server must recognize MachRPS.mycompany.com.

Parameter	Value
PeopleSoft Pure Internet Architecture Authentication Token Domain	mycompany.com
Authentication Domain in REN Server Cluster Configuration	mycompany.com
REN Server Cluster Root Path	/psren
REN Server Cluster URL	http://MachA:7180
REN Server Cluster Browser URL	http://MachRPS.mycompany.com:8080

Reverse Proxy Server with SSL Configuration and Secure HTTP

For SSL, install certificates on the reverse proxy server, set the server to encrypt all communications, and use HTTPS URLs from the browser. In this example, the reverse proxy server uses port 8443 for SSL:

Parameter	Value
PeopleSoft Pure Internet Architecture Authentication Token Domain	mycompany.com
Authentication Domain in REN Server Cluster Configuration	mycompany.com
REN Server Cluster Root Path	/psren

Copyright © 1988-2007, Oracle. All rights reserved.

Parameter	Value
REN Server Cluster URL	http://MachA:7180 **Note.** The cluster URL should not be a secure HTTP address if SSL is handled through a reverse proxy server.
REN Server Browser URL	https://MachRPS.mycompany.com:8443 **Note.** This is a secure HTTP address (HTTPS).

Note. If you use SSL between the browser and REN server, you must use a reverse proxy server or SSL accelerator.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, "Getting Started with Security Administration"

Page Used to Configure REN Servers

Page Name	Object Name	Navigation	Usage
REN Server Configuration	REN_SERVER_DET_PG	PeopleTools, REN Server Configuration, REN Server Definition	Define a REN server.

Configuring REN Servers and SSL-Enabled REN Servers

Specify REN server configuration parameters based on your network topology and server arrangement.

Define the parameters for REN server configuration in three locations:

- Authentication token domain, set during PeopleSoft Pure Internet Architecture installation or in web profile configuration.
- Specify REN server configuration parameters in an application server domain using PSADMIN.
- REN server parameters, including cluster and browser URLs, set in the PeopleTools REN Server and REN Cluster components.

  Configuration parameters that are set in the REN Server and REN Cluster components override any defaults in PSADMIN.

Authentication Domain

The authentication domain tells PeopleSoft Pure Internet Architecture the internet domain name that browser clients use when accessing PeopleSoft applications across the internet. The token is required to comply with the same-origin security policy that is enforced by most browsers. The domain name that is specified in the REN Server Configuration page must be identical to the domain name that is specified as the authentication token domain during PeopleSoft Pure Internet Architecture installation.

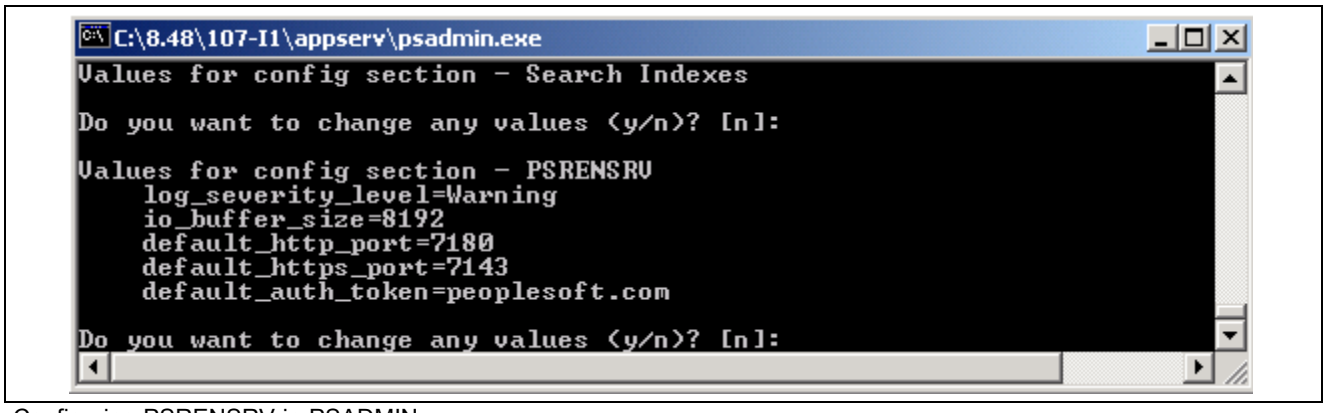
If authentication domain is not set during PeopleSoft Pure Internet Architecture installation, define the authentication domain in web profile configuration to match the REN server configuration.

Copyright © 1988-2007, Oracle. All rights reserved.

Note. You must specify the authentication token domain if you access the REN server and the PeopleSoft Pure Internet Architecture web server using different DNS names from the browser client (for example, if they are on different machines).

Configuring a REN Server and SSL-Enabled REN Server with PSADMIN

If necessary, you can specify parameters in the PSRENSRV section of PSADMIN application server domain configuration, as illustrated in the following example:



Configuring PSRENSRV in PSADMIN

Specify parameters as described in the following table:

Parameter	Default	Description
log_severity_level	Warning	This is the logging level for the REN server. Select from one of the following log severity levels, from less to more logged data: Error, Warning, Notice, Debug. **Note.** Do not use Debug in a production environment.
io_buffer_size	8192	This is the TCP buffer size in bytes that is used for serving content. Do not exceed a value of 65536. If the REN server is running on Microsoft Windows NT, change io_buffer_size to a minimum value of 56000.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameter	Default	Description
default_http_port	7180	This is the REN server's HTTP port. The default value is *7180.* The default_http_port parameter takes effect only when a REN server starts up for the first time and the database does not already contain configuration information for the REN server. **Note.** After the HTTP port number that is assigned to the REN server has been established in the database, the only way to change it is on the REN Server Definition page. Editing the port number in the psappsrv.cfg file does not overwrite the value that is stored in the database.
default_https_port	7143	This is the REN server's HTTPS port for SSL-enabled REN server. The default value is *7143.* **Note.** The https port is used only when the REN server is SSL-enabled. The default_https_port parameter is configured in psappsrv.cfg and will be used when a SSL-enabled REN server starts up for the first time. **Note.** To change the default https port, use the REN Server Definition page. Changing the SSL Port requires the reboot of the REN Server.
default_auth_token	example.com	This is the fully qualified domain name of the application server. This value should match the value of the web server's authentication domain. The default_auth_token parameter takes effect only when a REN server starts up for the first time and the database does not already contain configuration information for the REN server. When configuring the REN server parameters through PSADMIN, do not place a period or dot (.) before the default_auth_token value. For example, the parameter should read default_auth_token=example.com

After specifying REN server configuration parameters, be sure to specify Y (Yes) when asked if you want event notification configured and MCF server configured. Boot this domain from the Domain Administration menu.

Copyright © 1988-2007, Oracle. All rights reserved.

Note. Use PeopleSoft Pure Internet Architecture REN server definition and configuration pages to modify configuration parameters whenever possible. REN server configuration parameters that you make using PSADMIN are written to the psappsrv.cfg file in the application server directory. REN server configuration values that are found in the database override any values that are found in psappsrv.cfg.

Use static IP addresses for your web servers. If you use dynamic IP addresses (DHCP), ensure that the domain name server (DNS) can map fully qualified domain names to the dynamic IP addresses.

If you are using Microsoft Internet Explorer internet security zones, include both the web server and REN server addresses in the same security zone; alternatively, exclude both addresses from security zones.

Socket Binding

The REN server listens on the port that is defined in the REN Server Definition page, which is by default *7180*. However, the host name to which the REN server binds is determined by information in the psrenconfig.txt file for each application server domain. If the host machine contains multiple network interface cards (NICs), then the REN server binds by default to only one NIC, which is given by `uname()` on Unix, or `GetComputerName()` on Microsoft Windows.

To bind a REN server to a specific NIC, manually edit psrenconfig.txt for the appropriate application server domain, changing both set address and set hostname to the IP address and locally-known host name of the NIC. For example:

```
set address   192.168.10.1
set hostname  hostsrv.example.com
```

Note. If you enter an invalid IP address in the psreconfig.txt file, the REN server may not start correctly. Check the REN server log for error messages that identify the issue.

Configuring TCP_NODELAY

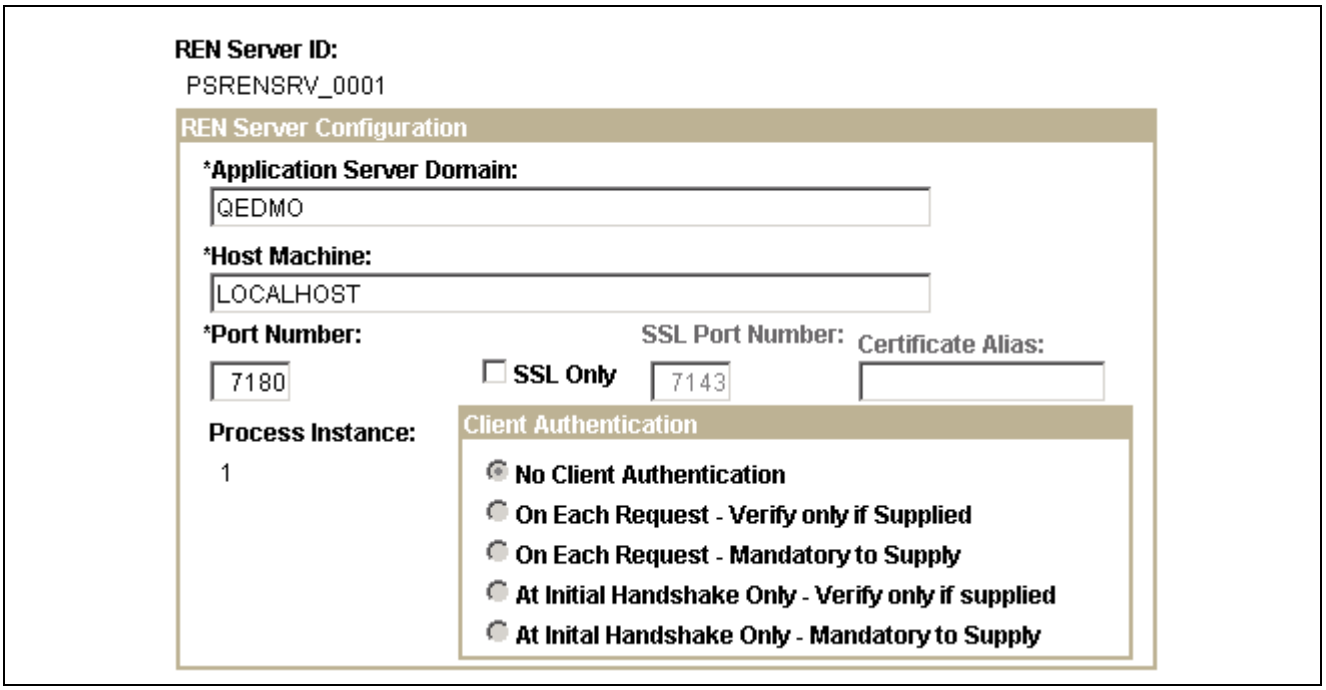
The parameter, TCP_NODELAY in psrenconfig.txt controls whether or not to disable the TCP Nagle algorithm on the TCP packets sent by the REN server. There are two instances of TCP_NODELAY in psrenconfig.txt. TCP_NODELAY in the nssock section is used by non-SSL REN servers, and the instance in the nsopenssl section is used by SSL-enabled REN servers. The TCP Nagle algorithm is generally enabled by default and inserts a short delay before sending small TCP packets. This helps in preventing network overload.

If TCP_NODELAY is set to 0, the TCP algorithm behaves normally. This is the recommended configuration for most applications. However, for certain CTI applications, this parameter must be set to 1 to improve performance. If TCP_NODELAY is set to 1, the TCP Nagle algorithm will be disabled on operating system platforms that support disabling this feature.

Defining REN Servers

Access the REN Server Configuration page.

 Copyright © 1988-2007, Oracle. All rights reserved.



REN Server Configuration page

Application Server Domain Enter the application server domain that is serving this REN server.

Host Machine Enter the name of the host machine on which the specified application server domain runs.

This entry requires the host machine name, not its DNS name. However, the host machine name may need to be fully qualified, for example, machineA.example.com. On a Unix machine, determine the host name by running `uname -a`. On a Microsoft Windows machine, determine the host name by running `hostname` at a command prompt.

Port Number Enter the HTTP port number on which this REN server is addressed.

Change the HTTP port value if multiple REN servers are running on the same host machine to avoid port conflicts.

SSL Only Select to enable SSL on REN server.

Note. If the option is selected, you have to enter the SSL Port.

SSL Port Enter the HTTPS port number on which this SSL-enabled REN server is addressed.

Process Instance Reserved for future use.

Certificate Alias Select a certificate alias to be used as a server certificate by the SSL-enabled REN server.

Note. The certificate alias is stored in the PSKEYDB, PSCERTDB, and PSREN records.

Client Authentication Select to determine the level of client authentication.

Copyright © 1988-2007, Oracle. All rights reserved.

Note. If the browser is configured for client authentication pop-up or the browser has more than one certificate configured, the SSL session ends if the user fails to provide the certificate within three heartbeats. To avoid such a session timeout, the user must either accept the client certificate within a heartbeat or increase the session timeout value in psrenconfig.txt.

The following table illustrates the client authentication values:

Parameter	Flag Value	Description
No Client Authentication	0	Client authentication is disabled.
On each Request-Verify only if Supplied	1	Client authentication is enabled The server sends a client certificate request to the client. Verification happens only if the certificate is provided. If the verification process fails, the TLS/SSL handshake is immediately terminated. If the client does not return any certificate, SSL communication still continues
On Each Request-Mandatory to Supply	3	Client authentication is enabled and mandates that the client provide the certificate. If the client does not return a certificate, the TLS/SSL handshake is immediately terminated with a handshake failure' alert. If the client returns a certificate, it is verified. The communication fails if the verification fails.
At Initial handshake Only-Verify only if Supplied	5	Client authentication is enabled and requests a client certificate on the initial TLS/SSL handshake only. Verification happens only if the certificate is provided. If the client does not provide any certificate, SSL communication still continues. If verification fails, the TLS/SSL handshake is immediately terminated.
At Initial handshake Only-Mandatory to Supply	7	Client authentication is enabled and mandates that the client provide the certificate only in initial TLS/SSL handshake.

Configuring REN Server and SSL-Enabled REN Server Clusters

A cluster is typically a collection of REN servers among which the session information is replicated. One cannot add both SSL and non-SSL servers in a single cluster.

 Copyright © 1988-2007, Oracle. All rights reserved.

To configure REN server clusters, use the REN Cluster (REN_CLUSTER_CMP) component.

This section discusses how to:

- Cluster REN servers and SSL-enabled REN servers.
- Specify REN server ownership.
- Specify REN server cluster members.

REN server clusters address failover and scalability.

Pages Used to Configure REN Server and SSL-Enabled REN Server Clusters

Page Name	Object Name	Navigation	Usage
REN Server Cluster	REN_CLUSTER_PG	PeopleTools, REN Server Configuration, REN Server Cluster, REN Server Cluster	Define a REN server cluster.
REN Server Cluster Owner	REN_OWNER_PG	PeopleTools, REN Server Configuration, REN Server Cluster, REN Server Cluster Owner	Define the ownership of the REN server cluster.
REN Server Cluster Members	REN_CLUST_RSERV_PG	PeopleTools, REN Server Configuration, REN Server Cluster, REN Server Cluster Members	Define the REN server cluster's member REN servers.

Clustering REN Servers and SSL-Enabled REN Servers

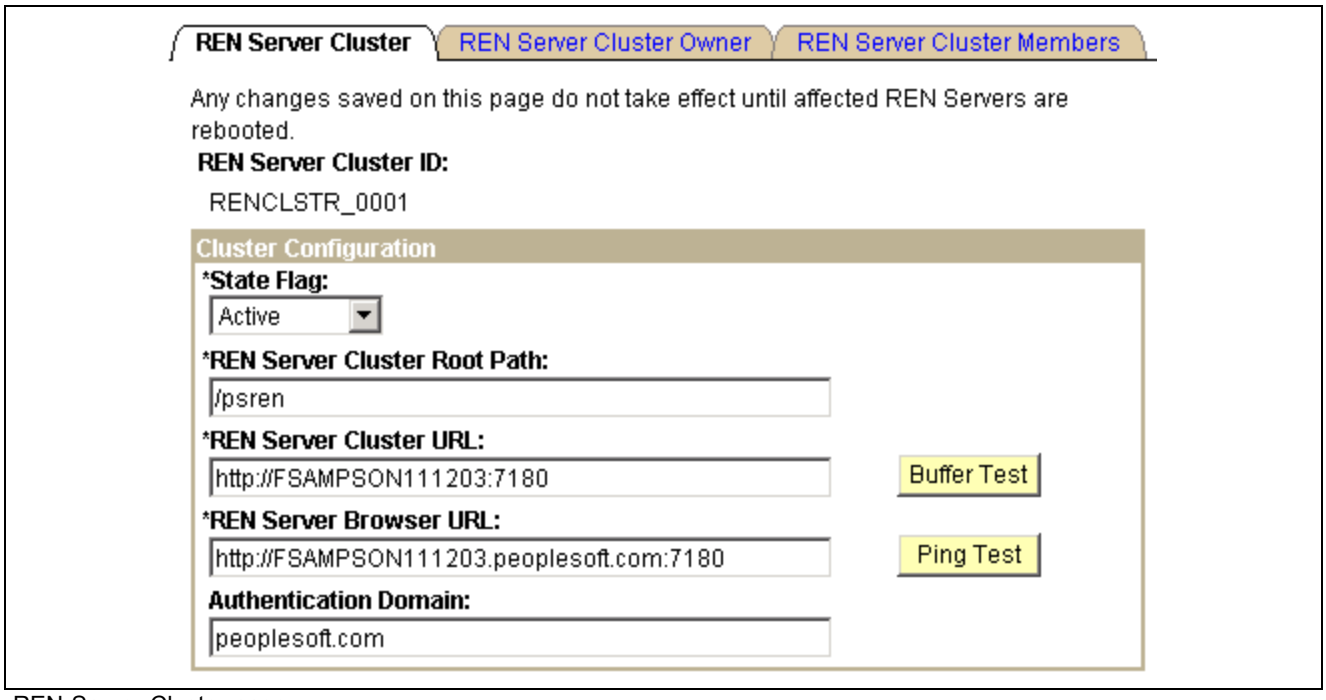
A REN server serves requests only if it is a part of the cluster. If the REN server is SSL-enabled:

- All the member REN servers must be SSL–enabled REN servers.
- All the member REN servers must use the same server certificate.
- Both REN Server Cluster URL and REN Server Browser URL must start with HTTPS and use the HTTPS port.

Note. When the administrator changes the REN server to be in SSL mode, he or she must also ensure that the REN server is a member of SSL clusters only. In any given REN cluster, all REN servers that are members must be either SSL–only servers or non–SSL servers. For SSL-enabled REN servers, use SSL-enabled PIA.

Access the REN Server Cluster page.

Copyright © 1988-2007, Oracle. All rights reserved.



REN Server Cluster page

By default, if you start a REN server from PSADMIN without configuring a REN server cluster, a cluster is created with a cluster ID RENCLSTR_000n

State Flag

Select *Active* or *Inactive.*

This field determines whether the cluster can receive new client requests. For scalability, configure multiple REN server clusters with the same ownership and set them to active status. Then the reporting window and customer chat applications will direct new client requests to a randomly chosen active REN server cluster. If all clusters are inactive, the client receives an error message.

If the cluster supports MCF servers, current chat sessions continue even after a cluster is inactive. But the MCF system does not route any additional requests to an inactive cluster.

Inactivate a cluster before deleting the cluster, or before removing a member REN server from the cluster. You can inactivate a REN server cluster without deleting the cluster.

REN Server Cluster Root Path

The default REN server cluster root path is /psren. Change this as required so that multiple REN server clusters are addressable through a single reverse proxy server.

Changes to the root path should also be reflected in the URL mapping of any reverse proxy server.

REN Server Cluster URL

The REN server cluster URL is the address that is used to reach the REN server cluster internally.

This is the URL that is used by internal processes. If the MCF cluster is served by a REN server cluster, the cluster URL is that of the switch or load balancer in front of the clustered REN servers. The cluster URL must be unique for each cluster. No two clusters can address the same cluster URL. Specify the cluster URL in the form <http://<DNS_machine_ name>:<port>, where:

 Copyright © 1988-2007, Oracle. All rights reserved.

- <DNS_machine_name> is the server machine name that is recognized by your DNS.
- <port> is the REN server port number; the default value is 7180.

  This port number is the REN server port number or the port number of a proxy server, load balancer, or other front end.
- The protocol must be HTTP if the REN server is non-SSL; for an SSL-enabled REN server, the protocol must be HTTPS.

Buffer Test Click Buffer Test to initiate a test of the REN servers' ability to break up and send a large file using multiple internal buffers.

The buffer test bypasses REN server security, and does not depend on specified domain names (authentication domain), so you can use it to verify that the REN server is running on the network.

REN Server Browser URL The REN server browser URL is the address that is used by external clients and by agent chat to reach the application that is served by this REN server cluster.

The browser URL may be different from the cluster URL, which should not have to go through any firewall, reverse proxy server, or other outward-facing security barrier. If the REN server is reached through a load balancer, switch, or reverse proxy server, specify the fully qualified URL of that device as accessed from the user's browser. The URL must be the address of the gateway machine (proxy server, load balancer, or SSL accelerator). Specify the address in the form http: or https://<DNS_machine_ name>.<domain_name>:<port>, where:

- <DNS_machine_name> is the server machine name that is recognized by your DNS.
- <domain_name> is the fully qualified domain name that is recognized by your DNS.
- <port> is the REN server port number; the default value is 7180.

  This port number is the REN server port number or the port number of a proxy server, load balancer, or other front end.

Note. If the REN server is SSL-enabled, the browser URL must be HTTPS.

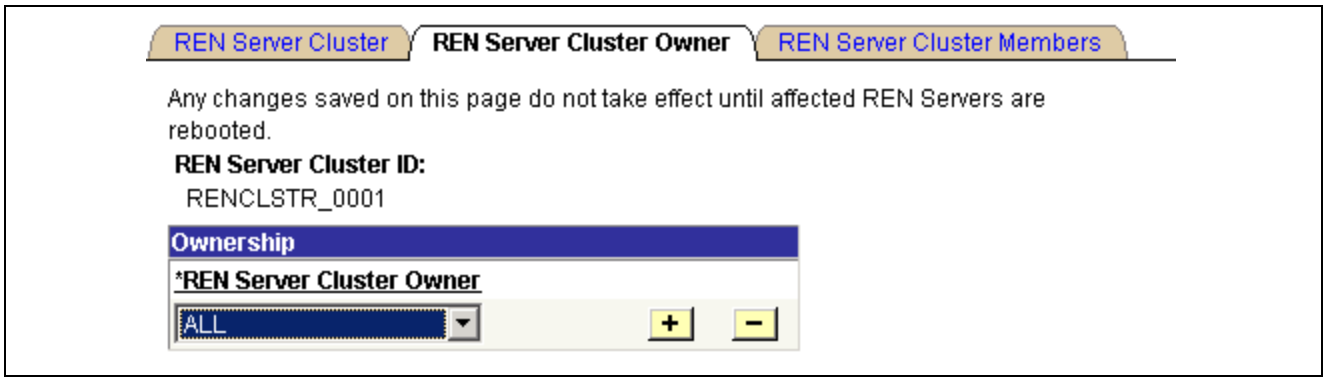
Ping Test Click Ping Test to initiate a test of the REN server that is specified in the browser URL fields. Failure may indicate that a URL or authentication domain is incorrectly specified, the REN server is not running, or single sign-in is not implemented.

Authentication Domain Enter the authentication domain. This must be the same as the authentication domain that is specified in the PeopleSoft Pure Internet Architecture installation or in the web profile configuration.

Specifying REN Server Ownership

Access the REN Server Cluster Owner page.

Copyright © 1988-2007, Oracle. All rights reserved.



REN Server Cluster Owner page

REN Server Cluster Owner Select the owner of this REN server cluster from the drop-down list box. Choose from the following values:

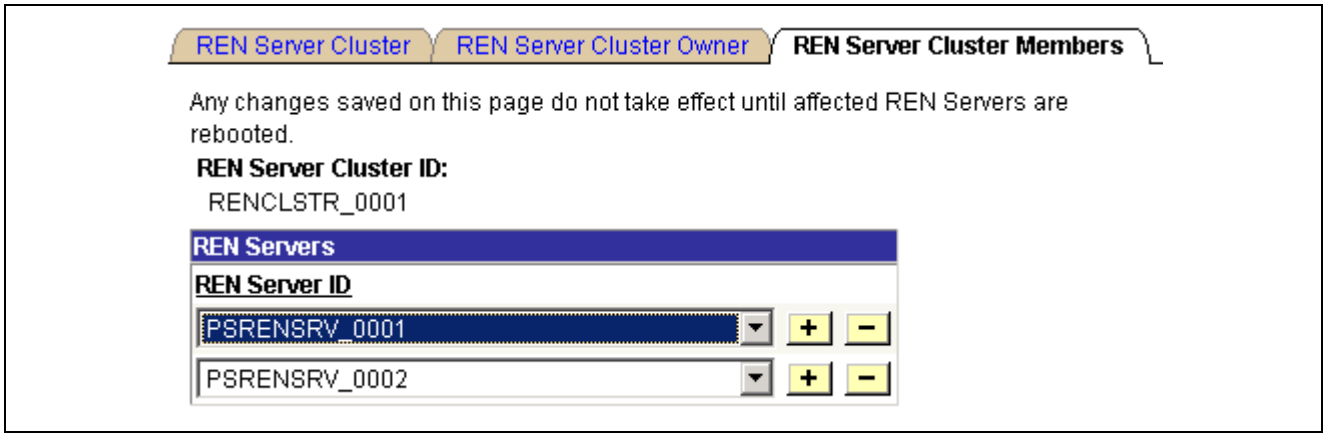
- *All*
- *MCF*
- *Optimization*
- *Reporting*

Specifying an owner for a REN server cluster limits client access to that cluster. This is useful to ensure performance under load.

Specifying an owner for a REN server cluster also supports security. For example, an MCF cluster can be created only on a REN server cluster that is owned by MCF or ALL.

Specifying REN Server Cluster Members

Access the REN Server Cluster Members page



REN Server Cluster Members page

REN Server ID Select a REN server from the drop-down list box.

Each REN server can belong to only one REN server cluster.

Copyright © 1988-2007, Oracle. All rights reserved.

Configuring a Reverse Proxy Server with a REN Server

This section provides an overview of reverse proxy server (RPS) configuration and provides examples.

Understanding RPS Configuration

Production PeopleSoft installations may configure the REN server behind an RPS. The RPS isolates the REN server and other web servers from the open internet, provides SSL session handling, and presents a single-server origin to outside clients. PeopleSoft customers may put REN servers and PeopleSoft Pure Internet Architecture web servers behind one RPS, or just REN servers.

This example presents one possible configuration for a REN server running on one host machine, and installs an RPS to run on a second host machine, using BEA WebLogic 8.1. The RPS redirects clients to both a REN server and to the PeopleSoft Pure Internet Architecture web server.

These examples assume that:

- You have installed PeopleTools 8.48 on both host machines.
- You have configured a web server using the default parameters on the first host machine.
- You have configured a REN server using the default parameters on the first host machine.

Example: Configuring a WebLogic 8.1 RPS for a REN Server on Another Host Machine

To configure an RPS for a REN server on another host machine:

1. Install a new web server domain on the second machine.

   Name the domain *rps*.

   Configure the following values:

   - AppServer Name: *<application_server_machine_name>*
   - JSL Port: *9999*

     The RPS will not make Jolt connections.
   - HTTP Port: *8080*
   - HTTPS Port: *8443*
2. Start the new web server.

   Navigate to <PS_HOME>\webserv\rps, and run `startPIA.cmd`.
3. Sign in to the WebLogic Server Administrative Console for the *rps* web server.

   Access the WebLogic Server Administrative Console at http://<webserver>:<port>/console (for example, http://localhost:8080/console).

   When prompted for a user name and password, specify the WebLogic system ID and password. If you've followed the default WebLogic Server install, the ID and password are *system* and *password*.
4. Using the console's hierarchical navigation, navigate to rps, Deployments, Applications, PeopleSoft. Select the Targets tab.

   Clear the PIA option.

Copyright © 1988-2007, Oracle. All rights reserved.

Click Apply.

5. Using the console's hierarchical navigation, navigate to rps, Deployments, Web Application Modules, HttpProxyServlet. Select the Targets tab. Select the PIA option. Click Apply.
6. For better web server performance, navigate to rps, Servers, PIA. Select the Protocols tab, select the HTTP tab, and set both Duration and HTTPS Durationto *120* secs.
7. Stop the *rps* web server.

   Navigate to <PS_HOME>\webserv\rps and run `stopPIA.cmd`.
8. Configure RPS parameters for the *rps* server.

   Locate the file web.xml at PS_HOME/webserv/rps/applications/HttpProxyServlet/WEB-INF.

   Edit web.xml in a text editor, changing the WebLogic port and WebLogic host from *8080* to *80* (the value *8080* is a default value that is derived during installation of the domain *rps*). For example:

```
<init-param>
 <param-name>WebLogicPort</param-name>
 <param-value>80</param-value>
 <description>HTTP listen port of WebLogic PIA/PORTAL server.</description>
</init-param>
```

   To specify the associated REN server, (which is on another machine), edit web.xml, changing the REN server host machine, port, and root URL from their default RPS values. For example:

```
<init-param>
 <param-name>WebLogicHost</param-name>
 <param-value>MACHINE_2</param-value>
 <description>Hostname of REN server.</description>
</init-param>
<init-param>
 <param-name>WebLogicPort</param-name>
 <param-value>7180</param-value>
 <description>Listen port of REN server.</description>
</init-param>
```

   Another example is:

```
<servlet-mapping>
 <servlet-name>RENHttpProxyServlet</servlet-name>
 <url-pattern>/psren/*</url-pattern>
</servlet-mapping>
```

9. Reboot the RPS web server.

   Navigate to <PS_HOME>\webserv\rps, and run `startPIA.cmd`.
10. (Optional) Configure and enable SSL on the RPS machine.

Note. When using Apache 1.3.x or 2.0.x RPS, you must configure the kn_response_flush_override and the flush_rps_buffer_size_for_knjs parameters in the psrenconfig.txt file. If you are using Apache 1.3.x, set both of these parameters to *4096*. If you are using Apache 2.0.x, set both parameters to *8192*. Apache needs both parameters present with the same buffer size. The kn_response_flush_override parameter flushes a message, while the flush_rps_buffer_size_for_knjs parameter flushes the stay-alive.

 Copyright © 1988-2007, Oracle. All rights reserved.

Example: Configuring Oracle Application Server (OAS) to Proxy for a REN Server

If your site uses Oracle Application Server (OAS) as the PIA/portal web server, then special proxy configuration is required to work with REN server. OAS automatically deploys the Oracle Http Server (OHS), an Apache-based reverse proxy server. OHS is installed to proxy requests to the PIA Web Application.

To proxy for RenServer, find and edit the httpd.conf configuration file (typically at OAS_HOME\Apache\Apache\conf). Make the following modifications to the file:

1. Move the line LoadModule proxy_module modules/ApacheProxyModule.dll to the bottom of the file.
2. Comment out the line AddModule mod_proxy.c.
3. Add the following five lines after LoadModule proxy_module:

```
<IfModule mod_proxy.c>
  ProxyRequests Off
  ProxyPass /psren http://machine:7180/psren
  ProxyPassReverse /psren http://machine:7180/psren
</IfModule>
```

4. Reboot OAS and OHS.

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 6

Configuring PeopleSoft MCF Servers and Clusters

This chapter provides an overview of PeopleSoft MultiChannel Framework (MCF) scalability and failover and discusses how to configure MCF clusters.

Understanding PeopleSoft MCF Server and Cluster Architecture

This section discusses:

- PeopleSoft MCF server configuration.
- PeopleSoft MCF cluster architecture.
- Queue server and queue server failover.
- Logical queues and physical queues.
- MCF log server and log server failover.
- Queue server scalability.
- Recommended configurations.

PeopleSoft MCF Server Configuration

PeopleSoft MCF depends on processes that are configured and booted as part of an application server domain. Configure MCF processes (servers) through PSADMIN, along with other processes in each application server domain.

Note. The REN server process can be used by applications that are separate from the queue server and MCF log processes. In this case you can configure the application server domain for event notification without creating the MCF servers.

After considering performance and failover issues, the MCF system administrator provides configuration information that describes the arrangement of queues, domains, queue server processes, REN server processes, MCF processes, and URL addresses.

REN server processes are configured on PeopleSoft Pure Internet Architecture pages before or after being initiated in an application server domain. Each queue server process is uniquely identified in the system by combining the machine name, domain subdirectory name, and process identifier. MCF log processes use the same queue-server identification scheme.

See Also

Chapter 5, “Configuring REN Servers,” page 55

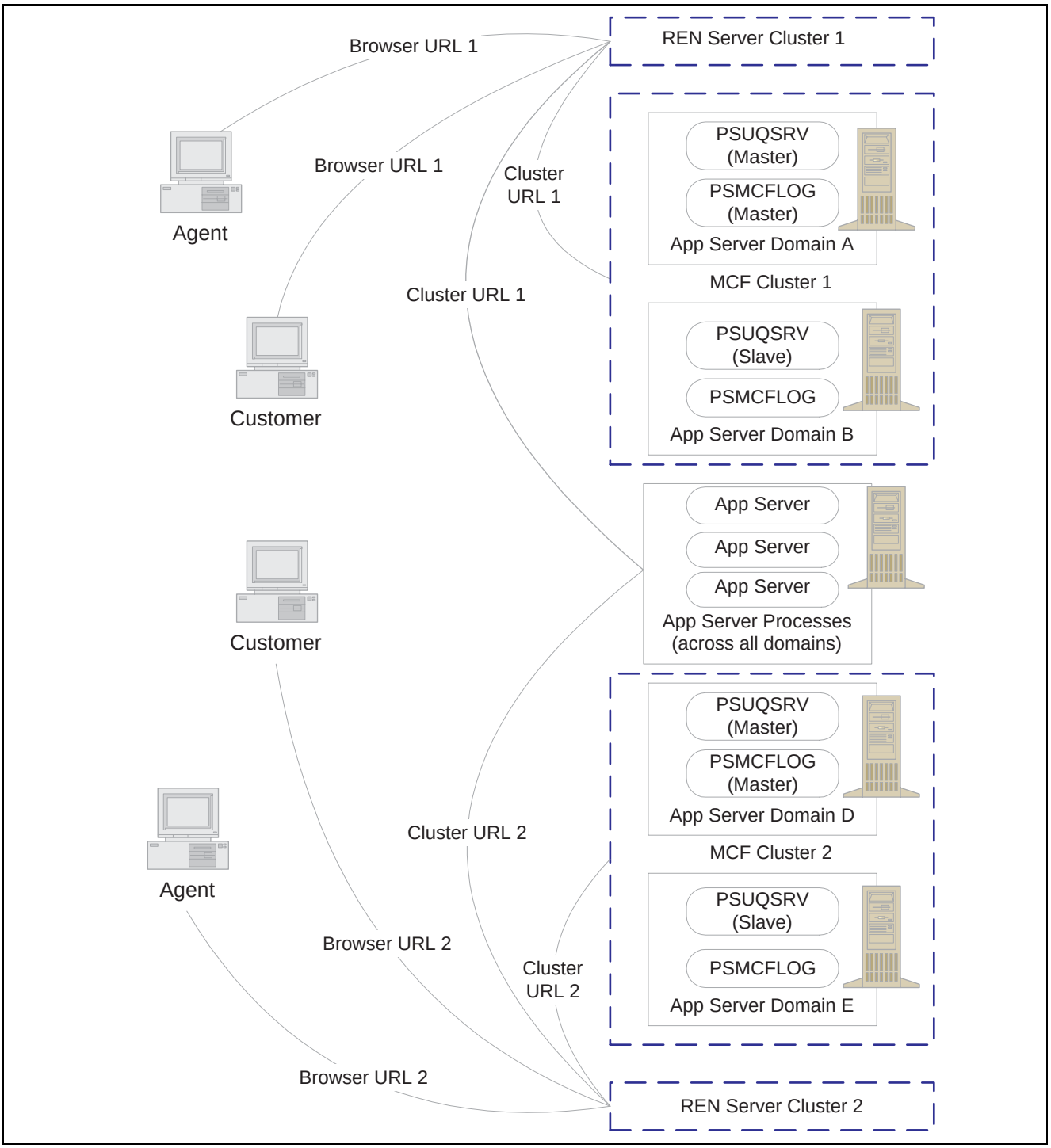
Copyright © 1988-2007, Oracle. All rights reserved.

PeopleSoft MCF Cluster Architecture

In a PeopleSoft system, all application server processes, including MCF servers and REN servers, belong to an application server domain. Each domain can have only one REN server process (PSRENSRV), one queue server process (PSUQSRV), and one MCF log server process (PSMCFLOG). Domains can be redundantly clustered to provide failover. Logical queues can be serviced by multiple clusters for scalability. Support for scalability and failover is integrated into the configuration process.

The following diagram illustrates MCF cluster architecture:

 Copyright © 1988-2007, Oracle. All rights reserved.



PeopleSoft MultiChannel Framework cluster architecture

Queue Server and Queue Server Failover

The queue server is a server process in the PeopleSoft application domain that routes email, chat, and generic tasks to the agent based on the agent properties, such as state and skill set and task routing properties, such as priority, language, and cost.

Copyright © 1988-2007, Oracle. All rights reserved.

The queue server process (PSUQSRV) is a BEA Tuxedo-managed server with a standard PeopleSoft database connection. Each queue server process is the central routing point for one or more physical queues. The queue server maintains state information for work requests, work in progress, agent availability, and agent workload. Queue server state is written to database records, except for the assignment of chat to agents.

The queue server can recover from a crash because most of its state is written to the database. When a queue server reboots, it checks the database and loads state information for open work tasks. The queue server connects to its REN server and issues restart queries to each console so that it can rebuild agent assignment information, which may have changed while the queue server was down.

Although a single queue server can recover state after recovering from software failure, this does not guard against hardware failure. Multiple queue server processes running on multiple host machines and configured in a cluster provide failover for hardware failure. Unlike the REN server cluster, the clustered queue servers operate as one master and many slaves. The master handles all routing decisions, while slave processes monitor the master and step in only if it fails. Any rebooted queue server rejoins the cluster in a slave role. Any slave that is promoted to master loads state from the database and issues queries to consoles as if it were the only process in the cluster.

Each queue server process follows a fixed procedure to ensure that the cluster has at most one functioning master. Database locks eliminate possible race conditions, and the master periodically writes a timestamp to indicate its health. The masterinterval parameter controls the frequency at which the master process must update the timestamp in the cluster table. The masterinterval parameter corresponds to the maximum time after a master queue server fails before another queue server process takes over. Minimizing this value provides rapid failover response time but also requires frequent database updates.

See Chapter 7, “Configuring PeopleSoft MCF Queues and Tasks,” Tuning Cluster Parameters, page 94.

Each queue server must be part of an MCF cluster, and each MCF cluster must include at least one queue server. An MCF cluster of only one queue server provides no redundancy against hardware failure.

Create a queue server that starts when an application domain is started by selecting MCF Servers from the quick-configure menu during application domain configuration.

In summary, configure queue servers to provide hardware failover. Each queue server is part of an MCF cluster. To support hardware failover, distribute master and slave queue servers over multiple hosts. Every queue server in an MCF cluster communicates with the same REN server ID. Therefore, REN server failover is also crucial.

The first queue server that places a valid master entry for itself in the cluster table becomes the master queue server. In most cases, the master queue server is the first queue server started. No configuration parameter exists to designate master or slave queue server within a cluster.

After an MCF cluster’s master queue server is established, all other cluster members become slave queue servers. If the slave queue servers within a cluster detect a failure of the master queue server, the remaining slave servers compete to become the master queue server. If the master queue server reboots before a slave takes over, the master queue server also competes. No configuration parameter exists to designate priority among slave servers.

See Chapter 5, “Configuring REN Servers,” page 55.

Logical Queues and Physical Queues

PeopleSoft MultiChannel Framework enables the configuration of both logical and physical queues.

A logical queue is an application-level queue that receives work requests (tasks) relating to an application area, such as chat requests regarding sales information, and routes them to agents that are capable of handling the work. For example, you might configure a logical queue called SALES for sales inquiries and another called SUPPORT for support issues.

 Copyright © 1988-2007, Oracle. All rights reserved.

Logical queues can be partitioned into physical queues for scalability. A physical queue is managed by a single MCF cluster. For scalability, the tasks that are enqueued on a logical queue are distributed by the framework among all available physical queues. For example, the SALES queue could be serviced by four MCF clusters across four physical queues: SALES1, SALES2, SALES3, and SALES4.

Each agent can be assigned to only one physical queue within each logical queue. Each agent can be assigned to multiple logical queues.

PeopleSoft MCF Log Server and Log Server Failover

The MCF log server (PSMCFLOG) is a BEA Tuxedo–managed server that is similar to the queue server. Each MCF log server receives events that are sent by a REN server and is responsible for writing MCF events to the database.

The MCF log server logs events to PS_MCFUQEVENTLOG. By default, the log server does not log periodic state information broadcasts from the queue server to the MultiChannel Console. If you need to log these events, configure logging on the Cluster Tuning page. You can also configure the log server to log the contents of chat sessions. Chat session logging is deactivated by default. Logged chats are stored in PS_MCFCHATLOG.

If the MCF log server crashes, it resumes functioning immediately after restarting. When the first slave log server detects a failed master, it takes over as the master log server for the cluster. The new master log server again receives all base topics, but it does not log chat sessions that started or continued during the time that the original master log server failed. The new master log server does not log per-agent events for agents that were signed in at, or during, the time of the failure.

An MCF log server is created along with a queue server when you enable MCF servers during application server domain configuration. No specific log server configuration is available during domain configuration.

Queue Server Scalability

PeopleSoft MultiChannel Framework is scalable to support large capacity call centers or other large organizations. The basic strategy is to divide the workload by spreading it over several MCF clusters. This is accomplished by creating multiple physical queues for each logical queue and spreading the management responsibility for each physical queue to separate queue server processes, preferably on multiple host machines. This technique should not be confused with failover protection, which also adds processes and machines. In failover, the added processes are clustered together and do not provide performance improvement.

Organize applications using PeopleSoft MultiChannel Framework around logical queues (for example, SALES queue and SUPPORT queue). Incoming work tasks are sent to a logical queue. PeopleSoft MultiChannel Framework then assigns the task to one of the corresponding physical queues. This assignment is random across the queues. The load across the servers is balanced by servicing only one physical queue per logical queue by single MCF cluster. balances load across the servers.

For example, a logical SUPPORT queue might be split into physical SUPPORT1 and SUPPORT2 queues such that work requests are randomly distributed between the two physical queues. Half the agents receive from one queue and half from the other. This splits the workload evenly between the two queue server processes, while still presenting one logical SUPPORT queue to the application.

Recommended Configurations

Consider the following configuration options to ensure maximum reliability and scalability of your PeopleSoft MultiChannel Framework installation:

- Configure multiple MCF servers in a cluster across multiple host machines.

Copyright © 1988-2007, Oracle. All rights reserved.

This provides protection against single-point failures.

Each MCF cluster requires a REN server cluster. Configuring multiple REN server clusters is functionally the same as configuring multiple MCF clusters for scalability. Inside a REN server cluster, configuring multiple REN servers is functionally the same as multiple queue servers for failover.

See Chapter 6, "Configuring PeopleSoft MCF Servers and Clusters," Configuring PeopleSoft MCF Clusters, page 83.

- Use REN server clusters only for failover.

  REN server clusters do not enhance performance.

- Split logical queues into more than one physical queue if more work is required on that queue than a single process or machine can handle.

- If an application server domain is likely to be restarted regularly for reasons that are not related to PeopleSoft MultiChannel Framework, configure PeopleSoft MultiChannel Framework in a separate domain.

  Regular restarting of MCF servers affects performance because the MCF servers must recover state when they are recycled or when a slave takes over from a master server.

Configuring PeopleSoft MCF Clusters

An MCF cluster is a group of multiple MCF-enabled application server domains in which all queue servers and log servers communicate with the same REN server cluster. Only one queue server and one log server in an MCF cluster are active at any one time. These are called the masters. The rest are dormant and redundant, and are called the slaves. If a master drops out of the cluster for any reason, the slaves elect a new master to take its place.

This section provides an overview of MCF cluster configuration and discusses how to configure MCF clusters.

Understanding PeopleSoft MCF Cluster Configuration

Each MCF cluster includes a minimum of one queue server and one log server communicating with one REN server. An MCF cluster is typically identified by the ID of the REN server cluster serving it. No configuration limit is placed on the maximum number of queue servers in an MCF cluster.

In MCF architecture, a chat client initiates contact with an agent through the designated external (browser) URL for a REN server. This URL can point to a load balancer, switch, reverse proxy server, or other hardware or software directing requests through a firewall. The external URL is also known as the browser URL, because it supports the MCF browser windows (agent chat, customer chat, and MultiChannel Console).

For communication of queue servers, log servers, and application servers with REN servers, for example, handling email requests behind a firewall, you can use an internal URL. Specify both internal (cluster) and external (browser) REN server URLs during MCF cluster configuration.

Note. If no security is implemented, the browser and cluster URLs may be the same.

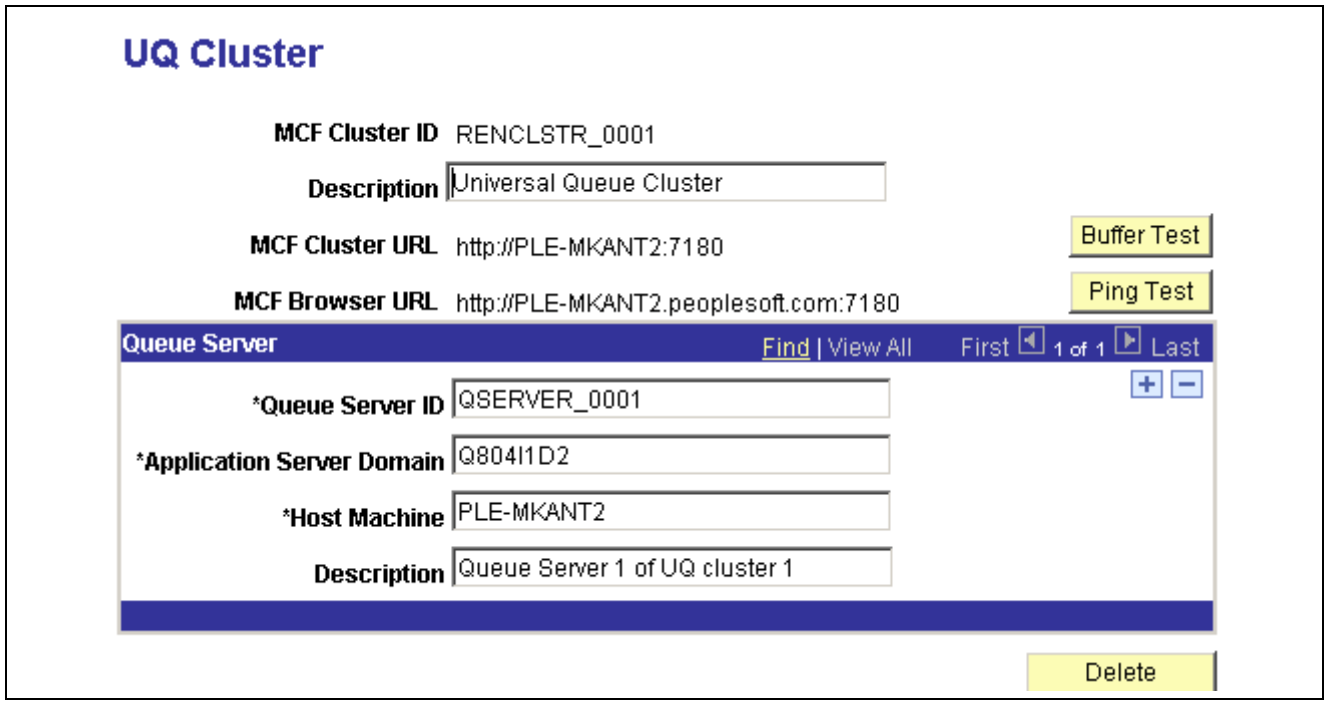
 Copyright © 1988-2007, Oracle. All rights reserved.

Page Used to Configure PeopleSoft MCF Clusters

Page Name	Object Name	Navigation	Usage
MCF Cluster	MCF_UQCLUSTER	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Cluster	Configure a cluster of queue servers.

Configuring PeopleSoft MCF Clusters

Access the UQ Cluster page.



UQ Cluster page

MCF Cluster ID Displays the MCF cluster Id.

MCF Cluster URL Displays the URL for the REN server that serves this cluster. This is the URL that is used by internal processes. If the MCF cluster is served by a REN server cluster, the cluster URL is that of the switch or load balancer in front of the clustered REN servers.

Buffer Test Click to initiate a test of the REN server's ability to break up and send a large file using multiple internal buffers.

The buffer test does not depend on specified domain names, so you can use it to verify that RENSRV is running on the network.

MCF Browser URL Displays the URL for a REN server cluster that serves this MCF cluster for external clients and for agent chat. The browser URL may be different from the cluster URL, which should not have to go through any firewall, reverse proxy server, or other outward-facing security barrier.

Ping Test Click to initiate a test of the REN server that is specified in the URL fields. Failure may indicate that a URL is incorrectly specified.

Copyright © 1988-2007, Oracle. All rights reserved.

Delete	Click to delete the entire MCF cluster. No active agents or tasks should be on the cluster.

Note. If the cluster's queue server configuration is changed, changes to the actual application server domains must be made manually using PSADMIN. For example, if a cluster member (queue server) is removed, the affected application server domain must be shut down and reconfigured (set the MCF Servers field to *No*) using PSADMIN. If the cluster URL is changed, all associated queue server domains must be shut down and rebooted.

Queue Server

An MCF cluster can consist of a primary queue server and any number of backup servers.

Each cluster requires a minimum of one queue server. The primary queue server is the first queue server started, and the remaining queue servers are backups. If the primary queue server fails, the system determines the subsequent primary queue server among the backups.

You can add a queue server to a cluster by adding a new row. Before removing a queue server, ensure that it is not the master, then shut down its domain. Then click Delete (the minus sign).

If a domain is started with a queue server that does not belong to a cluster, the universal queue server and MCF log server poll the MCF configuration tables indefinitely until the queue server is assigned to a cluster.

Queue Server ID	Enter a unique identifier for each queue server to identify its entries in the database control tables. The log server process that is paired with this queue server uses this same ID to identify its entry in the log cluster table.
Application Server Domain	Enter the application server domain of which this queue server is a member.
Host Machine	Enter the name of the application server host machine.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 7

Configuring PeopleSoft MCF Queues and Tasks

This chapter discusses how to:

- Define queues.
- Configure tasks.
- View the cluster summary .
- Tune cluster parameters.
- Notify clusters of changed parameters.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference, "Universal Queue Classes"

Defining Queues

To define queues, use the MCF Queue (MCF_Q_CONFIG_CMP) component.

This section discusses how to:

- Define queues.
- Define chat responses.
- Define static push URLs.

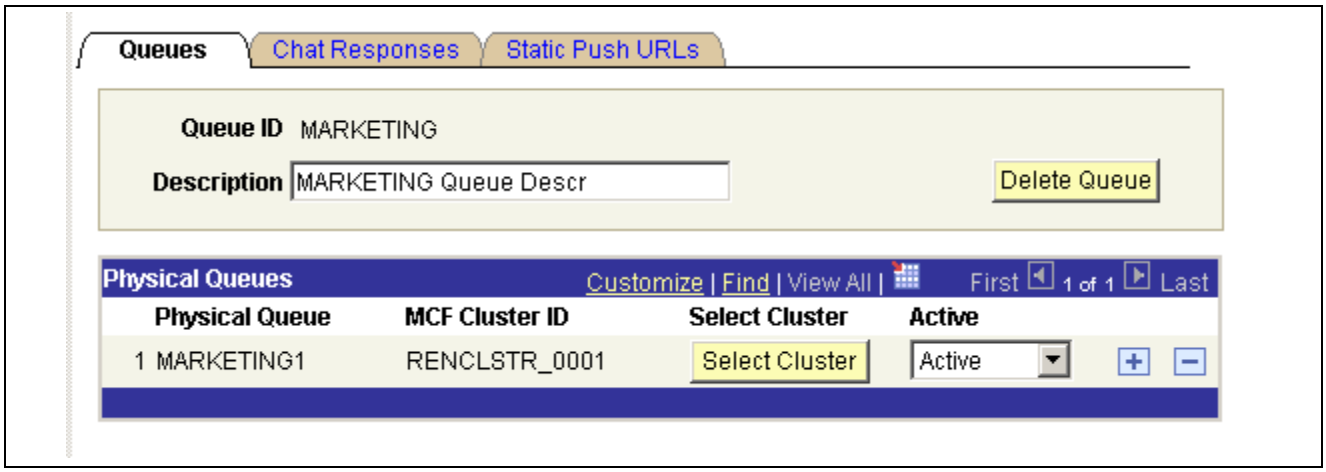
Copyright © 1988-2007, Oracle. All rights reserved.

Pages Used to Define Queues

Page Name	Object Name	Navigation	Usage
Queues	MCF_QUEUE_PG	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Queue, Queue	Define a queue.
Chat Responses	MCFSYSMSG_PG	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Queue, Chat Responses	Define messages that every agent can use.
Static Push URLs	MCFQUEUEURL_PG	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Queue, Static Push URLs	Define URLs that each agent can push to a client.

Defining Queues

Access the Queues page.



Queues page

Queue ID Queue IDs must be alphanumeric, cannot end in a numeral, but can include underscore characters.

Delete Queue Click to remove this queue.

Deleting a logical queue means that no work or agents can be assigned to the queue, and the queue is removed from all agents' available queues.

You can delete a logical queue only if all of its constituent physical queues are inactive and have no tasks. Verify that no application code assigns tasks to a queue before deleting the queue. All agents that are assigned to the child physical queue will receive a message notifying them to sign out of their MultiChannel Consoles when the logical queue is deleted.

Physical Queue Identifies that part of a logical queue that is serviced by the selected MCF cluster.

Copyright © 1988-2007, Oracle. All rights reserved.

Physical queue IDs always end in a number, for example, SALES2. The physical queue identifier automatically increments by one for each physical queue that is added.

Physical queue IDs are automatically generated. The maximum number of physical queues is nine.

MCF Cluster ID Identifies the MCF cluster that services this physical queue. Click Select Cluster to choose from configured MCF clusters.

Each MCF cluster can service only one physical queue per logical queue. For example, an MCF cluster could service physical queue SALES1 or SALES2, but not both.

An MCF cluster can service multiple physical queues belonging to different logical queues. For example, an MCF cluster could service physical queues SALES1, MARKETING2, and COBOL1.

Select Cluster Click to select a cluster from a list of available MCF clusters.

The selected MCF cluster will service this physical queue. The primary queue server in this cluster will manage tasks and agents that are assigned to this physical queue.

Active Select *Active* or *Inactive* from the drop-down list box.

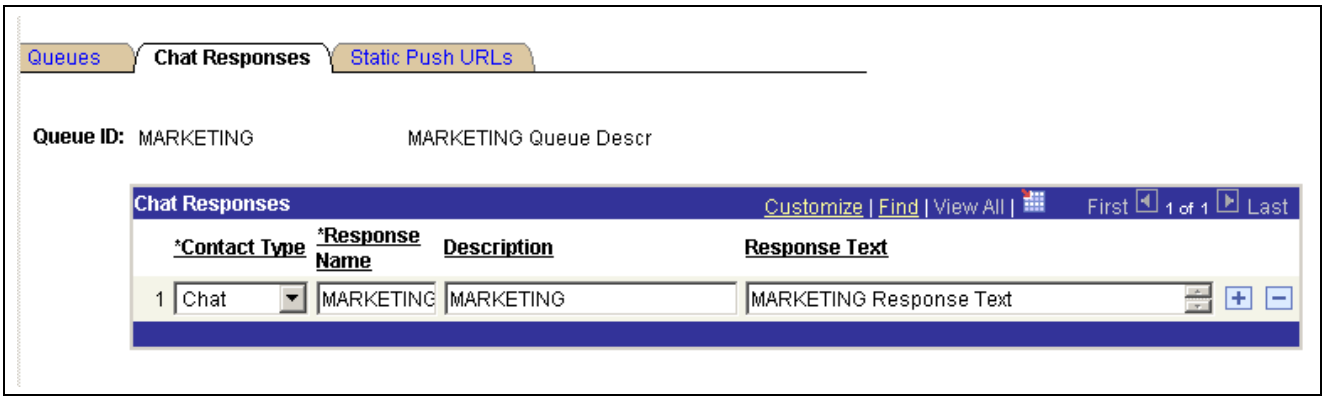
The queue server will not send new tasks to an inactive physical queue. Agents and existing tasks will remain on an inactive physical queue. The physical queue must be active to receive new queued tasks.

Only inactive physical queues can be deleted from a logical queue. Inactivate a physical queue and complete or transfer all assigned tasks before deleting the physical queue.

Active and inactive status support "follow-the-sun" practices. For example, SALES1 could be supported in the London office, and SALES2 could be supported in the San Francisco office when the London office is closed by activating and inactivating the appropriate queues.

Defining Chat Responses

Access the Chat Responses page.



Chat Responses page

Chat responses that are specified on this page are available to all agents who are signed in to this queue.

Copyright © 1988-2007, Oracle. All rights reserved.

Contact Type Select one of the following contact types:

- *Chat*
- *Email* (not currently supported)
- *Generic* (not currently supported)

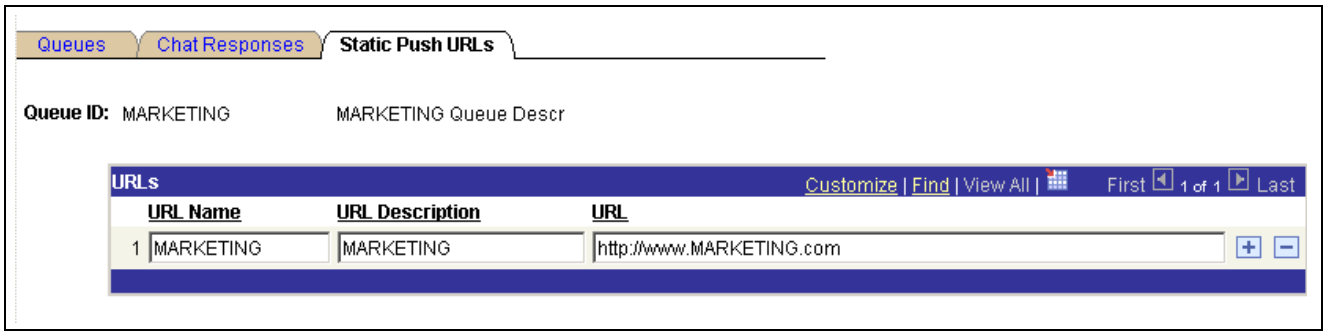
Response Name The response name appears in the agent's Template Messages drop-down list box for all agents who belong to a physical queue on this logical queue

All chat responses are downloaded when the agent launches the agent chat console by accepting a customer chat. Template messages are not available in collaborative chat.

Response Text The specified message appears in the client chat window when selected by the agent.

Defining Static Push URLs

Access the Static Push URLs page.



Static Push URLs page

Static URLs enable the agent to push a predefined URL to the customer, which appears in a pop-up window for the customer. The agent can edit these URLs in the chat window after selecting them.

URL Name Enter a name to identify the URL.

The URL name appears in the agent's Static URL drop-down list box for all agents that belong to this logical queue

URL Description Enter a description of the URL.

This description appears only on this page to further describe this URL or, for example, its reason for inclusion.

URL Enter the queue push URL.

The URL must include the opening http:// and any required parameters.

All static URLs that are defined for the queue are downloaded when the agent launches the agent chat console by accepting a customer chat. Static URLs are not available in collaborative chat.

If you send a static push URL that is a PeopleSoft Pure Internet Architecture URL, be sure that the recipient has permissions to access that portal, node, or page.

Copyright © 1988-2007, Oracle. All rights reserved.

Configuring Tasks

To configure tasks, use the MCF Task (MCF_TASKCFG_CMP) component.

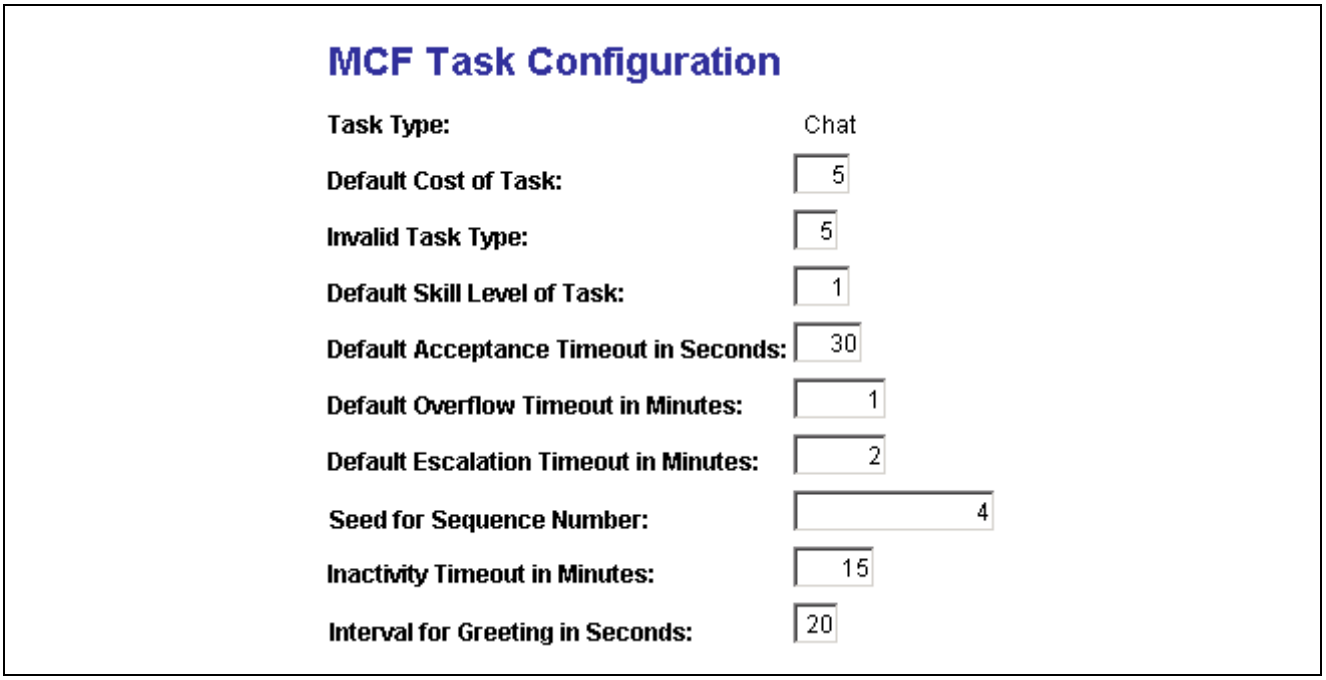
This section discusses how to configure tasks.

Page Used to Configure Tasks

Page Name	Object Name	Navigation	Usage
MCF Task Configuration	MCF_TASKCFG_PG	Use one of the following navigation paths: PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Tasks PeopleTools, Multichannel Framework, Third-Party Configuration, Tasks	Configure tasks such as chat, email, voice, and generic alerts using the queue server or third-party routing system.

Configuring Tasks

Access the MCF Task Configuration page.



MCF Task Configuration page

Define values for different types of tasks that the queue server uses to assign tasks to appropriate agents, and to manage tasks that are not accepted or closed within configurable time limits.

Copyright © 1988-2007, Oracle. All rights reserved.

After you define or change values for a task, you must use Refresh Task Properties button on the Cluster Notify page to propagate the changes to clusters. If you are working with MCF tasks, use the Cluster Notify page (MCF_CL_NOTIFY_PG) by navigating to PeopleTools, Multichannel Framework, Universal Queue, Configuration, Notify Cluster. If you are configuring third-party tasks, use the third-party Cluster Notify page (MCFTP_CL_NOTIFY_PG) by navigating to PeopleTools, Multichannel Framework, Third-Party Configuration, Cluster Notify.

Default Cost of Task

Enter the cost of the task.

Cost is a measure of the workload that each task places on an agent. The cost of a task is an estimate of the task's expected complexity and of the time that is required to resolve the task. The minimum value is *0,* and no maximum value exists.

The costs of tasks that are assigned to an agent are added up and evaluated against the maximum workload for each agent to determine whether the agent can receive additional tasks. For example, if an agent has a maximum workload of 100, and the default cost of a chat is 20, the agent can manage five concurrent chat sessions, assuming the default cost is not overridden in the InitChat() built-in function call and that no other task types have been assigned.

Note. Although priority has no effect on voice tasks (which are not queued), voice task cost is included in calculating an agent's workload.

Default costs are:

- Chat: *5*
- Email: *2*
- Generic: *1*
- Voice: *10*

Default Priority of Task

Enter the priority of this task. A higher value means a higher priority. Tasks are ordered on a physical queue based on their assigned priority.

The minimum value is *0,* and no maximum value exists.

A queue server gives precedence to a task of higher-priority value over a task of lower-priority value when looking for an agent to assign the task to. This means that the queue server will always assign a task of priority 100 to a qualified available agent before it looks for an agent for a task of priority 10. If two tasks have the same priority, they will be assigned in the order of their enqueue time.

The value that is specified here can be overridden in the EnQueue() or InitChat() built-in function call.

Note. Priority has no effect on voice tasks, which are not queued; however, voice task cost is included in calculating an agent's workload.

Default priorities are:

- Chat: *5*
- Email: *2*
- Generic: *1*

 Copyright © 1988-2007, Oracle. All rights reserved.

- Voice: *10*

Default Skill Level of Task

Enter the minimum agent skill that is required to handle this task.

The queue server assigns this task type to an available agent with the lowest skill level on that queue that is greater than or equal to the skill level that is required by the task.

The minimum value is *0,* and is no maximum value exists.

The value that is specified here can be overridden in the EnQueue() or InitChat() built-in function call.

Default skill levels are:

- Chat: *1*
- Email: *1*
- Generic:*1*
- Voice : *2*

Note. Only the third-party routing server supports voice channel. The queue server does not route voice tasks.

Default Acceptance Timeout in Seconds

Specify the period of time that an agent has to accept an assigned task (to click the flashing icon on the MultiChannel Console). If the task is not accepted within this time, the task is re-enqueued for assignment to another agent.

The queue server uses an algorithm to minimize reassignment of tasks which previously timed out to the same agent.

The value that is specified here can be overridden in the EnQueue() or InitChat() built-in function call.

Default acceptance timeouts are:

- Chat: *30*
- Email: *30*
- Generic: *30*
- Voice : *10*

Note. Only the third-party routing server supports voice channel. The queue server does not route voice tasks.

Default Overflow Timeout in Minutes

Specify the overflow timeout.

The overflow timeout is the time period that a queue server has to find an agent who accepts a task (click flashing icon on the MultiChannel console). If the task is not accepted within this time, the task is removed from the queue and placed in the overflow table. This table can be managed from the Overflow Administration page.

The value that is specified here can be overridden in the EnQueue() or InitChat() built-in function call.

Default overflow timeouts are:

Copyright © 1988-2007, Oracle. All rights reserved.

- Chat: *2*
- Email: *120*
- Generic: *120*
- Voice: *1*

Note. Only the third-party routing server supports voice channel. The queue server uses CTI to route voice tasks.

Default Escalation Timeout in Minutes

Specify the default escalation timeout.

The escalation timeout is the time period within which a task must be closed. If the task is not closed within this time, the task is removed from the queue and from the agent's accepted task list (that is, the task is unassigned) and the task is placed in the escalation table. This table can be managed from the Escalation Administration page.

Escalation timeout is valid on a chat session only after it is accepted by an agent, and has no effect on voice tasks (CTI).

The value that is specified here can be overridden in the EnQueue() or InitChat() built-in function call.

Default escalation timeouts are:

- Chat: *10*
- Email: *480*
- Generic: *480*
- Voice : *2*

Note. Only the third-party routing server supports voice channel. The queue server uses CTI to route voice tasks.

Seed for Sequence Number

This field displays the current cumulative count of tasks of this type that are enqueued. Do not modify this value until it has reached its upper limit of 2,147,483,647.

This value does not apply for the voice task type.

Inactivity Timeout in Minutes

Specify the inactivity timeout

Inactivity timeout applies to chat only. The inactivity timeout is the time period within which an agent or customer must participate in a chat. If the chat session is dormant for more than this time, the chat is terminated.

The default value is *15* minutes.

Interval for Greeting in Seconds

Specify the interval, in seconds, over which the initial greeting appears in a customer chat window while the system searches for an available agent.

Note. All task parameters are delivered with sample values. Determine a range for these values that is appropriate to your business requirements. For example, task cost could vary over a range of 1 to 100 instead of 1 to 10.

Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 7, “Configuring PeopleSoft MCF Queues and Tasks,” Notifying Clusters of Changed Parameters, page 101

Chapter 12, “Configuring PeopleSoft MCF for Third-Party Routing Systems,” Notifying PeopleSoft MCF Clusters of Changed Parameters for a Third Party, page 187

Chapter 8, “Configuring PeopleSoft MCF Agents,” Creating Agents, page 103

Viewing the Cluster Summary

To view the Cluster Summary page, use the Cluster Summary (MCF_RSERV_CFG_CMP) component.

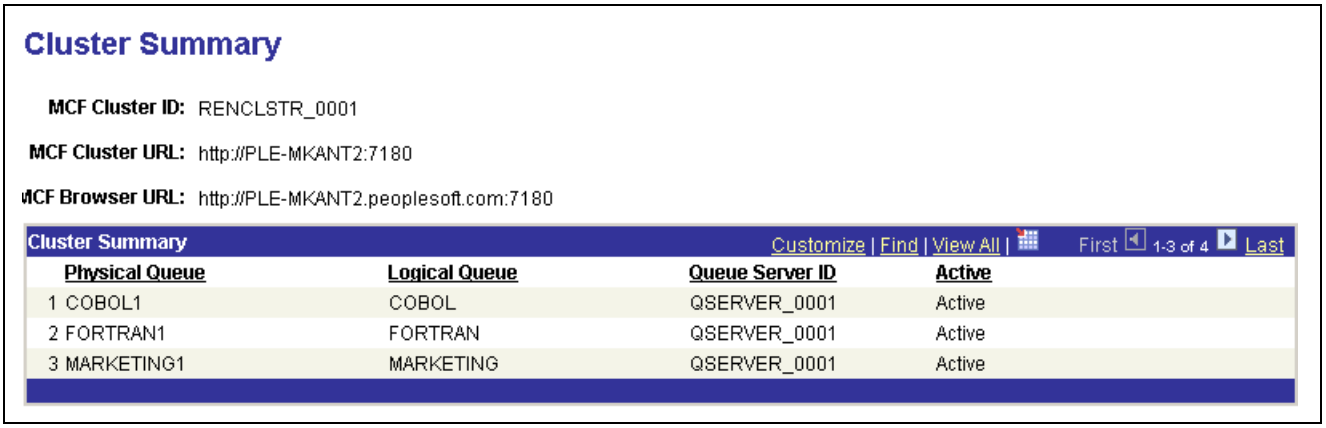
This section discusses how to view the Cluster Summary page.

Page Used to View the Cluster Summary

Page Name	Object Name	Navigation	Usage
Cluster Summary	MCF_RENSRV_PG	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Cluster Summary	View summary information about MCF clusters.

Viewing the Cluster Summary

Access the Cluster Summary page.



Cluster Summary page

The Cluster Summary page displays the associated MCF cluster URLs and queue details for the selected MCF cluster. The information cannot be changed from this page.

Tuning Cluster Parameters

To tune cluster parameters, use the Cluster Tuning (MCF_SYSTEM_NV_CMP) component.

Copyright © 1988-2007, Oracle. All rights reserved.

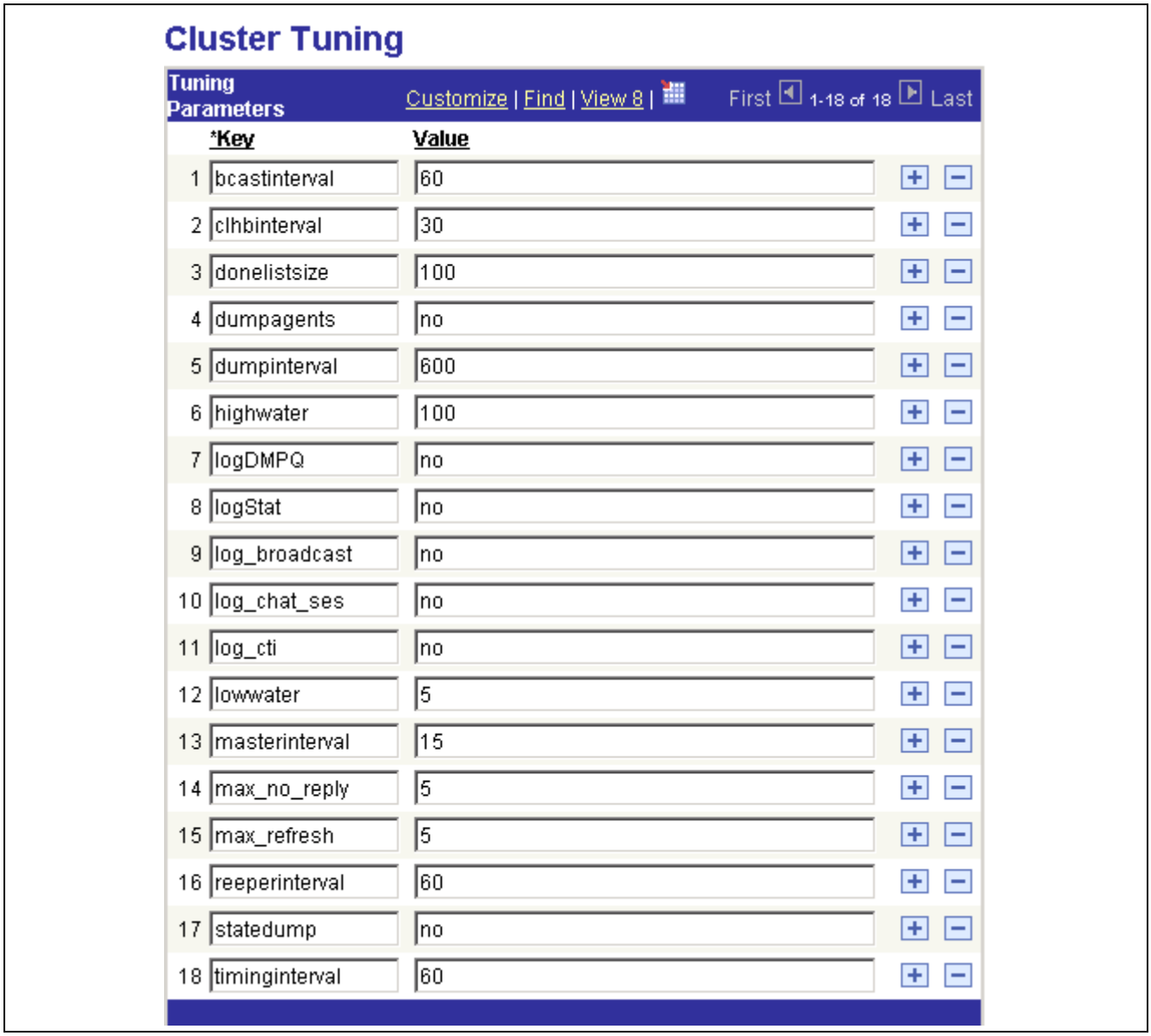
This section discusses how to tune cluster parameters.

Page Used to Tune Clusters

Page Name	Object Name	Navigation	Usage
Cluster Tuning	MCF_SYSTEM_KV_PG	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Cluster Tuning	Modify MCF cluster parameters to tune performance.

Tuning Cluster Parameters

Access the Cluster Tuning page.



Cluster Tuning page

Copyright © 1988-2007, Oracle. All rights reserved.

Use the Cluster Tuning page to set MCF cluster parameters to optimize performance or enable logging.

If you make changes to a cluster parameter, you must use the Notify Cluster page to propagate the changes.

See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.

The following table lists the cluster tuning parameters you can modify and describes the default values and usage of each.

Key	Default value	Usage
bcastinterval	*60*	The interval, in seconds, after which the number of unassigned tasks per physical queue and the number of agents that are logged into each physical queue are broadcast to the MultiChannel Consoles for display next to the queue names. A smaller value provides more accurate queue statistics, but increases the load on the queue server and REN server. A larger value decreases queue server and REN server load, but also decreases statistical accuracy. The bcastinterval value also determines how frequently onStat2 event statistics are calculated. A smaller value provides updated statistics more frequently. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101; Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the Monitor Queues Page and Sample Monitor - Queue Statistics Page, page 172 and Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the Monitor Agents Page and Sample Monitor - Agent States Page, page 170.

Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
clhbinterval	*30*	The interval, in seconds, in which the queue server expects to receive a heartbeat from a connected JSMCAPI client. If the queue server receives no heartbeat during this interval, the queue server stops the client session. For a queue server, to avoid a session timeout for the client when CPU usage of the machine on which the client is running is high, increase the heartbeat interval of the client. **Note.** For third-party server, increase the heartbeat interval on the third-party server side to avoid a session timeout. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using and Demonstrating JSMCAPI, page 155.
donelistsize	*100*	The number of completed tasks that are stored in the list that is used to calculate average task duration. Configure the donelistsize value depending on the task volume that is encountered and the interval over which you want to monitor tasks. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using and Demonstrating JSMCAPI, page 155.
dumpagents	*No*	Enter *Yes* if the status of agent activity should be written to the database during the periodic state dumps. Logging agent status increases queue server load, but provides information about agent performance. This is a task parameter. If you change the value of this parameter you must use the Refresh Task Properties button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing the Agent State Summary, page 118.

 Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
dumpinterval	*600*	The interval, in seconds, after which the queue state is written to the database. A smaller value increases load on the queue server, but provides more frequent statistics. A larger value decreases load on the queue server, but provides less frequent statistics. A value of less than one minute will significantly reduce queue server performance. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101; Chapter 9, "Administering Queues, Logs, and Tasks," Viewing the Queue Server State, page 117 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing the Queue State Summary, page 117.
highwater	*100*	The maximum number of persistent tasks that are retrieved from the database and cached in memory in the queue server. The highwater and lowwater mark values determine how often, and how many, persistent tasks should be read into memory. A higher value causes the queue server to retrieve more persistent tasks at one time, which results in less frequent access to the database, but also more tasks for the queue server to manage, slowing performance. A lower value speeds performance, but requires more frequent access to the database. If a large number of enqueued tasks cannot be routed by the queue server (for example, a call center handling a specific physical queue is offline), increase the highwater mark so that tasks that can be routed can fit into memory cache. This is a threshold parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.

Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
logDMPQ	*No*	Enter *Yes* if you want PSMCFLOG to log REN server event notifications resulting from bcastinterval broadcasts. Only the event is logged, not its contents. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Event Logs, page 123.
logStat	*No*	Enter *Yes* to log the statistics that are returned by the queue server for the onStat1 user and group events to the database. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using and Demonstrating JSMCAPI, page 155.
log_broadcast	*No*	Enter *Yes* to activate logging of the broadcast messages that are sent. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Broadcast Logs, page 120.
log_chat_ses	*No*	Enter *Yes* to activate logging of the contents of chat sessions. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Chat Logs, page 121.

Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
log_cti	*No*	Select *Yes* to activate logging of CTI events. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 3, "Configuring PeopleSoft CTI," CTI Event Logging, page 40.
lowwater	*5*	The minimum number of persistent tasks that are cached in memory in the queue server. When the lowwater value is reached, the queue server retrieves another batch of persistent tasks, up to the highwater value. The highwater and lowwater mark values determine when, and how many, persistent tasks should be read into memory. A higher value requires the queue server to access the database more frequently. A lower value can cause the queue server to run out of persistent tasks before refreshing its queue. The lowwater value should be greater than or equal to the maximum number of agents that are logged onto any physical queue at one time. This is a threshold parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.

Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
masterinterval	*15*	The interval, in seconds, after which a cluster master updates its timestamp in its cluster tables. Slave clusters check the timestamp to determine whether the master cluster is still running. A lower value enables rapid discovery of a failed master server, but increases queue server overhead. A higher value reduces queue server overhead, but delays discovery of a failed master server. If only one queue server is configured for an MCF cluster, this value can be large. The masterinterval value also acts as a heartbeat interval for the master queue server connection to user consoles. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.
max_no_reply	*5*	Sets the maximum number of consecutive agent timeouts before the queue server automatically logs out the agent and sets the agent's console status as Assumed Unavailable. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.
max_refresh	*5*	Sets the maximum number of consecutive times that results are discarded when refreshing the task queue from the database if there is an intervening notification of new persistent tasks. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.

 Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
reeperinterval	*60*	The interval, in seconds, after which deleted tasks are cleared from memory in the queue server. A lower value increases queue server load but clears memory more frequently. A higher value decreases queue server load but clears memory less frequently. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.
statdump	*No*	Specify *Yes* to write queue server state to the database during the periodic state dumps. The state dump interval is set by the dumpinterval parameter. This is a task parameter. If you change the value of this parameter you must use the Refresh Task Properties button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101 and Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using and Demonstrating JSMCAPI, page 155.
timinginterval	*60*	The interval, in seconds, after which the database is checked for expired or overflowed persistent tasks. This parameter does not affect real-time tasks. A lower value increases queue server load but detects timed out tasks more quickly. A higher value decreases queue server load but detects timed out tasks less frequently. This is a timing parameter. If you change the value of this parameter you must use the Refresh Threshold/Timing Parameters button on the Cluster Notify page to notify clusters of the changed parameter. See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Notifying Clusters of Changed Parameters, page 101.

Notifying Clusters of Changed Parameters

To notify clusters of changed parameters, use the Cluster Notify (MCF_AD_NOTIFY_CMP) component.

Copyright © 1988-2007, Oracle. All rights reserved.

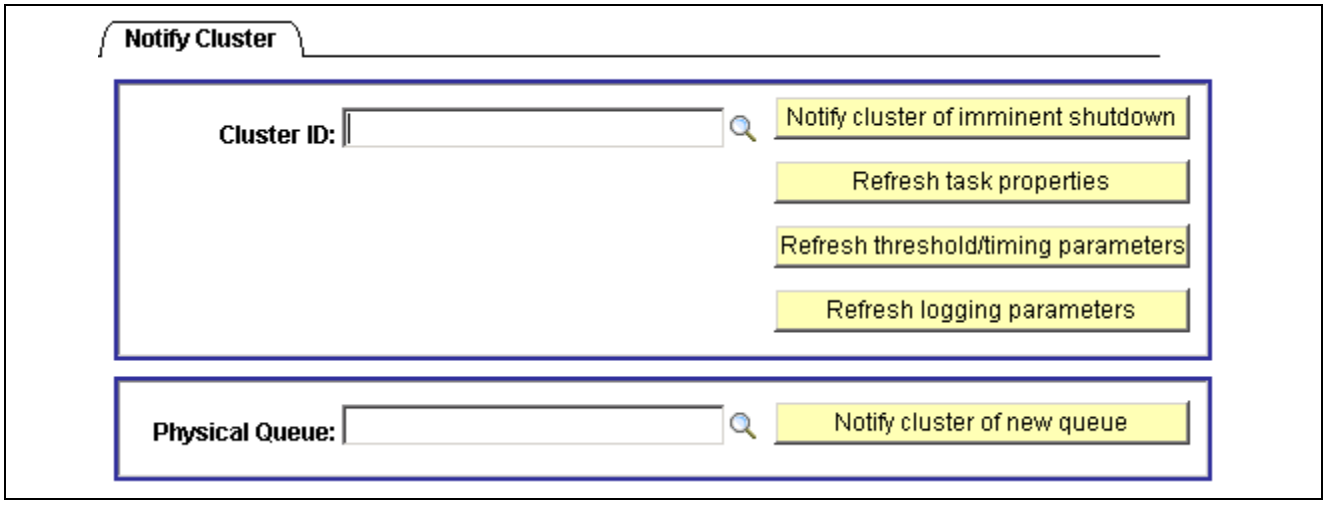
This section discusses how to notify clusters of changed parameters.

Page Used to Notify Clusters of Changed Parameters

Page Name	Object Name	Navigation	Usage
Notify Cluster	MCF_CL_NOTIFY_PG	PeopleTools, MultiChannel Framework, Universal Queue, Configuration, Cluster Notify	Notify a cluster of changes to parameters or configuration, or of shutdown.

Notifying Clusters of Changed Parameters

Access the Notify Cluster page.



Notify Cluster page

Use the Notify Cluster page to notify an MCF cluster of certain changes to its parameters or constituent queues, or that its application servers are being shut down.

For example, after changing MCF cluster parameters on the Cluster Tuning page, use the Notify Cluster page to refresh the tuning parameters.

Notify cluster of imminent shutdown Click to send a message to all agents who are logged on to the selected MCF cluster that they have been logged off.

Send this notification if the cluster's application servers are being shut down.

Refresh task properties Click to load task properties that have been changed on the Tasks page for the selected MCF cluster.

Refresh threshold/timing parameters Click to reload threshold and timing parameters that have changed on the Cluster Tuning page for the selected MCF cluster.

Refresh logging parameters Click to reload logging parameters that have changed on the Cluster Tuning page for the selected MCF cluster.

Notify cluster of new queue Click to notify the selected MCF cluster that the selected physical queue has been added.

Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 8

Configuring PeopleSoft MCF Agents

This chapter discusses how to:

- Define agents.
- Define optional agent characteristics.

Defining Agents

To define agents, use the MCF Agent (MCF_AGENT_CMP) component.

This section discusses how to:

- Create agents.
- Specify languages that an agent supports.
- Personalize an agent's presence.

The above three agent definition pages form the basis of agent configuration. Other parameters are optional.

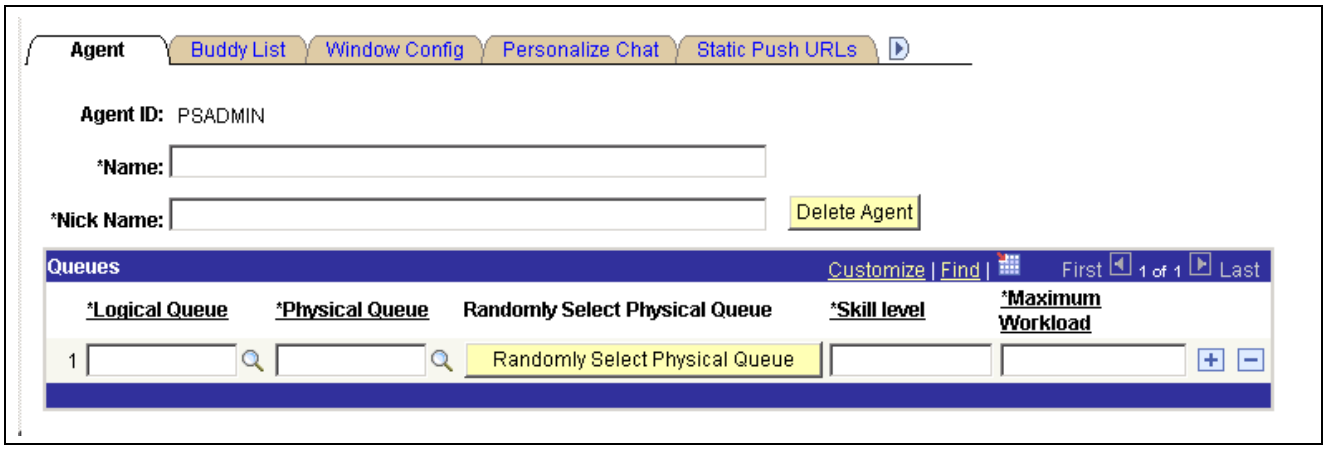
Pages Used to Define Agents

Page Name	Object Name	Navigation	Usage
Agent	MCF_AGENT_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Agent	Create agents. Specify name, skill level, maximum workload, and queues for each agent.
Languages	MCF_AGENTLANG_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Languages	Specify the languages that an agent is qualified to use. Specify at least one language.
Personalize Presence	MCFAGENTPRES_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Personalize Presence	Specify an agent's presence options.

Creating Agents

Access the Agent page.

Copyright © 1988-2007, Oracle. All rights reserved.



Agent page

Name

Enter the full name, in (lastname,firstname) format, of this agent.

The agent name appears in other agents' buddy lists.

Note. There is no space in between lastname, firstname.

Nick Name

Enter a short name for this agent.

The agent nickname identifies this agent in chat sessions and logs.

Delete Agent

An agent cannot be deleted if the agent still has accepted tasks on any queues to which the agent belongs. Before deleting an agent, ensure that the agent is logged off from all queues to which the agent is assigned.

Logical Queue ID

Enter the ID of a logical queue to which this agent is assigned.

Each agent can be assigned to more than one logical queue. An agent can log on to only one queue at a time from the MultiChannel Console.

Note. Do not overwrite the logical queue except when first creating an agent, as the agent's tasks may lose their assignments.

Physical Queue ID

Agents are randomly assigned to a physical queue when the logical queue is associated with the agent. An agent who services a logical queue logs on to a physical queue that is managed by a specific MCF cluster.

While an agent can service multiple logical queues, the agent can belong to only one physical queue per logical queue.

Note. Do not overwrite the physical queue except when first creating an agent, as this may orphan tasks. Use the Physical Queues Move Agent page.

Randomly Select Physical Queue

Click to assign another physical queue (within this logical queue) randomly. This will help to spread multiple agents evenly over available physical queues.

Skill Level

Select the skill level of this agent for the tasks that are assigned for this queue. This is a required field.

The agent is assigned only tasks requiring a skill level that is less than or equal to the skill level that is specified here. If more than one qualified agent is

Copyright © 1988-2007, Oracle. All rights reserved.

available to accept the task, the queue server gives preference to the agent with the lowest skill level.

Each agent can have a different skill level for each queue to which the agent is assigned.

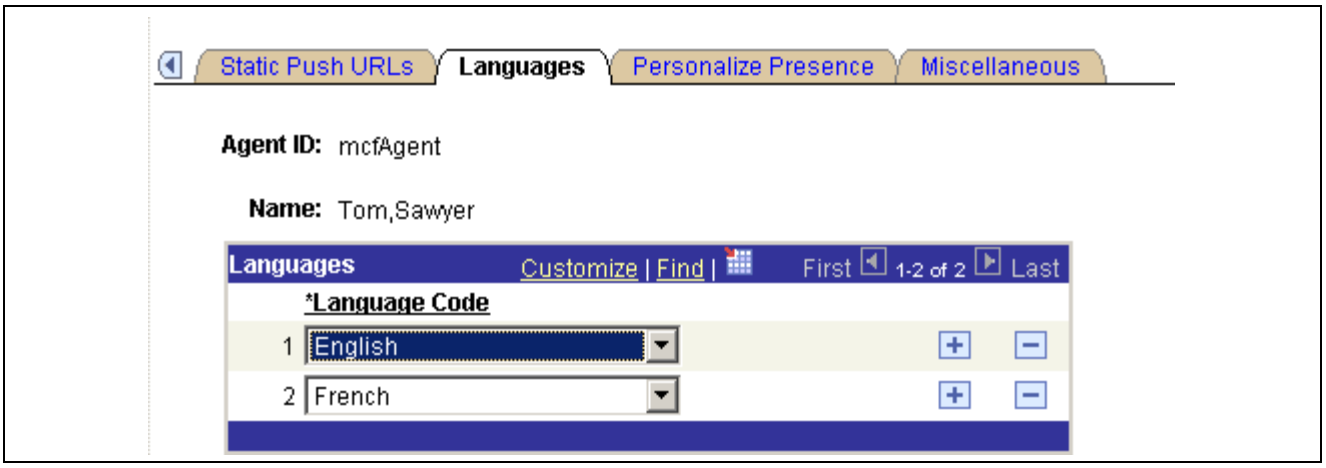
Maximum Workload Select the maximum load that this agent can be assigned before tasks are held or assigned to other agents. This is a required field.

The cost of each accepted task is added to the agent's current workload. A task is not assigned to an agent if its cost pushes the agent's current workload over the maximum.

Note. Do not delete a queue from an agent's list unless that agent has no open accepted tasks on that queue.

Specifying Languages That an Agent Supports

Access the Languages page.



Languages page

Specify the languages that each agent is qualified to support.

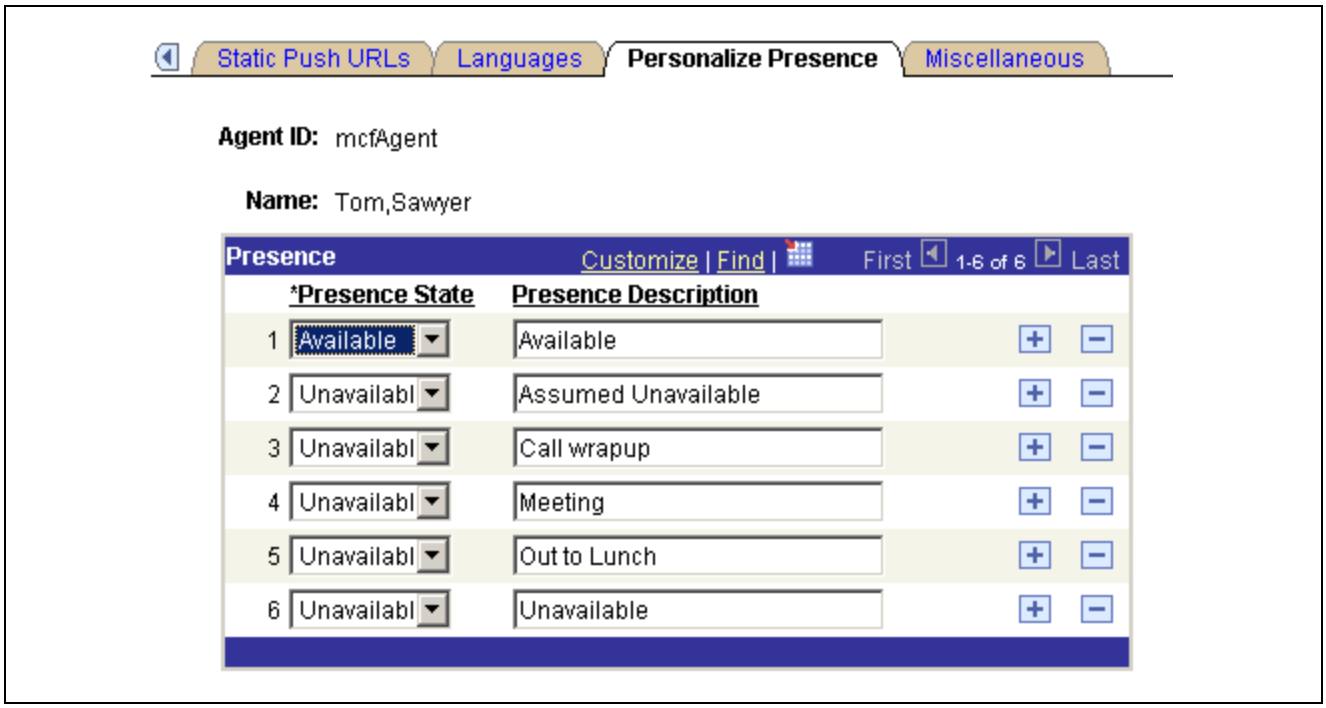
The agent is assigned only tasks that have been enqueued with a language code in the agent's language list. For the EnQueue() built-in function, the language code is specified as a parameter. For the InitChat() built-in function, the language code is determined by the user profile of the initiator.

If you do not specify a language code for a new agent, the default value is *English.*

Personalizing an Agent's Presence

Access the Personalize Presence page.

Copyright © 1988-2007, Oracle. All rights reserved.



Personalize Presence page

Each agent can configure the presence description that is displayed when the agent is available or unavailable. The queue server understands only the presence state, available or unavailable, but you can specify more specific presence descriptions when displaying or logging an agent's presence. For example, Lunch, Meeting, or Indisposed are unavailable states that can be used for tracking agent time and activity.

If you do not specify presence descriptions, default values are used.

Presence State Select *Available* or *Unavailable.*

Presence Description Enter a description for each agent state.

The description appears in logs of agent activity and when agent presence is displayed.

Note. There is only one description for *Available* state. For unavailable state, you can enter any presence description, such as tea break, coffee break, lunch, and so on.

Defining Optional Agent Characteristics

To define optional agent characteristics, use the MCF Agent (MCF_AGENT_CMP) component.

This section discusses how to:

- Set up buddy lists.
- Configure windows.
- Personalize chat.
- Specify agent-specific URLs.
- Specify miscellaneous parameters.

Copyright © 1988-2007, Oracle. All rights reserved.

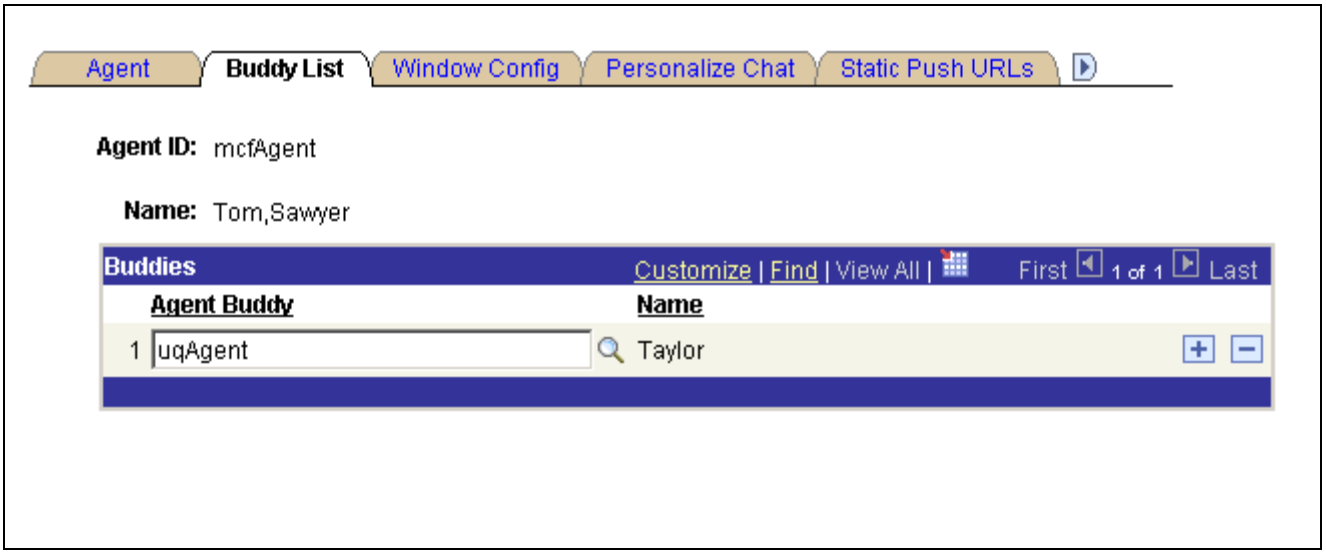
The agent configuration pages are considered optional because most do not have default values and can remain unconfigured without affecting an agent's ability to log on to a queue and accept tasks.

Pages Used to Define Optional Agent Characteristics

Page Name	Object Name	Navigation	Usage
Buddy List	MCFAGENTBUDDY_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Buddy List	Specify a list of other agents.
Window Config (window configuration)	MCFAGENTCUST_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Window Config	Configure the specified agent's window.
Personalize Chat	MCFAGENTMSG_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Agent Messages	Specify agent-specific messages, in addition to available system messages.
Static Push URLs	MCFAGENTURL_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Agent Push URLs	Specify agent-specific URLs, in addition to system push URLs.
Miscellaneous	MCF_AGENTMISC	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent, Miscellaneous	Specify miscellaneous behaviors.

Setting Up Buddy Lists

Access the Buddy List page.



Buddy List page

The agent's buddy list facilitates collaborative chat and chat conferencing.

Copyright © 1988-2007, Oracle. All rights reserved.

Agent Buddy Select another agent with whom this agent can have a chat session or can ask to conference onto another chat.

Each agent buddy must be logged on a physical queue on the same cluster to be able to chat. If two agents must be able to chat but they do not share a cluster, use the Physical Queue Move Agent page to move the agents onto physical queues on the same MCF cluster.

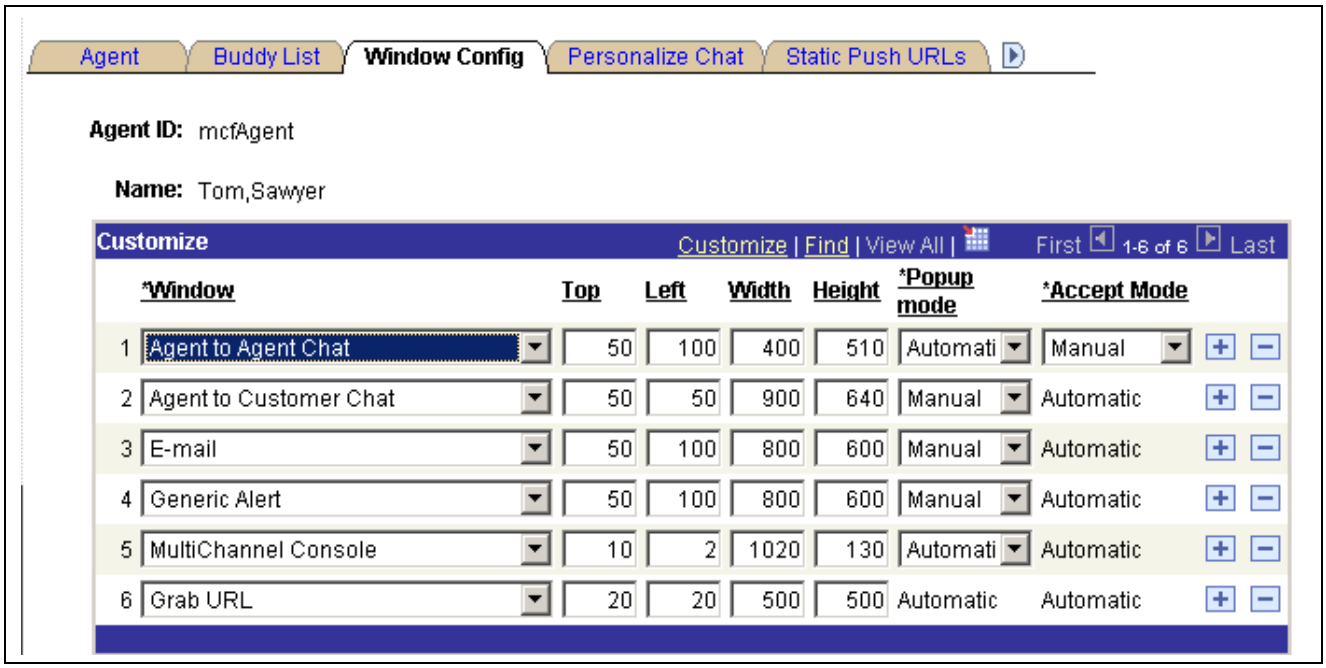
Agent buddies are listed with their login status in the buddy list on the multi-channel console.

Name Displays the buddy agent's nickname.

An agent's presence as shown in the buddy list on the MultiChannel Console or on the Invite Agent list on the chat console indicates the agent's availability for chat or conference.

Configuring Windows

Access the Window Config page.



Window Config page

Set the initial agent window placement and size by specifying parameters on this page. The windows may be resized and moved by an agent.

Window Select the window to which the specified configuration applies.

Select from:

- *Agent to Agent Chat*
- *Agent to Customer Chat*
- *E-mail*
- *Generic Alert*
- *Grab URL*

 Copyright © 1988-2007, Oracle. All rights reserved.

- *MultiChannel Console*

Top and **Left** — Enter the distance in pixels from the top and left edge of the screen when the window first appears.

Width and **Height** — Enter the width and height, in pixels, of the window when it first appears.

Popup Mode — Select from:

Automatic: The window appears automatically. For customer-initiated chat or tasks that are initiated from the EnQueue() built-in function, the task is automatically accepted as well. For agent-initiated chat, the agent can elect not to accept the task; in effect, the agent can preview the task. If this is the desired behavior, select *Manual* from the Accept Mode drop-down list box. If you want agent-to-agent tasks to function like customer-initiated tasks, select *Automatic* from the Accept Mode drop-down list box.

Manual: The window does not appear until the agent clicks the task on the agent MultiChannel Console. For customer-initiated chat or tasks that are initiated from the EnQueue() built-in function, clicking the task means that the task is automatically accepted as well. For agent-initiated chat, the behavior depends on the setting for the Accept Mode field.

Accept Mode — Select from:

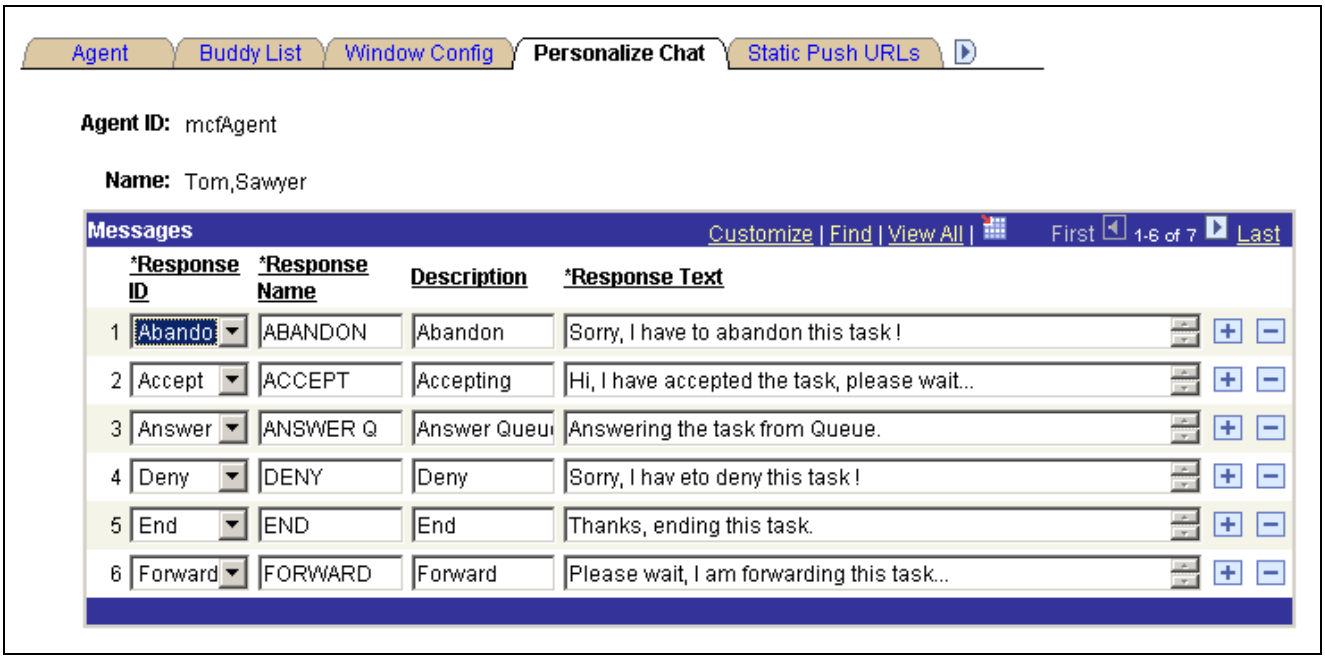
Automatic: Agent-to-agent chats are automatically accepted without requiring the agent to click the icon.

Manual: Agent-to-agent chats require the agent to click the icon.

Accept mode affects only collaborative chat.

Personalizing Chat

Access the Personalize Chat page.



Personalize Chat page

Copyright © 1988-2007, Oracle. All rights reserved.

An agent can create personalized responses in addition to the system responses that are defined for each queue.

Response ID Responses, except those that are identified by Other, are linked to specific events. These responses are always sent on these events from this agent. If an agent does not have a customized response for a specific event, the response is read from a default value that is set in the Message Catalog. The response text that is set here overrides the default text that is set in the Message Catalog.

Select from:

- *Abandon:* A chat is abandoned when a chat initiator closes the chat window before the chat is accepted by an agent. This message appears when the agent accepts the abandoned chat.
- *Accept:* This response is automatically sent to the chat initiator when an agent accepts a chat in response to a chat request that does not include a question.

  If the chat request includes a question, the agent's *Answer Question* text is sent in response instead of the *Accept* response.
- *Answer Question:* This response is automatically sent to the chat initiator when an agent accepts a chat in response to a chat request that includes a question.
- *Deny:* This response applies only to collaborative chat. If an agent elects not to accept a chat, this message is automatically sent to the chat initiator.
- *End:* If either party quits a chat after the chat is accepted, this message is displayed from the agent.
- *Forward:* If the agent forwards a chat session to another queue, this message is sent to the customer.
- *Other:* These messages are never automatically sent in a chat session. Their message names appear in the Template Messages drop-down list box on the agent chat page. These messages are appended to the template messages (chat responses) that are defined for the queue.

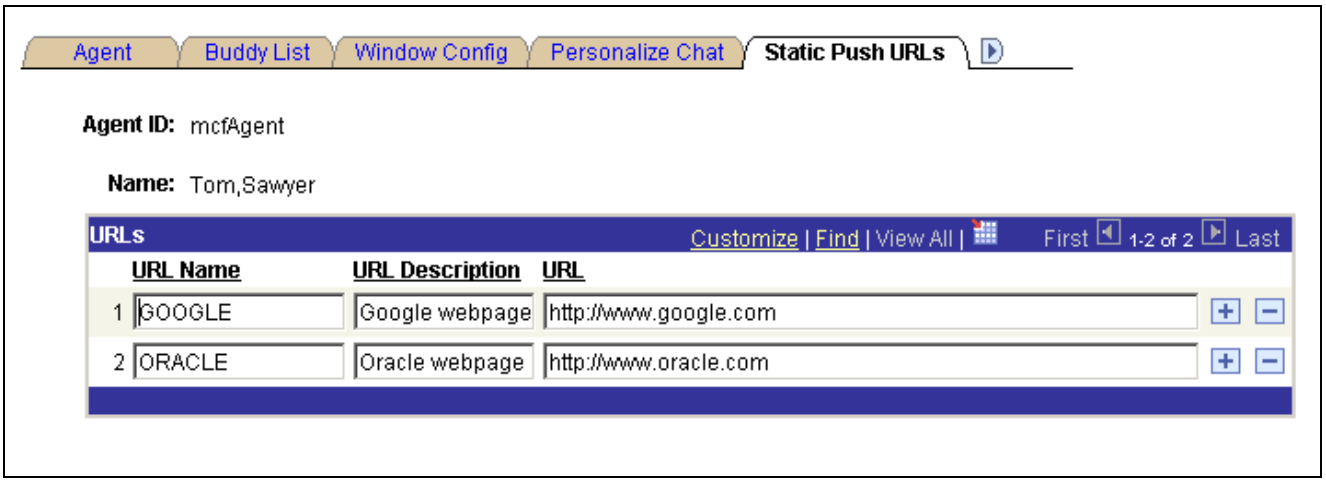
Response Name This name appears in the agent's template response drop-down list box.

Response Text Enter the response text to appear in the chat window.

Specifying Agent-Specific URLs

Access the Static Push URLs page.

 Copyright © 1988-2007, Oracle. All rights reserved.



Static Push URLs page

This page defines URLs that this agent can send to a client browser. These URLs are in addition to the URLs that are defined in the queue configuration page.

URL Name The URL name appears in the agent's static URL drop-down list box.

URL Description This description appears only on this page, to further describe this URL or, for example, its reason for inclusion.

URL Enter the queue push URL.

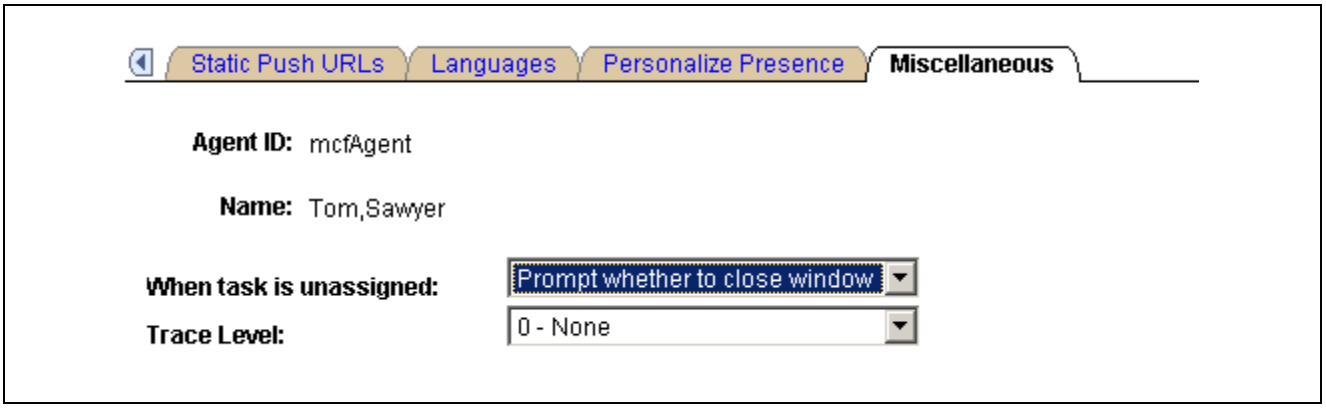
The URL must include the opening http:// and any required parameters.

All static URLs that are defined for the agent are downloaded when the agent launches the agent chat console by accepting a customer chat. Static URLs are not available in collaborative chat.

If you send a PeopleSoft Pure Internet Architecture URL, be sure that the recipient has permissions to access that portal , node, or page.

Specifying Miscellaneous Parameters

Access the Miscellaneous page.



Miscellaneous page

When task is unassigned Select from the following values the action that occurs when a task that is assigned to an agent is unassigned:

Copyright © 1988-2007, Oracle. All rights reserved.

- *Prompt whether to close window* (default).
- *Close the task window.*
- *Do not close the task window.*

Trace Level

Select from the following log trace levels:

- *0 - None*
- *1 - Information*
- *2 - Debug*

If a value other than *0* is selected, a tracer window appears to display activities and events on the chat or MultiChannel Console for debugging purposes.

Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 9

Administering Queues, Logs, and Tasks

This chapter discusses how to:

- Administer physical queues.
- View queue server, queue, and agent state.
- View broadcast, chat, and event logs.
- Administer overflow and escalated tasks.

Administering Physical Queues

To administer physical queues, use the Physical Queues (MCF_ACCPT_TASK_CMP) component.

This section discusses how to:

- Move agents between physical queues.
- Move queues.
- Balance queues.

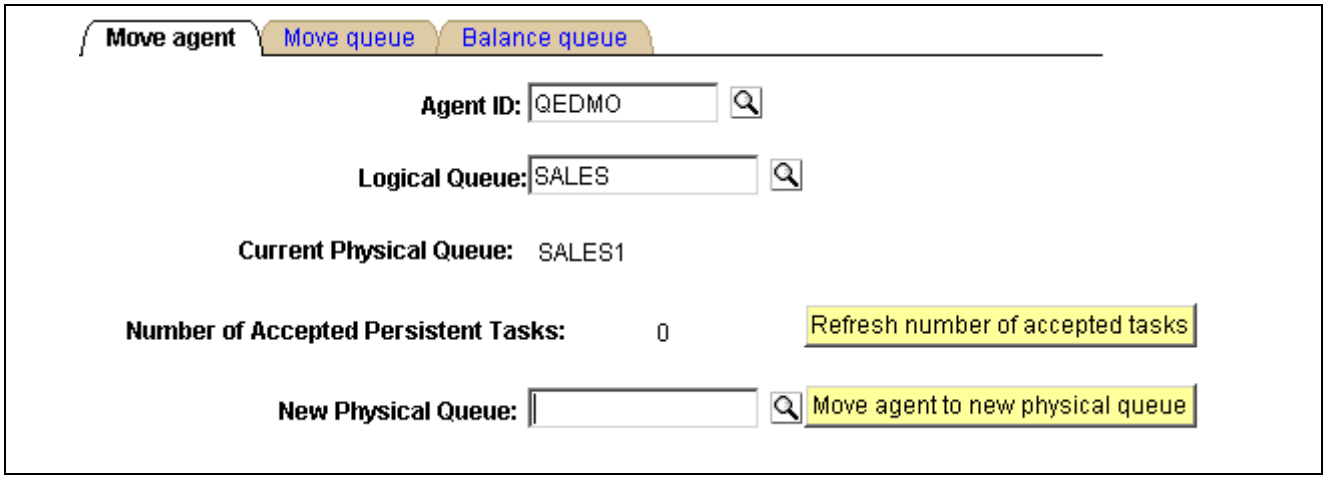
Pages Used to Administer Physical Queues

Page Name	Object Name	Navigation	Usage
Move agent	MCF_DEMO_AGENTQ	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Physical Queues, Move agent	Move an agent to a new physical queue.
Move queue	MCF_DEMO_QUEUEQ	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Physical Queues, Move queue	Move agents and tasks from one physical queue to another on the same logical queue.
Balance queue	MCF_DEMO_Q_BALANCE	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Physical Queues, Balance queue	Balance agent and task assignments over active physical queues.

Copyright © 1988-2007, Oracle. All rights reserved.

Moving Agents Between Physical Queues

Access the Move agent page.



Move agent page

You can move an agent, and any open persistent tasks that are associated with that agent, from one physical queue in one cluster to another physical queue in another cluster on the same logical queue. For example, if an agent is transferred to another office, you can move the agent to another physical queue on a server closer to the new office.

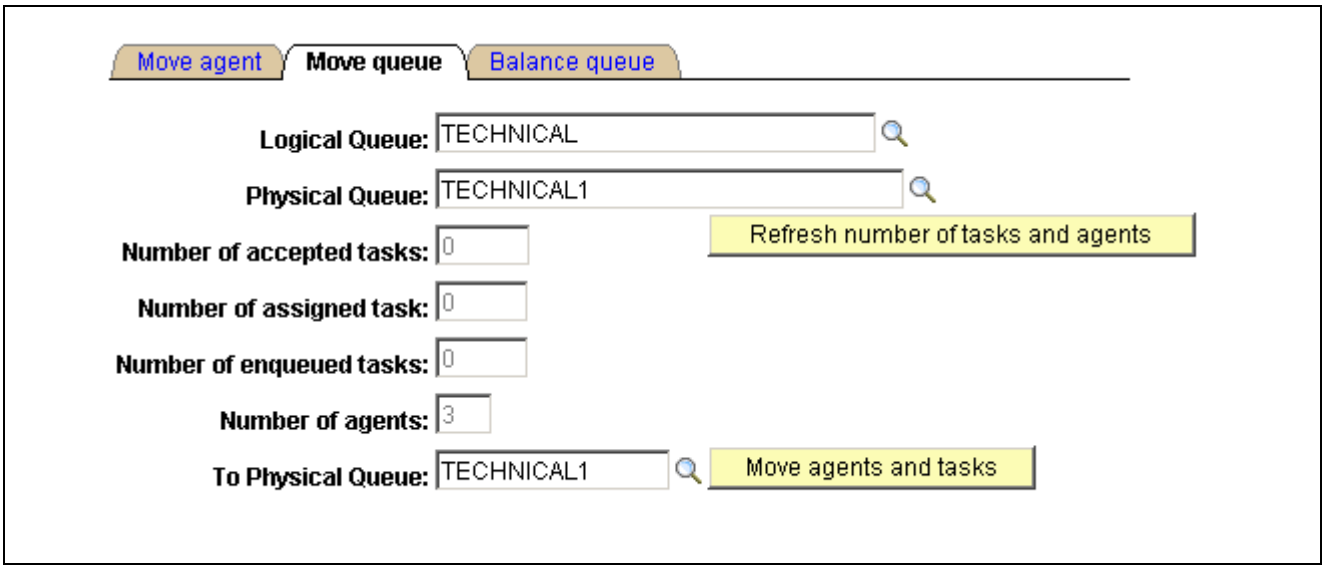
Logical Queue	Select the logical queue within which the agent is to be moved.
Number of Accepted Persistent Tasks	Displays the number of persistent tasks this agent has accepted on this physical queue. This number is updated and displayed when you select the logical queue.
Refresh number of accepted tasks	Click to update the number of persistent tasks accepted by this agent on the current physical queue.
New Physical Queue	Select the new physical queue to which this agent and the persistent tasks accepted by this agent will be assigned.
Move agent to new physical queue	Click to perform the action.

Note. Ongoing chat sessions, which are not persistent tasks, are not affected by the move agent action.

Moving Queues

Access the Move queue page.

Copyright © 1988-2007, Oracle. All rights reserved.



Move queue page

You can move all agents and their open persistent tasks from one physical queue to another physical queue within the same logical queue. For example, to delete a physical queue, move its agents and persistent tasks to another queue before deleting the first queue. Or, if the cluster serving this physical queue is overloaded, you can create another physical queue on another cluster, mark the first physical queue as inactive, create another physical queue on another cluster and move agents and persistent tasks from the inactive physical queue to the new physical queue.

Logical Queue Select the logical queue within which the selected physical queue's agents and persistent tasks will be moved.

Physical Queue Select the physical queue from which agents and persistent tasks will be moved.

Number of accepted tasks and **Number of assigned tasks** Displays the number of accepted and assigned tasks open on this physical queue.

Refresh number of tasks and agents Click to update the number of agents and persistent tasks assigned to this queue.

Number of enqueued tasks and **Number of agents** Displays the number of enqueued tasks and assigned agents on this physical queue.

To Physical Queue Select the physical queue to which the currently assigned agents and tasks will be moved.

Move agents and tasks Click to move the assigned agents and tasks to the specified physical queue.

Note. The physical queue must be inactive before moving agents and tasks. Inactivate the physical queue on the Queues page. Agents on the inactive physical queue are automatically logged off before being moved.

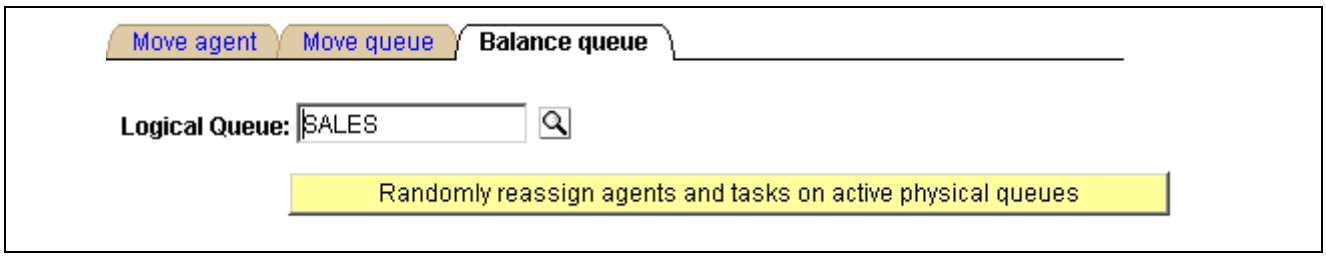
See Also

Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Defining Queues, page 85

Copyright © 1988-2007, Oracle. All rights reserved.

Balancing Queues

Access the Balance queue page.



Balance queue page

Over time, the distribution of agents and their associated skill levels, languages, and so on, across the physical queues belonging to a logical queue may change as agents are added or deleted. Rather than manually rebalancing the queue by moving individual agents, you can use the Balance queue page to randomly reassign agents and their open persistent tasks across all the active physical queues belonging to the selected logical queue.

Logical Queue Select the logical queue across which agents and persistent tasks will be balanced.

Randomly reassign agents and tasks on active physical queues Click to balance agents and tasks across the active physical queues for the selected logical queue.

This action redistributes agents and tasks assigned to this logical queue across all active physical queues without regard to previous assignments. Ensure that agents assigned to the affected physical queues have shut down their MultiChannel Consoles. They are *not* logged off automatically.

Viewing Queue Server, Queue, and Agent States

To view queue server state, queue state, and agent state, use the Queue Server State (MCF_QSERVSTATE_CMP), Queue State Summary (MCF_QUEUESTATE_CMP), and Agent State Summary (MCF_AGENTSTATE_CMP) components.

This section discusses how to:

- View the queue server state.
- View the queue state summary.
- View the agent state summary.

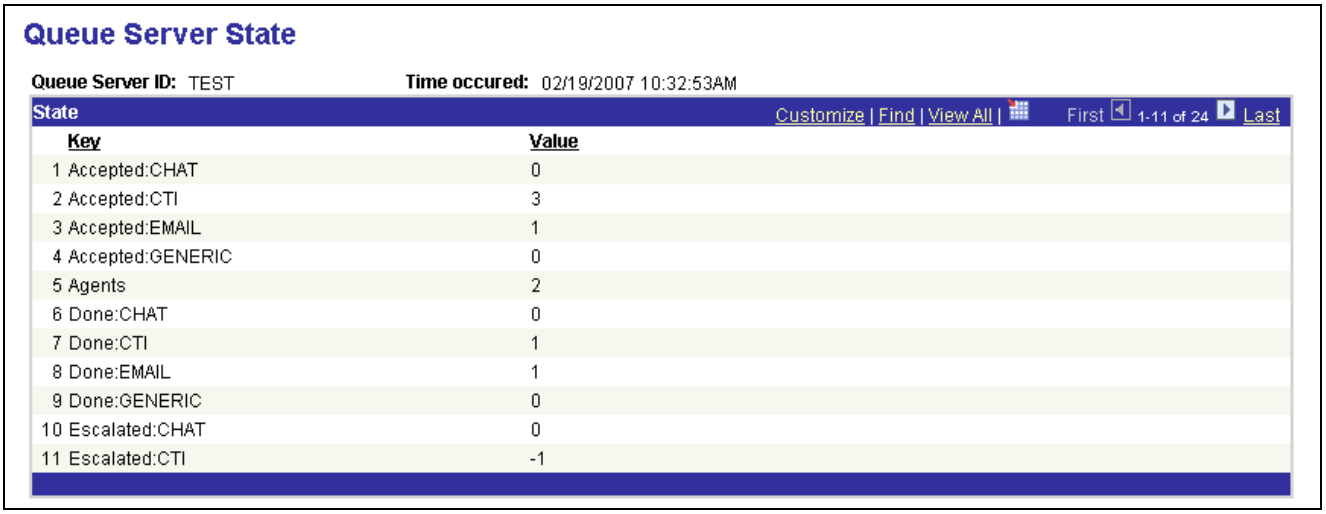
Copyright © 1988-2007, Oracle. All rights reserved.

Pages Used to View Queue Server, Queue, and Agent States

Page Name	Object Name	Navigation	Usage
Queue Server State	MCFQSERVSTATE_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Queue Server State	Examine the state of a queue server at a particular time.
Queue State Summary	MCFQUEUESTATE_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Queue State Summary	Display the state of a queue.
Agent State Summary	MCFAGENTSTATE_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Agent State Summary	View the agent state.

Viewing the Queue Server State

Access the Queue Server State page.



Queue Server State page

The Queue Server State page displays cumulative diagnostic totals for the selected queue server, at the selected time, across all physical queues serviced by this queue server. Cumulative totals are reset after each state dump.

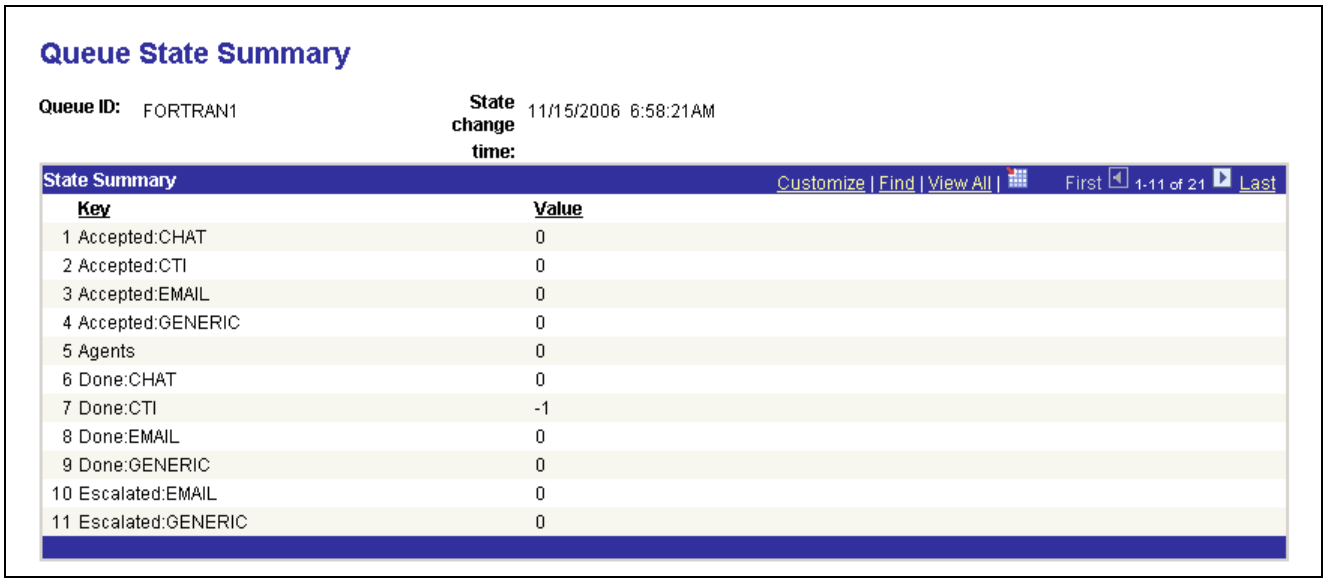
This state information comprises statistics to measure load and throughput that can be analyzed and used for tuning the system. For example, if counts are very high, consider adding another physical queue to balance the load.

The state stamps are inserted for every primary queue server by cluster at configurable intervals, irrespective of user activity. The intervals are configured on the Cluster Tuning page.

Viewing the Queue State Summary

Access the Queue State Summary page.

Copyright © 1988-2007, Oracle. All rights reserved.



Queue State Summary

Queue ID: FORTRAN1 State change time: 11/15/2006 6:58:21AM

State Summary Customize | Find | View All | First 1-11 of 21 Last

	Key	Value
1	Accepted:CHAT	0
2	Accepted:CTI	0
3	Accepted:EMAIL	0
4	Accepted:GENERIC	0
5	Agents	0
6	Done:CHAT	0
7	Done:CTI	-1
8	Done:EMAIL	0
9	Done:GENERIC	0
10	Escalated:EMAIL	0
11	Escalated:GENERIC	0

Queue State Summary page

The Queue State Summary page displays cumulative diagnostic totals for the selected physical queue.

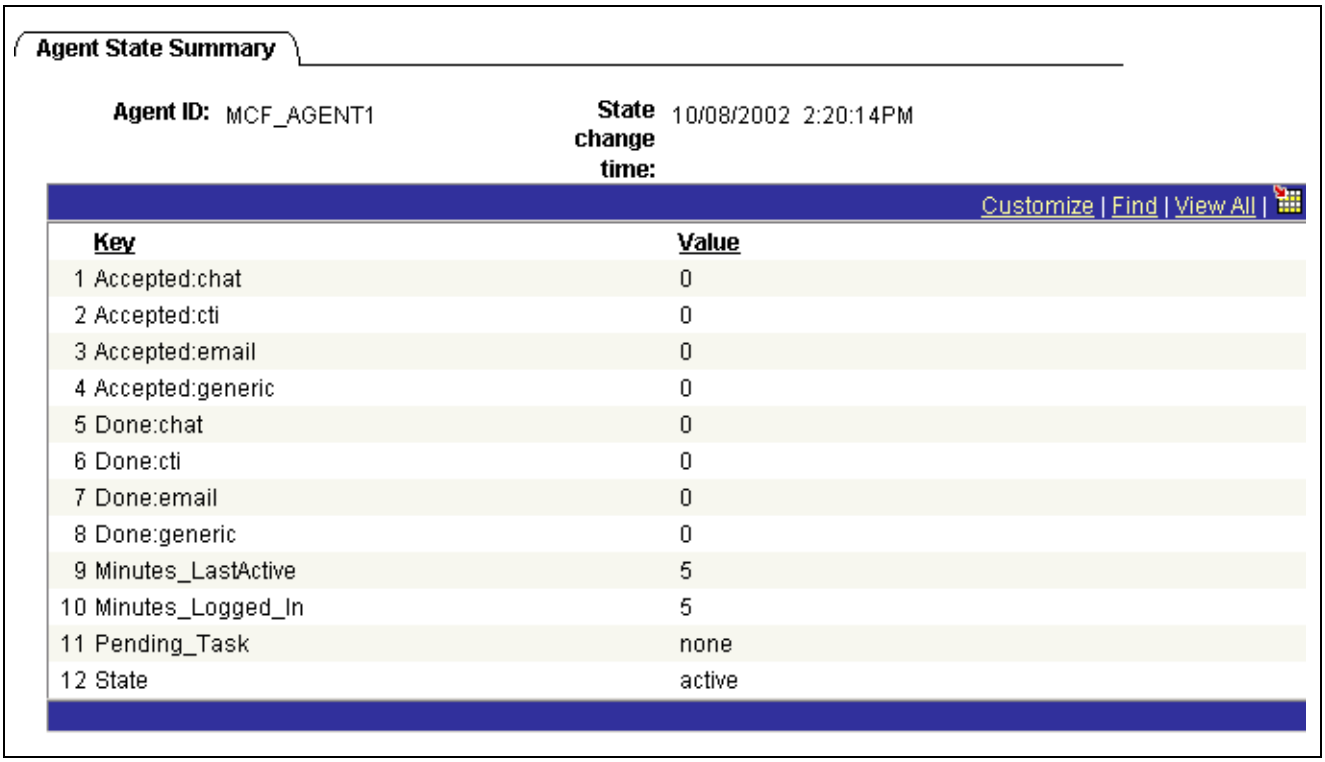
This state information comprises statistics to measure load and throughput that can be analyzed and used for tuning the system. For example, if counts are very high for this queue, consider adding another physical queue to balance the load.

The state stamps are inserted for every physical queue at configurable intervals, irrespective of user activity. The intervals are configured on the Cluster Tuning page.

Viewing the Agent State Summary

Access the Agent State Summary page.

Copyright © 1988-2007, Oracle. All rights reserved.



Agent State Summary

Agent ID: MCF_AGENT1 State change time: 10/08/2002 2:20:14PM

Customize | Find | View All |

	Key	Value
1	Accepted:chat	0
2	Accepted:cti	0
3	Accepted:email	0
4	Accepted:generic	0
5	Done:chat	0
6	Done:cti	0
7	Done:email	0
8	Done:generic	0
9	Minutes_LastActive	5
10	Minutes_Logged_In	5
11	Pending_Task	none
12	State	active

Agent State Summary page

The Agent State Summary page displays cumulative totals for the selected agent.

This state information comprises statistics to measure agent performance and status that can be analyzed and used for performance evaluation.

The state stamps are inserted for every agent at configurable intervals, irrespective of user activity. State is only recorded for agents currently logged on. The intervals are configured on the Cluster Tuning page by setting the dumpinterval. Enable agent logging by setting dumpagent to *yes*.

An agent state of *Active* indicates the agent is available; *Inactive* indicates the agent is not available.

Note. On a busy system, recording state information frequently may affect performance.

See Also

Chapter 7, “Configuring PeopleSoft MCF Queues and Tasks,” Tuning Cluster Parameters, page 93

Viewing Broadcast, Chat, and Event Logs

To view broadcast, chat and event logs, use the Broadcast Log (MCF_BCAST_LOG_CMP), Chat Log (MCF_CHAT_LOG_CMP), and Event Log (MCF_EVENTLOGL_CMP) components.

This section discusses how to:

- View broadcast logs.
- View chat logs.
- View event logs.

Copyright © 1988-2007, Oracle. All rights reserved.

- View MCF logs.

Pages Used to View Broadcast, Chat, and Event Logs

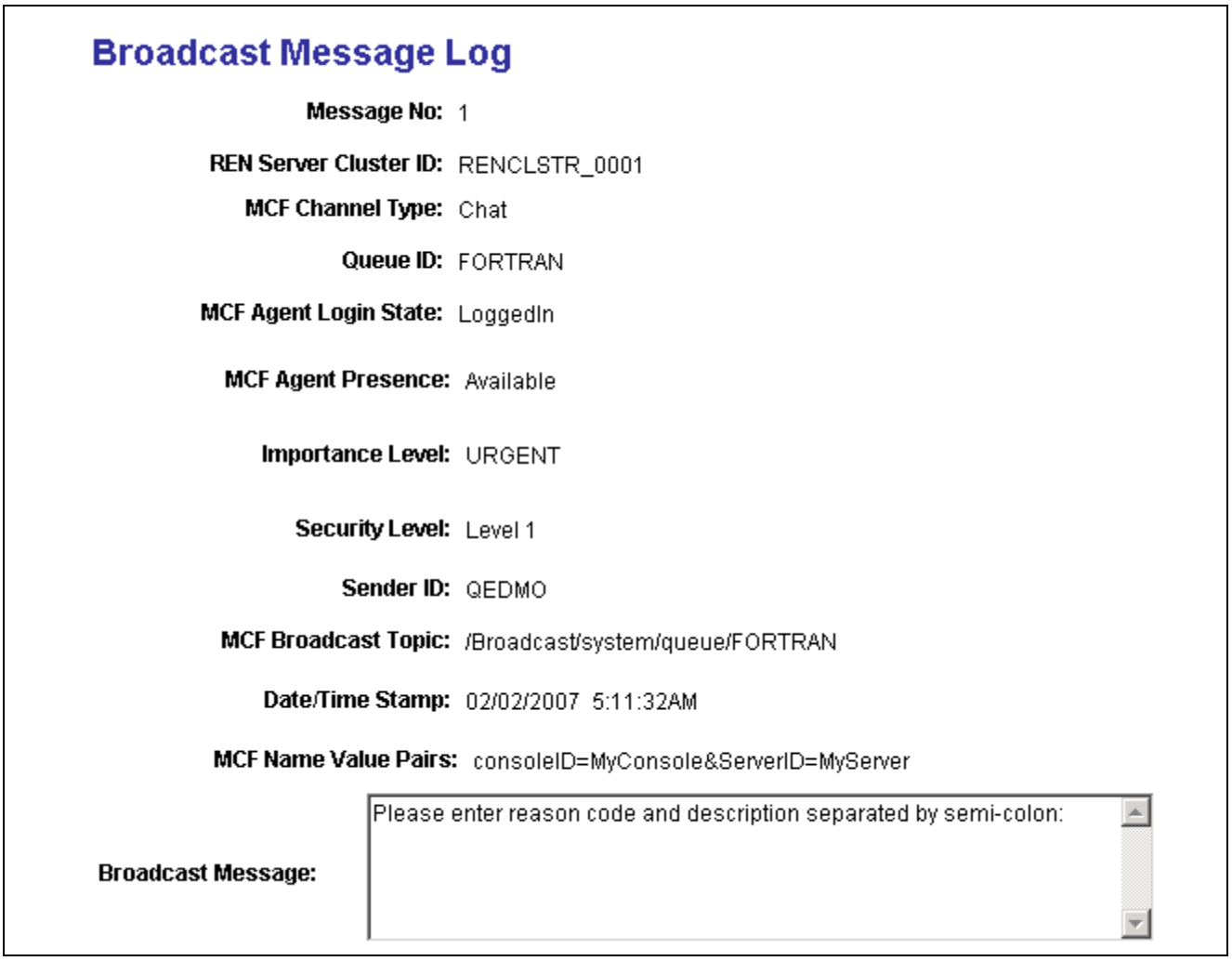
Page Name	Object Name	Navigation	Usage
Broadcast	MCF_BCASTLOG_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Broadcast Log	View the log of broadcast events.
Chat	MCF_CHAT_MINLOG_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Chat Log	View the log of a specified chat session.
Event Log	MCF_EVENTLOG_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Event Log	View a log of events and display details.

Viewing Broadcast Logs

The broadcast log page displays detailed information about the any broadcast message that was sent.

Access the Broadcast Log page by searching by message number, queue ID, or REN cluster ID.

 Copyright © 1988-2007, Oracle. All rights reserved.



Broadcast Message Log page

Note. To view the broadcast logs, set log_bcast to *Yes* on the Cluster Tuning page. For this parameter to take effect, click Refresh logging parameters on the Cluster Notify page.

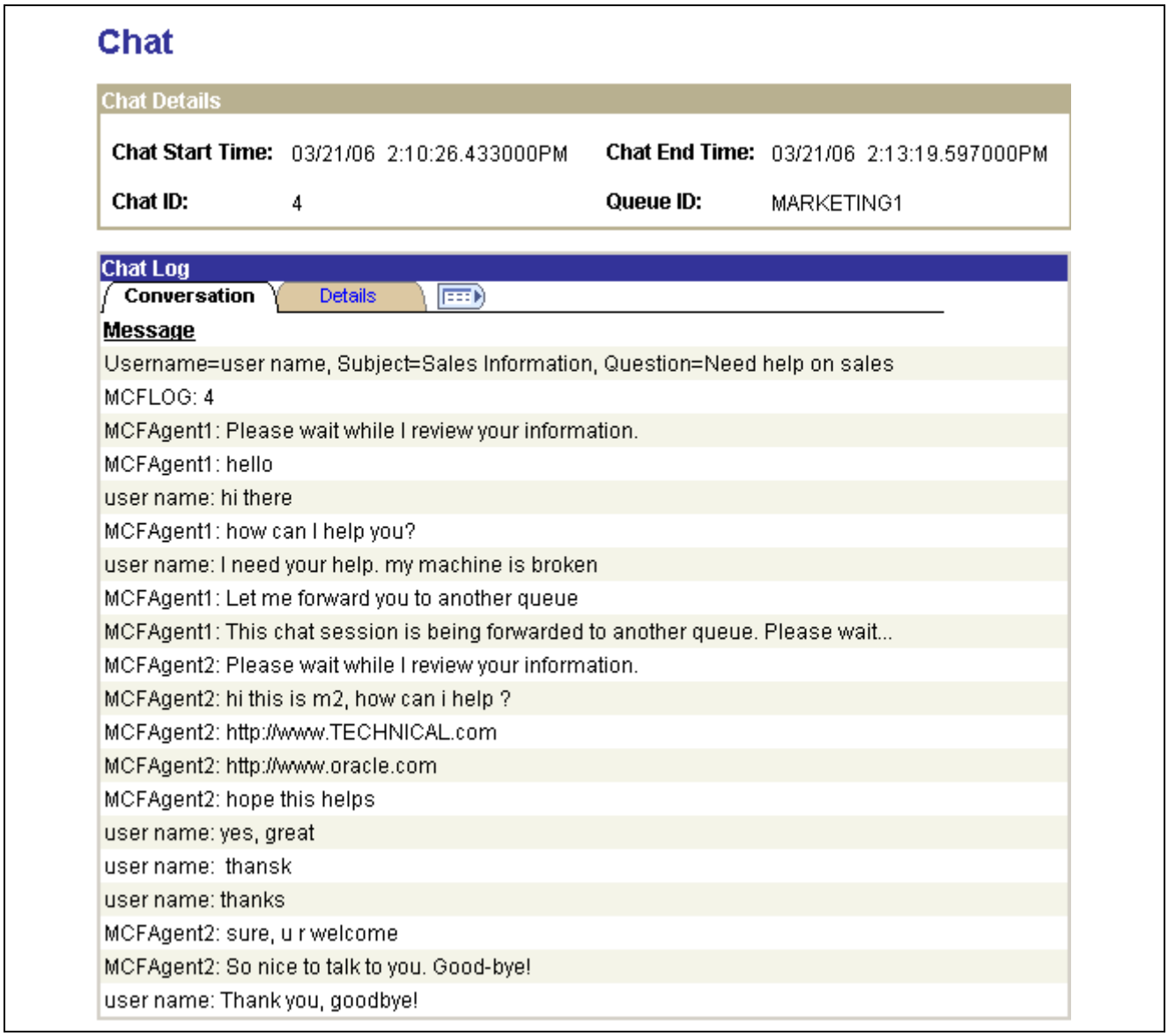
See Also

Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using PeopleSoft MCF Broadcast, page 137

Viewing Chat Logs

Access the Chat page.

Copyright © 1988-2007, Oracle. All rights reserved.



Chat Log- Conversation tab

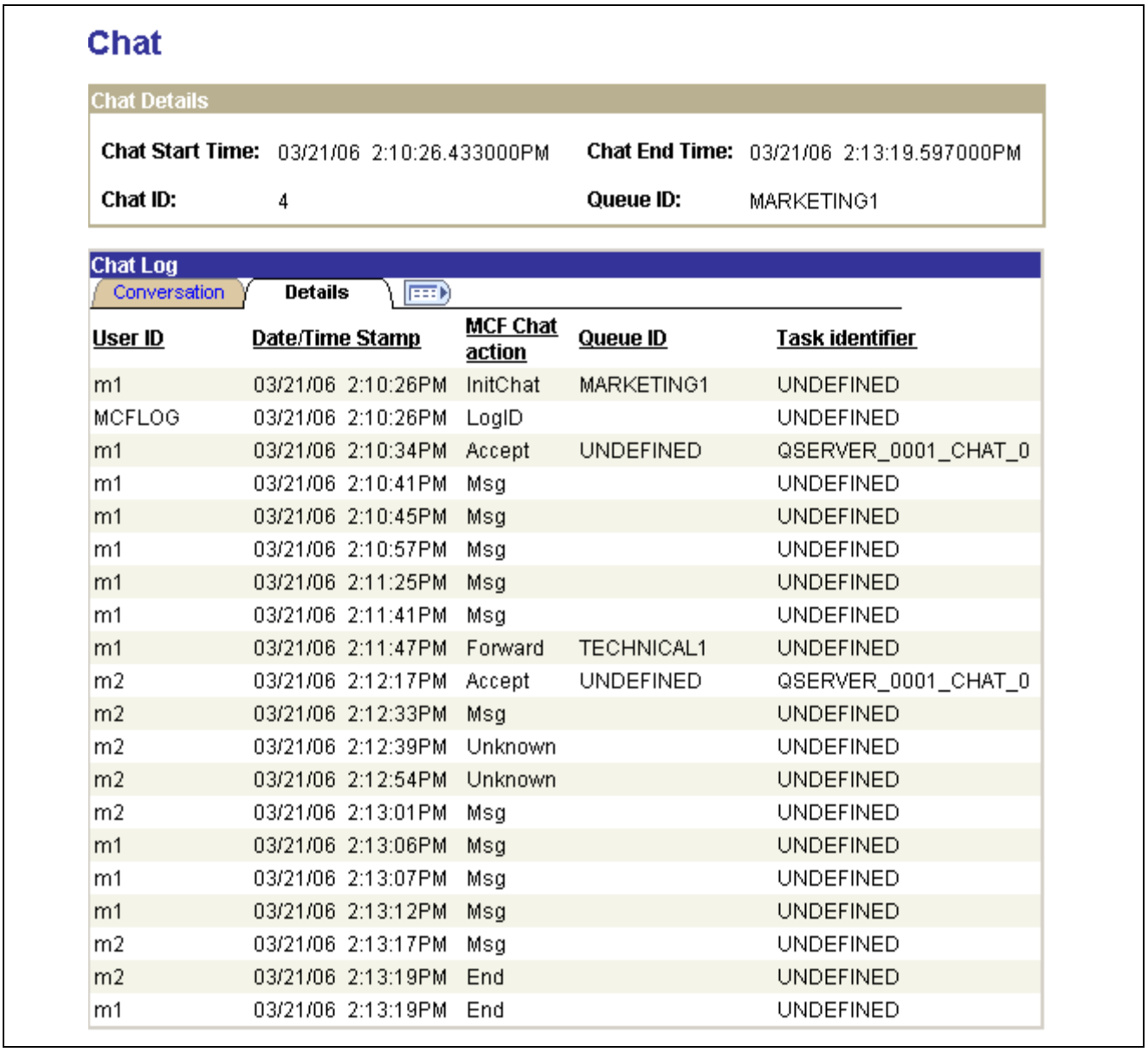
If enabled, this log records the contents and events of every chat session.

To enable chat logging, set log_chat_ses to *yes* on the Cluster Tuning page. For this parameter to take effect, click Refresh logging parameters on the Cluster Notify page.

Details Tab

Select the Details tab.

Copyright © 1988-2007, Oracle. All rights reserved.



Chat

Chat Details

Chat Start Time: 03/21/06 2:10:26.433000PM Chat End Time: 03/21/06 2:13:19.597000PM

Chat ID: 4 Queue ID: MARKETING1

Chat Log

Conversation Details

User ID	Date/Time Stamp	MCF Chat action	Queue ID	Task identifier
m1	03/21/06 2:10:26PM	InitChat	MARKETING1	UNDEFINED
MCFLOG	03/21/06 2:10:26PM	LogID		UNDEFINED
m1	03/21/06 2:10:34PM	Accept	UNDEFINED	QSERVER_0001_CHAT_0
m1	03/21/06 2:10:41PM	Msg		UNDEFINED
m1	03/21/06 2:10:45PM	Msg		UNDEFINED
m1	03/21/06 2:10:57PM	Msg		UNDEFINED
m1	03/21/06 2:11:25PM	Msg		UNDEFINED
m1	03/21/06 2:11:41PM	Msg		UNDEFINED
m1	03/21/06 2:11:47PM	Forward	TECHNICAL1	UNDEFINED
m2	03/21/06 2:12:17PM	Accept	UNDEFINED	QSERVER_0001_CHAT_0
m2	03/21/06 2:12:33PM	Msg		UNDEFINED
m2	03/21/06 2:12:39PM	Unknown		UNDEFINED
m2	03/21/06 2:12:54PM	Unknown		UNDEFINED
m2	03/21/06 2:13:01PM	Msg		UNDEFINED
m1	03/21/06 2:13:06PM	Msg		UNDEFINED
m1	03/21/06 2:13:07PM	Msg		UNDEFINED
m1	03/21/06 2:13:12PM	Msg		UNDEFINED
m2	03/21/06 2:13:17PM	Msg		UNDEFINED
m2	03/21/06 2:13:19PM	End		UNDEFINED
m1	03/21/06 2:13:19PM	End		UNDEFINED

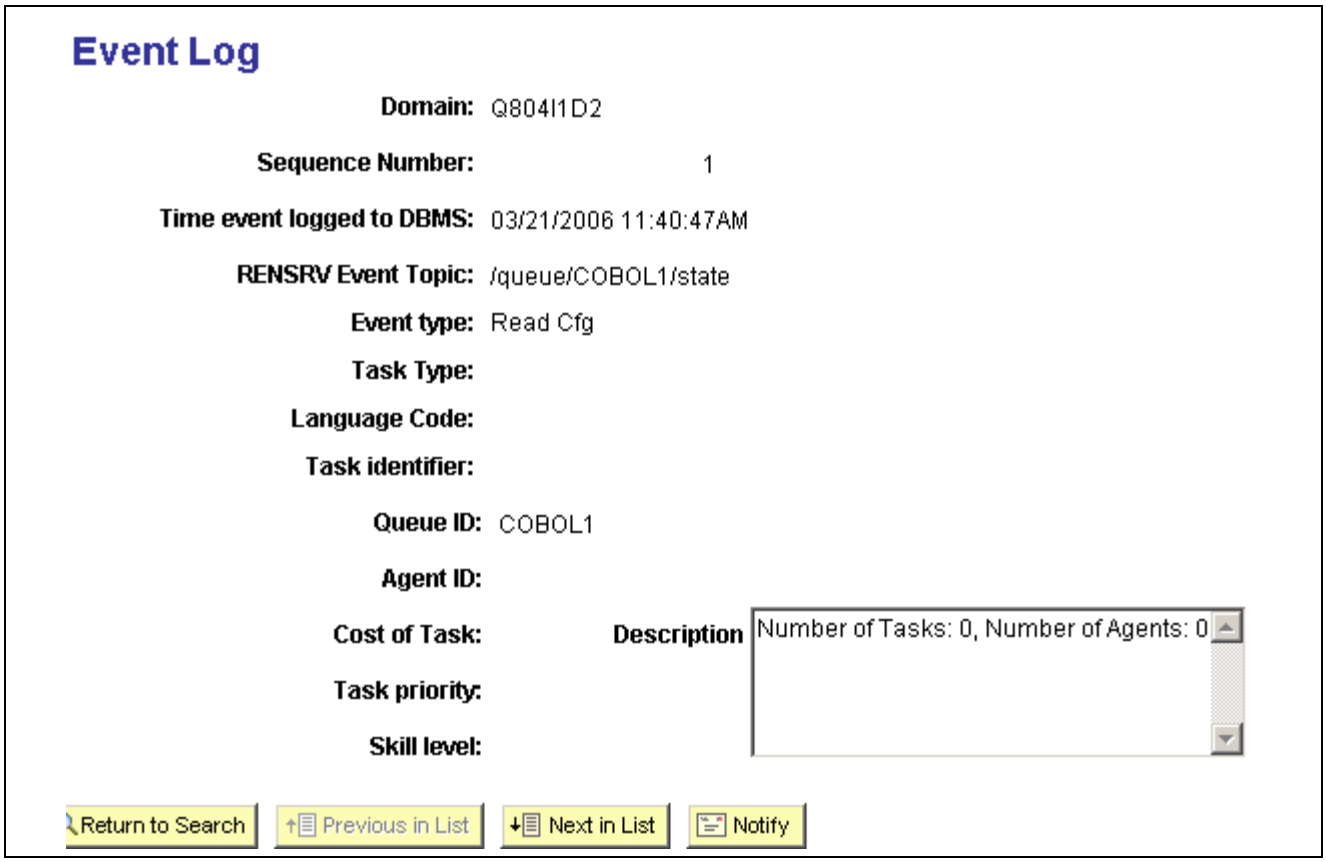
Chat Log-Details tab

The Details tab displays detailed information about the selected chat conversation.

Viewing Event Logs

Access the Event Log page.

Copyright © 1988-2007, Oracle. All rights reserved.



Event Log page

Domain	The application server domain on which this event occurred.
Time event logged to DBMS	The time that this event was recorded in the database.
Event Type	The type of event, as described in the table that follows.
Task Type	The type of task for this event: chat, email, or generic. CTI events are logged in the CTI event log.

The event log records PeopleSoft MultiChannel Framework events sent to the real-time event notification (REN) server, excluding chat content (which is logged in the chat log). The event log can be used for debugging as well as for system monitoring. For example, you can determine when agents log in and log out, or when the queue server was first notified of newly enqueued events.

Data displayed in the event log depends on the event type. Not all fields apply to every event.

To enable logging of state broadcast events, set logDMPQ to yes on the Cluster Tuning page. For this parameter to take effect, click Refresh logging parameters on the Cluster Notify page.

Time is displayed and searched on in the format MM/DD/YYYY HH:MM:SSA/PM.

The following table lists possible event types:

Copyright © 1988-2007, Oracle. All rights reserved.

Name	Translate Table Value	REN MultiChannel Framework Topic	Description
Accepted	ACPD	/agent/<agentID> /accepted Agent 's list of accepted tasks	Agent's list of accepted tasks
Accept	ACPT	/queue/agents/accept	Agent accepts an assigned task
Bcst Admin	BCST	/queue/admin/statedump	Broadcast universal queue information
Contact	CNCT	/queue/contact	Real-time contact (for example, chat)
DB Cntct	DBCT	/queue/dbcontact	Database management system contact (such as email or generic)
Dump Q	DMPQ	/queue/<queueID>/state	Dump queue state information to log
Done	DONE	/queue/agents/dequeue	Done (dequeue)
Forward	FWD	/queue/agents/forward	Forward
Notify	NTFY	/agent/<agentID>/notify	Notify agent of assigned task
Presence	PRES	/queue/agents/presence	Agent's presence change
Restrt Ack	RACK	/queue/agents/restartack	Restart acknowledgement
Read Cfg	READ	/uqsrv/reread/defaults	Reread defaults
Restart	RSRT	/queue/<queueID>/restart	Restart
Unknown	UNKN	UNKNOWN	Unknown REN server event
Unassign	USGN	/agent/<agentID> /unassign	Unassign

Copyright © 1988-2007, Oracle. All rights reserved.

The following table lists event logs:

Name	Translate Table Value	REN MultiChannel Framework Topic	Required Argument Value	Meaning
Abandon	ABAN	/chat/<userID> /<chatID>	ps_type=abandon	Abandoned chat session
End	END	/chat/<userID> /<chatID>	ps_type=end	End chat session
Login	LGIN	/queue/agents /loginstate	ps_state=login	Log on
Logout	LGOT	/queue/agents /loginstate	ps_state=logout	Log out
Message	MSG	/chat/<userID> /<chatID>	ps_type=msg	Message
Push URL	PUSH	/chat/<userID> /<chatID>	ps_type=pushurl	Push URL
Timeout	TOUT	/chat/<userID> /<chatID>	ps_type=timeout	Timeout chat session

Viewing PeopleSoft MCF Logs

Diagnostic PSUQSRV and PSMCFLOG traces are written to each application server domain's log directory. The trace level is determined by the LogFence setting in the domain configuration. You can set the LogFence parameter with PSADMIN configuration of the application server domain. The following table lists LogFence setting values:

LogFence Setting	Tracing Level
1	Fatal errors
2	Errors
3	Warnings
4	Level 1 Diagnostic: Logs the queues that the universal queue is servicing, logs new queues that are added, and logs agent logon and logout.

 Copyright © 1988-2007, Oracle. All rights reserved.

LogFence Setting	Tracing Level
5	Level 2 Diagnostic: Logs most debugging information, except periodic events (such as timer check and heartbeat).
6	Level 3 Diagnostic: Logs everything, including periodic events.

Administering Overflow and Escalated Tasks

To administer overflow and escalated tasks, use the Overflow Administration (MCF_OVERFLOWL_CMP) and Escalation Administration (MCF_ESCAL_CMP) components.

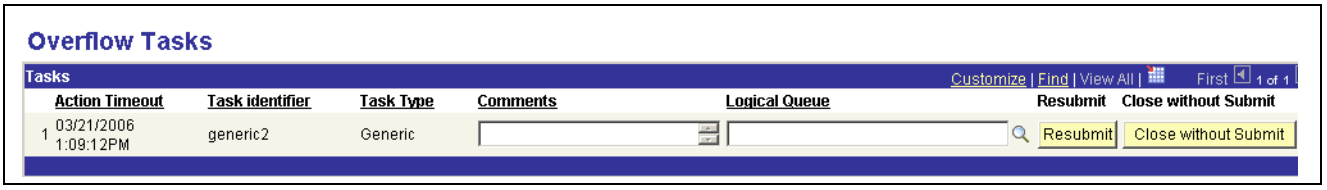
This section discusses how to administer overflow and escalated tasks.

Pages Used to Administer Overflow and Escalated Tasks

Page Name	Object Name	Navigation	Usage
Overflow tasks	MCF_OVERFLOWL_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Overflow Admin (overflow administration)	View a list of overflow events.
Escalation Admin (escalation administration)	MCF_ESCAL_PG	PeopleTools, MultiChannel Framework, Universal Queue, Administration, Escalation Admin (escalation administration)	View a list of escalated events.

Administering Overflow Tasks

Access the Overflow tasks page.



Overflow Tasks page

Use this page to manage tasks that could not be assigned to an agent within the specified overflow timeout.

Action Timeout Displays the time at which the overflow occurred.

Task Identifier Displays the task identifier.

Task Type Types include chat, email, generic, and voice (not supported).

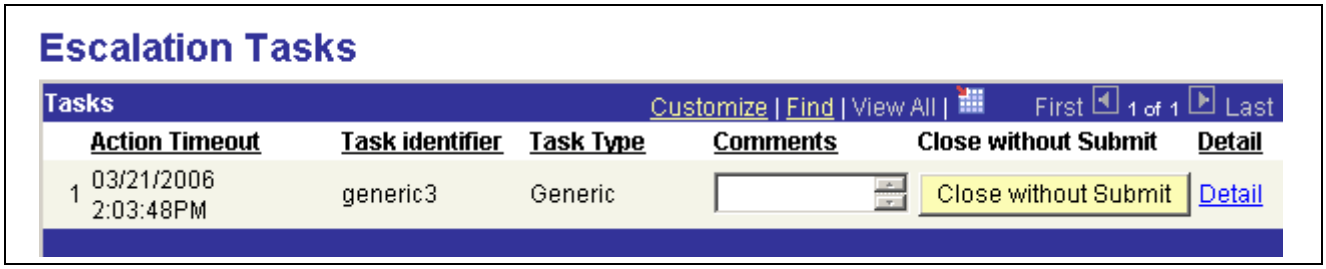
Comments Enter optional text commentary about the task's resolution.

Copyright © 1988-2007, Oracle. All rights reserved.

	For example, note that the customer sent follow-up email that was answered by an agent.
Logical Queue	Select a logical queue to resubmit this task to.
Resubmit	Click to send the task to the specified logical queue to retry assignment. Only persistent tasks can be resubmitted; chat cannot be resubmitted.
Close without Submit	Click to close this task without sending it to retry assignment.
Detail	Click to display additional information about the task.

Administering Escalated Tasks

Access the Escalation Admin page.



Escalation Tasks page

Use this page to manage accepted persistent tasks that have been automatically unassigned from agents because they were not closed within the specified escalation timeout. Tasks can be resubmitted or closed.

Action Timeout	Displays when the escalation occurred.
Task Identifier	Displays the task identifier.
Task Type	Types include chat, email, generic, and voice (not currently supported).
Logical Queue	Select a logical queue to submit this task to.
Resubmit	Click to send the task to the selected logical queue to retry assignment. Only persistent tasks can be resubmitted; chat cannot be resubmitted.
Comments	Enter optional text commentary about the task's resolution. For example, note that the customer sent follow-up email that was answered by an agent.
Detail	Click to view additional information about this escalated task.
Close without Submit	Click to close this task without sending it back to retry assignment.

Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 10

Managing Tasks and Using Chat in PeopleSoft MultiChannel Framework

This chapter discusses how to:

- Manage tasks with the MultiChannel Console.
- Communicate with customers and agents using chat.

Managing Tasks with the MultiChannel Console

Use the MultiChannel Console to accept, respond to, transfer, and complete tasks, and to initiate or join chat sessions.

This section discusses how to use the MultiChannel Console to work with tasks.

Using the MultiChannel Console to Work with Tasks

If your user ID includes security permissions for PeopleSoft MultiChannel Framework (MCF), the MultiChannel Console navigation tool appears in the universal navigation header. To access the console, click the tool.

MultiChannel Console

You cannot launch two MultiChannel Consoles on the same workstation.

You can launch consoles on different workstations using the same user ID, but you cannot log on to physical queues served by the same MCF cluster.

When the MultiChannel Console is launched, it attempts to connect to the real-time event notification (REN) server associated with the last physical queue listed in the My Queue drop-down list box. If that REN server is not running, the agent receives an error message. If one or more of the agent's physical queues are associated with a running REN server, the agent can log on by selecting another queue.

Newly assigned tasks appear as a link above the Tasks drop-down list box, and display a flashing icon. Click the link to accept the task. Accepted tasks appear in the Tasks drop-down list box.

Collaborative chat requests (agent-to-agent chats) may appear as a link above the Agent Chats drop-down list box, but all collaborative chat requests (accepted or not) appear in the Agent Chats drop-down list box.

Copyright © 1988-2007, Oracle. All rights reserved.

Configure the size and initial location of the MultiChannel Console on the agents Window Config (window configuration) page.

Note. If you use web content zones in Microsoft Internet Explorer security options, your PeopleSoft Pure Internet Architecture web server address and your REN server address must be in the same security zone.

If you use Secure Sockets Layer (SSL) security, you may receive a security warning message when your console first opens. You can accept the warning message without compromising SSL security.

Tasks

After you accept an assigned task, the task appears here.

Activate a task to work on by selecting it. Activating a task brings the window associated with that task to the foreground. If the associated task window is not running, it is launched by the console.

A task is removed from the task list when you mark the task as done using one of the following methods:

- For email and generic tasks, the task's application page may include a Done button; refer to the application's documentation.
- For chat, the task is done when the chat dialog ends.

Agent Chats

Displays a list of your collaborative chats, both accepted and requested, including a label indicating if the chat is inbound or outbound. Select a chat to activate it.

If you are invited to join in a chat session (conference), the conference chat is added to the list. After you accept the conference, the conference is added to the Tasks drop-down list box and removed from the Agent Chats drop-down list box. The chat is removed from the list when the chat dialog ends.

Buddies

Includes agents that are identified as buddies in your agent configuration. Add buddies on the Buddy List page of the Agent component.

To initiate a collaborative chat, select an agent. A chat dialog box displays.

Note. You can only initiate a collaborative chat with an agent who appears on your buddy list. However, if you appear on another agent's buddy list, that agent can initiate a collaborative chat with you, even if that agent is not one of your buddies.

My Queue

Select a queue to log on to that queue.

The drop-down list box includes the physical queues that you can access.

To change queues, select a new queue. You are logged off from the old queue and logged on to the selected queue.

Add or delete queues on the Queues page of the Agents component.

Presence

Displays your current presence and enables selection of a new presence. Select from:

- *Active*
- *Inactive*

Customize presence labels to more closely track agent activity. Edit your presence options on the Presence page of the Agent component.

 Copyright © 1988-2007, Oracle. All rights reserved.

If you select an inactive presence, an icon appears in the upper-left corner of the MultiChannel Console. Changes in agent presence are logged and include the customized presence description.

Presence status persists for eight hours from the time of its last change.

You can use hot key combinations, described in the following table, to navigate between MultiChannel Console fields:

Hot Key Combination	Opens
ALT + A	Agent chats
ALT + B	Buddies
ALT + P	Presence
ALT + Q	My Queues
ALT + T	Tasks

Communicating with Customers and Agents Using Chat

This section discusses how to:

- Use the agent chat window.
- Use the customer chat window.

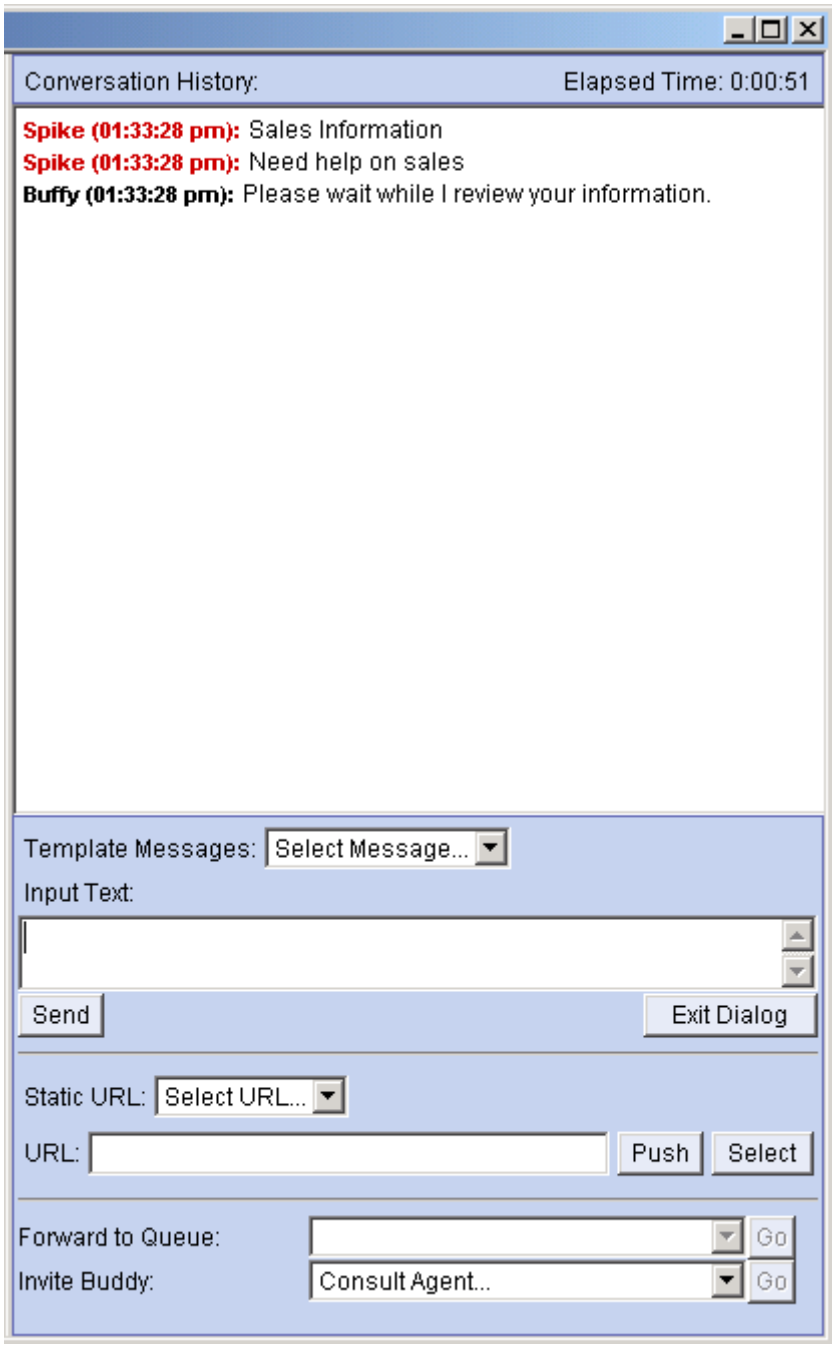
Using the Agent Chat Window

When you accept a chat task, a chat window opens.

The format of the agent chat window and the contents of its right pane are delivered as part of PeopleSoft MultiChannel Framework. The content of the left pane is determined by application developers, and is passed as a parameter to the InitChat() built-in function.

The following example shows an agent chat window:

Copyright © 1988-2007, Oracle. All rights reserved.



Agent chat window

Configure the size and initial location of the agent chat window on the agent's Window Config (window configuration) page.

Conversation History Lists progress of the chat, line by line.

If chat logging is enabled, the conversation is recorded in the database chat log by the MCF log server. View the chat log on the Chat Log page.

If accessibility features are not turned off in the My Personalizations component, an additional text box appears below the conversation history. The

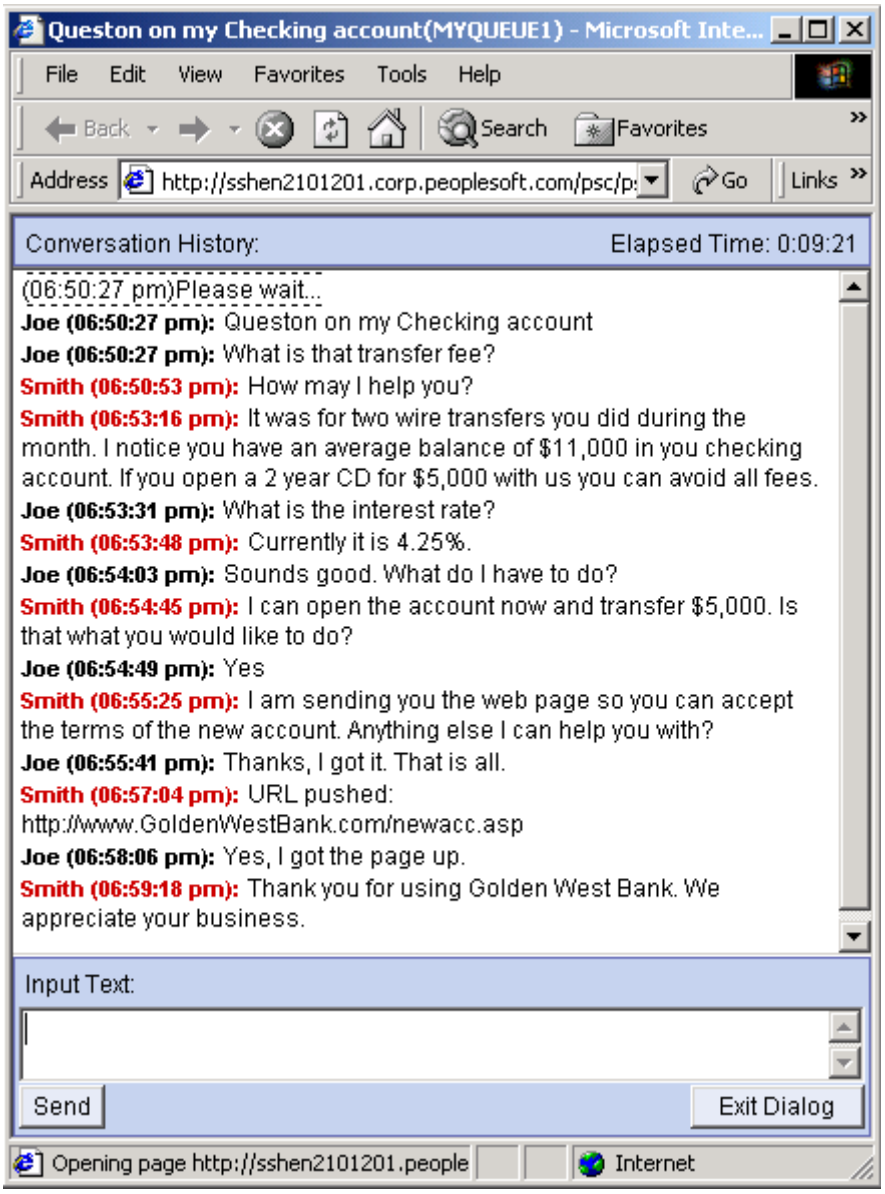
Copyright © 1988-2007, Oracle. All rights reserved.

	most recent customer input appears in the secondary text box, which can be read by screen reading software.
Template Messages	Send the customer a standard message by selecting one from the list. The message text appears in the Input Text text box. Click Send to send the message. Edit template messages for each queue on the queue Chat Responses page.
Input Text	To respond to the customer, enter text, then click Send or press ENTER. The maximum size of the text that can be sent at one time is 4096 bytes (4 kilobytes).
Send	Click to send the contents of the Input Text text box.
Exit Dialog	Click to end the chat. The chat window remains open, enabling you to return to the page for follow-up work after the customer exits the dialog.
Static URL and **Push**	To send a static URL to the customer, select a URL name, then click Push. When a URL is pushed to a customer, a browser window for that URL is launched on the customer's workstation.
URL	Displays static and grabbed URLs.
Select	Click to launch an application page, from which a URL can be returned to populate the URL field. The format of the URL wizard page that appears when you click Grab is defined by application developers. PeopleSoft supplies a sample URL wizard page. See Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the URL Wizard, page 149.
Forward to Queue	To forward the current chat to another queue, select the queue. You can only forward to another physical queue, and that physical queue must be served by the same MCF cluster as the physical queue that assigned the chat.
Invite Buddy	To request that another agent join the chat, select the agent and click Go. The buddy must be logged on to a physical queue that is served by the same MCF cluster as the physical queue that sent you this chat. Chat conferencing is only supported on customer-initiated chat sessions.

Using the Customer Chat Window

When a customer initiates a chat, a chat window appears:

Copyright © 1988-2007, Oracle. All rights reserved.



Customer chat window

The format and content of the customer chat window is delivered as part of PeopleSoft MultiChannel Framework. The customer chat window includes a conversation history text box, input text box, and Send and Exit Dialog buttons.

If accessibility features are not turned off in the My Personalizations component, an additional text box appears below the conversation history. The most recent agent input appears in the secondary text box, which can be read by screen reading software.

Input Text The customer enters text here.

The maximum size of the text that can be sent at one time is 4096 bytes (4 kilobytes).

Send Click to send the input text to the agent.

Exit Dialog Click to end the chat and close the chat window.

Copyright © 1988-2007, Oracle. All rights reserved.

Note. The agent collaborative chat window is substantially the same as the customer chat window.

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 11

Using PeopleSoft MCF Broadcast and Working with Sample Pages

This chapter discusses how to:

- Configure PeopleSoft MCF Broadcast.
- Work with sample pages.
- Use and demonstrate the JavaScript MultiChannel API (JSMCAPI).
- Use PeopleCode built-in functions.
- Use Universal Queue classes.

Using PeopleSoft MCF Broadcast

Use the broadcast function to broadcast a notification message. This function is typically used by a supervisor to send a notification message to specific recipients based on the parameters that are provided by the sender. Broadcast notifications can be sent system-wide, cluster-wide, queue-based, task-based, or activity-based.

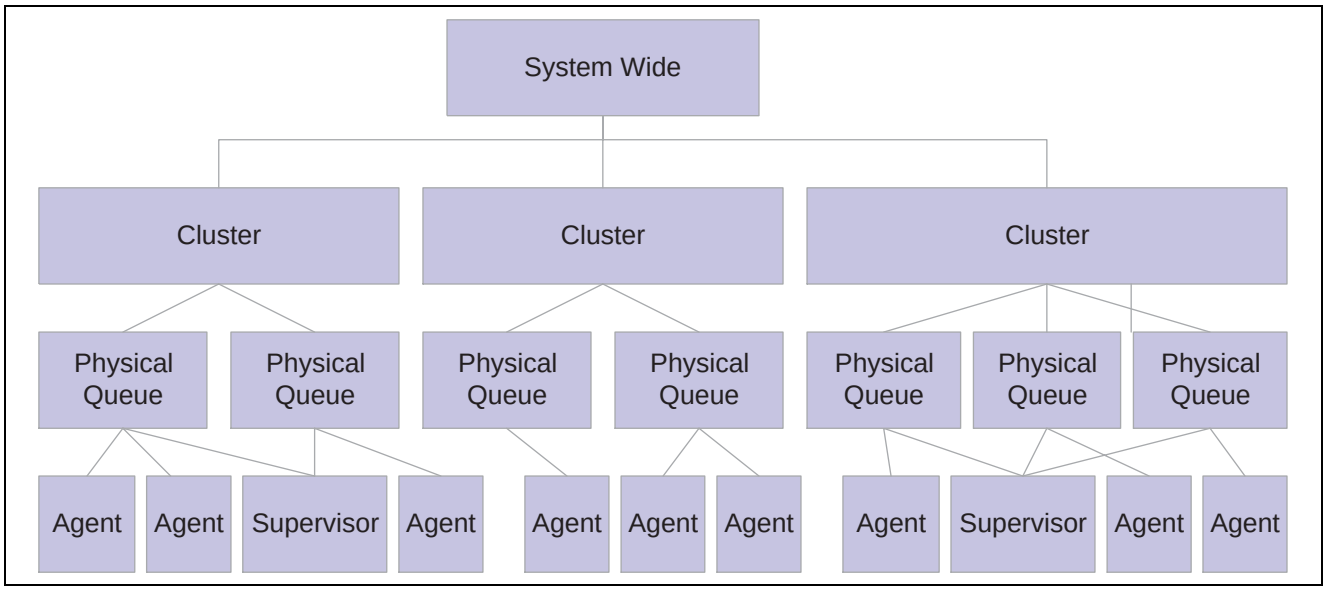
System-wide broadcast notifications can be sent only using the PeopleCode API to all logical queues on the system. The same notification using JSMCAPI is sent to all the physical queues on the cluster.

Cluster-wide broadcast enables a user to broadcast to an entire audience that is logged on or subscribed to this broadcast topic on a particular cluster. A cluster broadcast can be configured to target a specific audience by specifying a particular queue, a channel, or an agent log state in the broadcast call.

A queue-based broadcast can send a notification to all agents on a particular queue. A channel-type broadcast is received by agents serving a particular channel. An agent working on multiple channels can receive broadcast that is meant for all those channels. An agent login state or activity-based broadcast allows a broadcast notification to be sent to all agents in a particular state.

The following diagram illustrates a cluster-wide broadcast:

Copyright © 1988-2007, Oracle. All rights reserved.



Cluster-wide broadcast

This section provides an overview of JSMCAPI broadcast and discusses how to:

- Implement MCF broadcast.
- Configure JSMCAPI Broadcast using MCF Supervisor console.
- Configure PeopleCode broadcast.
- View broadcast logs.

Pages Used to Configure PeopleSoft MCF Broadcast

Page Name	Object Name	Navigation	Usage
JSMCAPI Broadcast	MCF_BROADCAST	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, JSMCAPI Broadcast	Configure JSMCAPI Broadcast using MCF Supervisor Console.
PCodeBroadcast	MCF_DEMO_BROADCAST	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, PCodeBroadcast	Send a broadcast message using PeopleCode broadcast.

Understanding JSMCAPI Broadcast

A JSMCAPI console primarily operates on physical queues. Whenever an agent or a supervisor uses a JSMCAPI console, the agent chooses a physical queue and logs onto it. To broadcast a message, a console typically uses this physical queue as a parameter to determine the target cluster. JSMCAPI broadcast works on the following restrictions:

- JSMCAPI broadcast function does not provide any routing logic because it operates within a limited set of physical queues that are assigned to the agent or supervisor.

  The agent can choose any physical queue from the assigned queue list without logging onto any particular physical queue.

 Copyright © 1988-2007, Oracle. All rights reserved.

- The supervisors or the users of JSMCAPI broadcast can use only the physical queues on their list to send a broadcast notification.
- JSMCAPI works within the confine of a single cluster and does not have any knowledge or access to other clusters in the system.

Implementing MCF Broadcast

To enable you to implement broadcast, MultiChannel Framework provides a PeopleCode built-in function and a JSMCAPI interface for JSMCAPI users (agents and supervisors) to send broadcast notifications.

Note. The broadcast notifications are displayed only in the third-party sample pages.

Subscribing and Publishing Broadcast

The JSMCAPI broadcast function is a specialized publish call to the REN server. Both JSMCAPI and PeopleCode publish to physical-queue broadcast topic (such as /Broadcast/SALES1), and the subscribers always subscribe to broadcast topic for the physical queues to which they were assigned.

To subscribe, use the following function:

```
BroadcastSubscribe(cluster,queue,task,state,presence,method)
```

To publish, use the following function:

```
MCFBroadcast(cluster,queue,task,state,presence,message,securitylevel,
importancelevel, senderid, NameValuePairString
```

To unsubscribe, use the following function:

```
void broadcastUnsubscribe(type)
```

where <type> determines the type of broadcast, system-wide, queue-wide, or agent-wide.

See Also

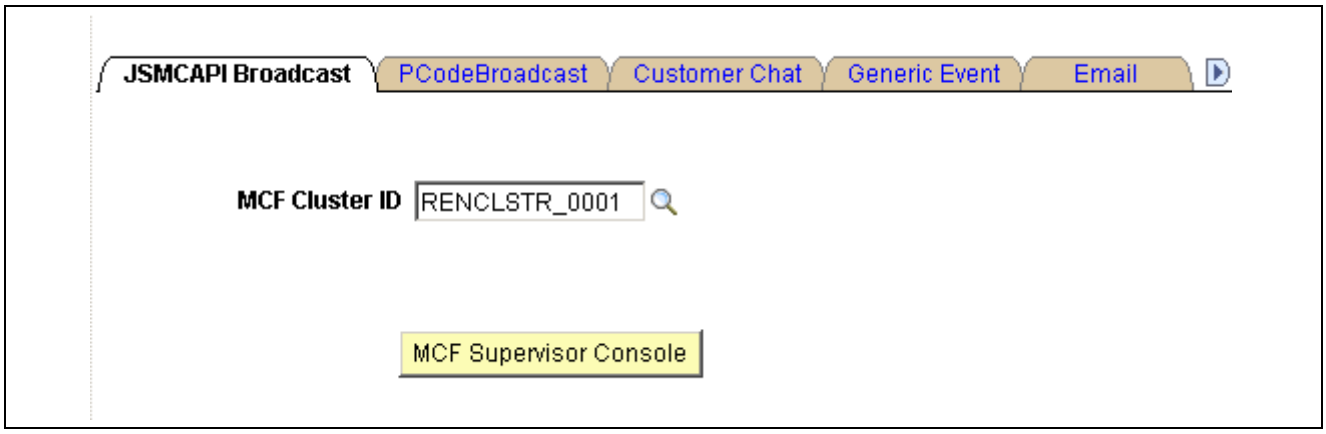
Chapter 13, “Understanding JSMCAPI Classes,” broadcastSubscribe, page 365

Chapter 13, “Understanding JSMCAPI Classes,” broadcastUnsubscribe, page 365

Using JSMCAPI Broadcast with MCF Supervisor Console

Access the JSMCAPI Broadcast page.

Copyright © 1988-2007, Oracle. All rights reserved.



JSMCAPI Broadcast page

REN Server Cluster ID Select the REN server cluster on which to test the MCF Supervisor console.

Note. The MCF Supervisor console is specific to the REN cluster Id selected.

MCF Supervisor Console Click to initiate the supervisor console.

Note. To demonstrate the MCF Supervisor console, you must have MCF servers, both queue and log servers, running and communicating with the specified REN server cluster.

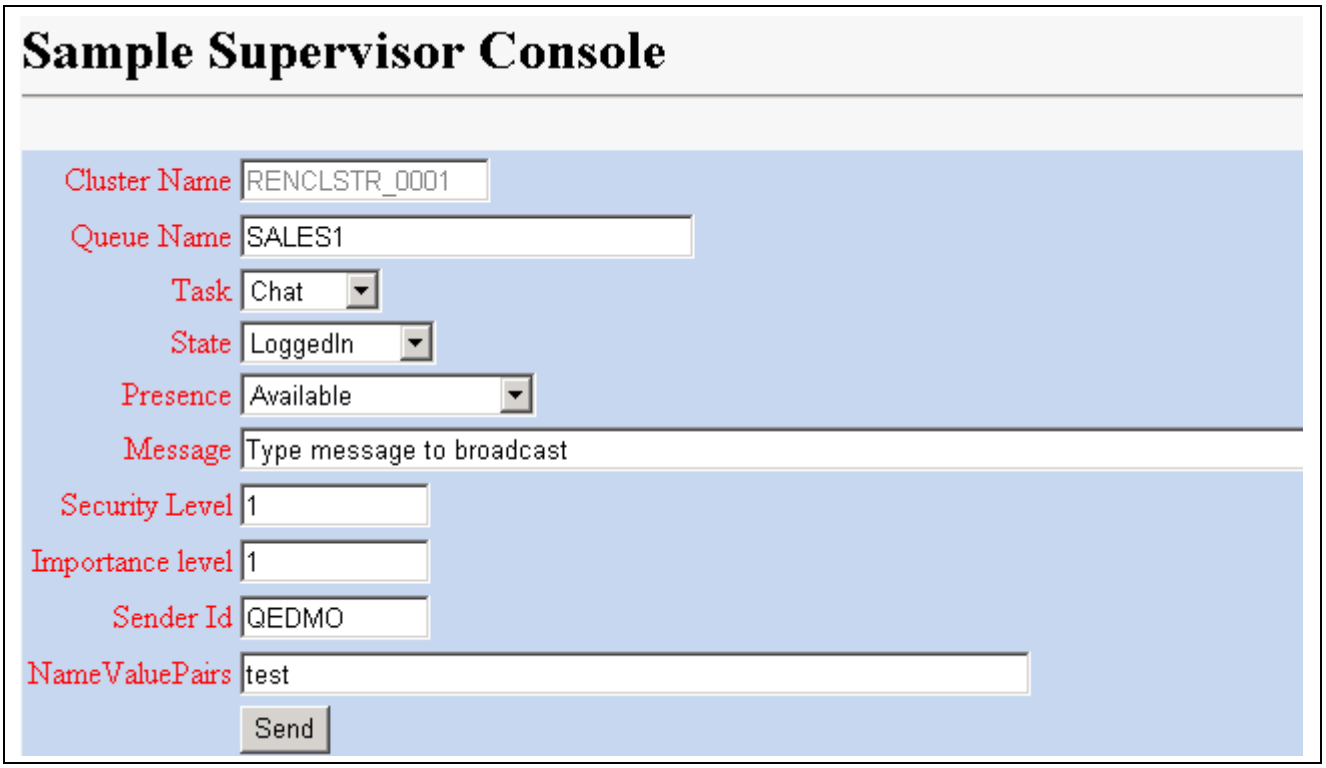
After you click MCF Supervisor console, a new browser window appears that displays the sample supervisor console.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.

MCF Supervisor Console

Access the MCF Supervisor console.

Copyright © 1988-2007, Oracle. All rights reserved.



MCF Supervisor Console

Cluster Name Displays the REN server cluster on which to test broadcast.

Queue Name Enter the name of the queue that is in the specified cluster.

Note. Enter the queue name only when you want to broadcast the message to a particular queue.

Task Select the task. Values are *email, chat, voice, generic,* and *none.*

State Select the state of the agent.

Note. If the state of the agent is *LoggedIn* when the broadcast message is being sent by the supervisor, the agent receives the broadcast message using Agent console.

Presence Select the presence.

Note. If the agent is logged out, the presence should be inActive. The broadcast message is received only when the agent is currently logged in to the queue on which broadcast is sent.

Message Enter the message that you want to broadcast.

Note. To view the broadcast message that is sent by the supervisor, use the Agent console sample page. Select the same REN cluster that is used by the supervisor to broadcast, and the broadcast message is displayed on the Agent console.

Copyright © 1988-2007, Oracle. All rights reserved.

See Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the Agent Console Page, page 167.

Security Level (Optional) Enter the security level.

Importance Level (Optional) Enter the importance level.

Sender Id (Optional) Enter the sender ID. Use this option only when you want the sender ID to appear as something other than the ID that actually sent the message.

NameValue Pairs Enter the name and value pairs to configure your data.

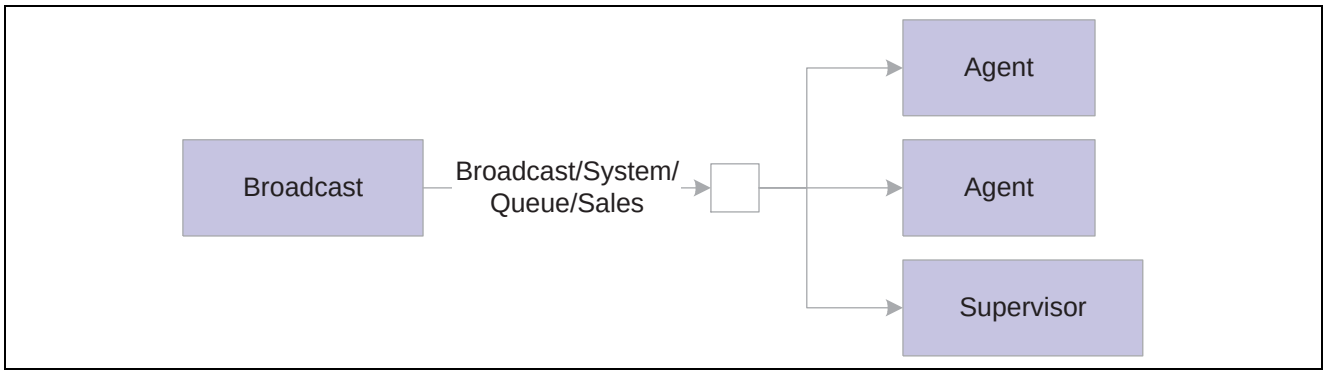
Note. These name-value pairs are concatenated as a string. You can define any name-value parameters for your application and send them in the name-value pair string. The JSMCAPI passes the same as a string to the console. Your particular application needs to process that string accordingly.

Note. For a system-wide or a cluster-wide broadcast, the broadcast message from the supervisor is sent to all agents, as well as the supervisors who are logged in to the cluster that is specified by the supervisor. The system-wide broadcast is same as cluster-wide broadcast because a JSMCAPI supervisor knows only the cluster it is operating on.

For a queue-wide broadcast, the broadcast message from the supervisor is sent to all agents who are currently logged in to the queue that is used by the supervisor to broadcast the message. This assumes that the queue is in the same cluster as the supervisor who is sending the broadcast message.

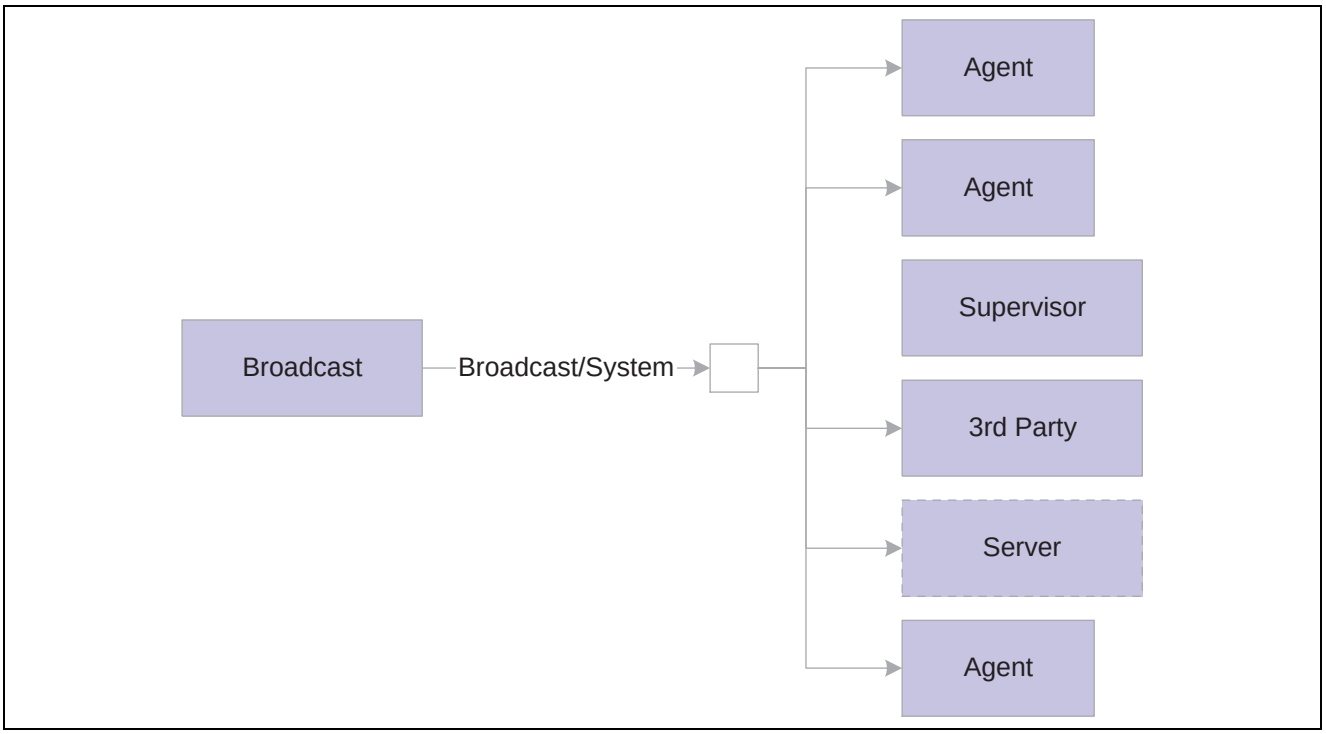
For agent-wide broadcast, the broadcast message will be sent to all agents who are currently logged to any queue in the same cluster as the supervisor who is sending the message.

The following diagrams illustrate queue-wide and cluster-wide broadcast:



Queue-wide broadcast

Copyright © 1988-2007, Oracle. All rights reserved.



System-wide broadcast

The following table describes the combinations:

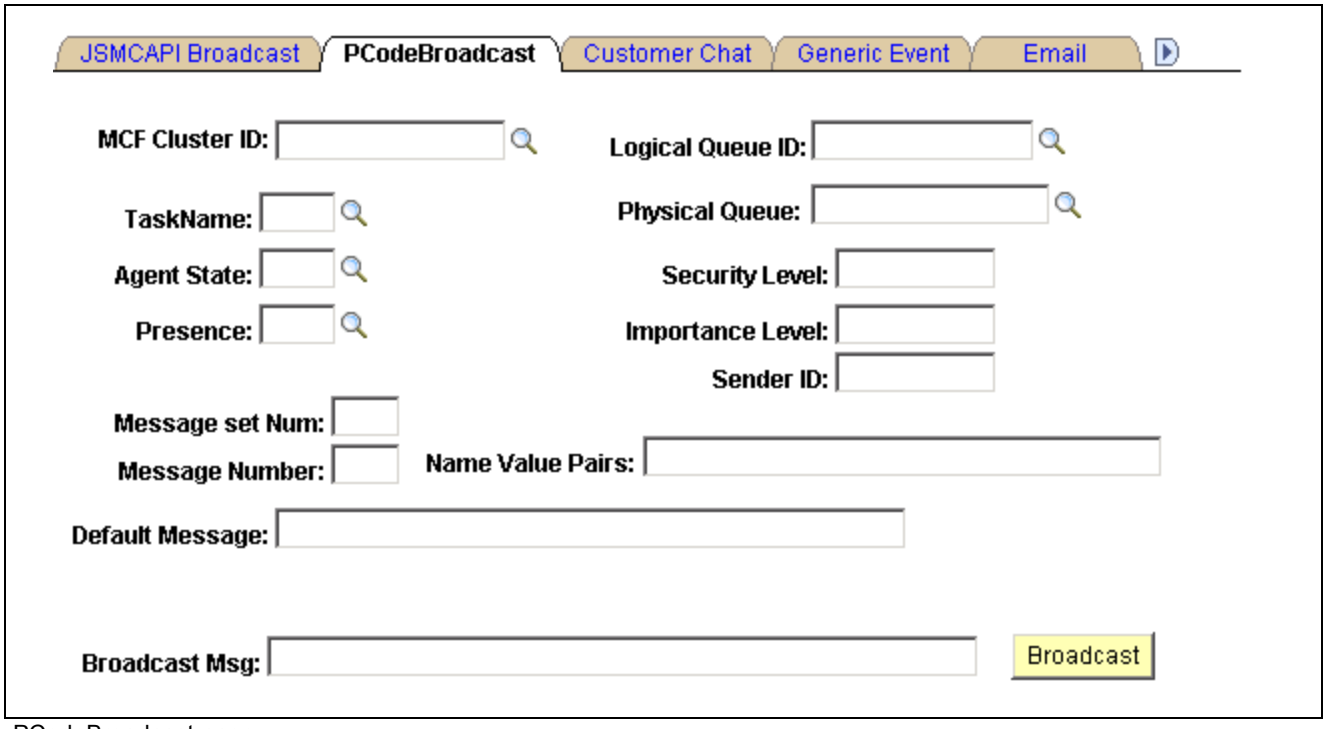
Type of Broadcast	Cluster	Queue	Agent Login State	Agent Activity State	Broadcast Audience and Description
JSMCAPI	Not specified	Not specified	Not specified	Not specified	This is a cluster-wide broadcast. All agents on Cluster_1 will receive the broadcast.
JSMCAPI	Cluster_1	Not specified	Not specified	Not specified	Same as the preceding. All agents on Cluster_1 will receive the broadcast.

Copyright © 1988-2007, Oracle. All rights reserved.

Type of Broadcast	Cluster	Queue	Agent Login State	Agent Activity State	Broadcast Audience and Description
JSMCAPI	Cluster_1	SALES	Not specified	Not specified	All agents subscribing to SALES will receive the broadcast. This assumes that SALES is on Cluster_1.
JSMCAPI	Cluster_1	SALES	Logged	Not specified	All agents who are logged (active and inactive) to sales will receive the broadcast. This assumes that SALES is on Cluster_1.

Using PeopleCode Broadcast

Access the PCodeBroadcast page.



PCodeBroadcast page

MCF Cluster ID Select the target cluster receiving the broadcast message.

Logical Queue ID Select the name of the target logical queue.

Physical Queue ID Select the name of the target physical queue.

Copyright © 1988-2007, Oracle. All rights reserved.

TaskName Select the task. Select from *email, chat*, *voice*, *generic* or *none.*

Agent State Select the state of the agent.

Note. The agent receives the broadcast message only if the agent is logged in.

Presence Select the presence.

Note. If the agent is logged out, the presence should be *inActive.* All agents who are currently logged in receive the broadcast message.

Message Set Number Select the message set number if you want to broadcast a message from the Message Catalog.

Message Number Select the message number in the message set.

Default Message Enter the default message.

If no message is found in Message Catalog with the preceding message number, the text of the default message is used for the broadcast instead.

Security Level (Optional) Enter the security level.

Importance Level (Optional) Enter the importance level.

Sender Id Enter the sender ID if you want the sender that is displayed with the notification to be an ID other than the one that sent the broadcast.

Message Enter the message that you want to broadcast.

Note. To view the broadcast message that is sent by the supervisor, use the Agent console sample page. Select the same REN cluster as used by the supervisor to broadcast, and the broadcast message is displayed on the Agent console.

See Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the Agent Console Page, page 167.

NameValue Pairs Enter the name and value pairs to configure your data.

Note. These name-value pairs are concatenated as a string. You can define any name-value parameters for your application and send them in the name-value pair string. The JSMCAPI passes the same as a string to the console. Your particular application needs to process that string accordingly

Copyright © 1988-2007, Oracle. All rights reserved.

Note. For a system-wide PeopleCode Broadcast, the broadcast message from the supervisor is sent to all agents, as well as the supervisors on all active REN clusters.

For a cluster-wide PeopleCode Broadcast, the broadcast message from the supervisor is sent to all agents, as well as the supervisors on the specified REN clusters.

For a queue-wide broadcast, the broadcast message from the supervisor is sent to all agents who are currently logged in to the queue that is specified by the queue ID in all clusters if the MCF cluster ID is not specified. If the cluster ID is specified, all agents that are logged into the queue that is specified by the queue ID in the cluster that is specified by the cluster ID receive the broadcast message.

For agent-wide broadcast, if both the cluster ID and queue ID are specified, the broadcast message is sent to all agents that are currently logged into the queue that is specified by the queue ID. This assumes that the queue is in the cluster that is specified by the cluster ID. However, if only the cluster ID is mentioned, all agents that are logged and active on any queue on that cluster receive the broadcast message.

The following table describes the combinations:

Type of Broadcast	Cluster	Queue	Agent Login State	Agent Activity State	Broadcast Audience and Description
PeopleCode	Not specified	Not specified	Not specified	Not specified	This is a true system-wide broadcast. All agents on all active REN clusters receive the broadcast.
PeopleCode	Cluster_1	Not specified	Not specified	Not specified	All agents on Cluster_1 receive the broadcast.
PeopleCode	Cluster_1	SALES	Not specified	Not specified	All agents subscribing to SALES receive the broadcast. This assumes that SALES is on Cluster_1.
PeopleCode	Not specified	SALES	Logged	Not specified	All agents that are logged (active and inactive) to SALES (all clusters with physical queues that are associated with SALES) receive the broadcast.

Copyright © 1988-2007, Oracle. All rights reserved.

Type of Broadcast	Cluster	Queue	Agent Login State	Agent Activity State	Broadcast Audience and Description
PeopleCode	Cluster_1	SALES	Logged	Active	All agents that are logged into SALES and are in active state receive the broadcast. This assumes that SALES is on Cluster_1.
PeopleCode	Cluster_1	Not specified	Logged	Active	All agents that are logged and active on any queue on Cluster_1 receive the broadcast.
PeopleCode	Cluster_1	Not specified	Not specified	Active	All agents that are logged and active on any queue on Cluster_1 receive the broadcast. This is the same as the preceding because only agents that are logged in can be active.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, "PeopleCode Built-in Functions," MCFBroadcast

Viewing Broadcast Logs

To view broadcast log, use the Broadcast log page.

See Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Broadcast Logs, page 120.

Working with Sample Pages

To demonstrate MCF tools and functionality, use the MCF Sample Pages (MCF_DEMO_CMP) component.

This section discusses how to:

- Use the Customer Chat sample page.
- Use the URL wizard.
- Use the Generic Event sample page.

Copyright © 1988-2007, Oracle. All rights reserved.

- Use the Generic Event window.
- Use the Email sample page.
- Use the Email window.

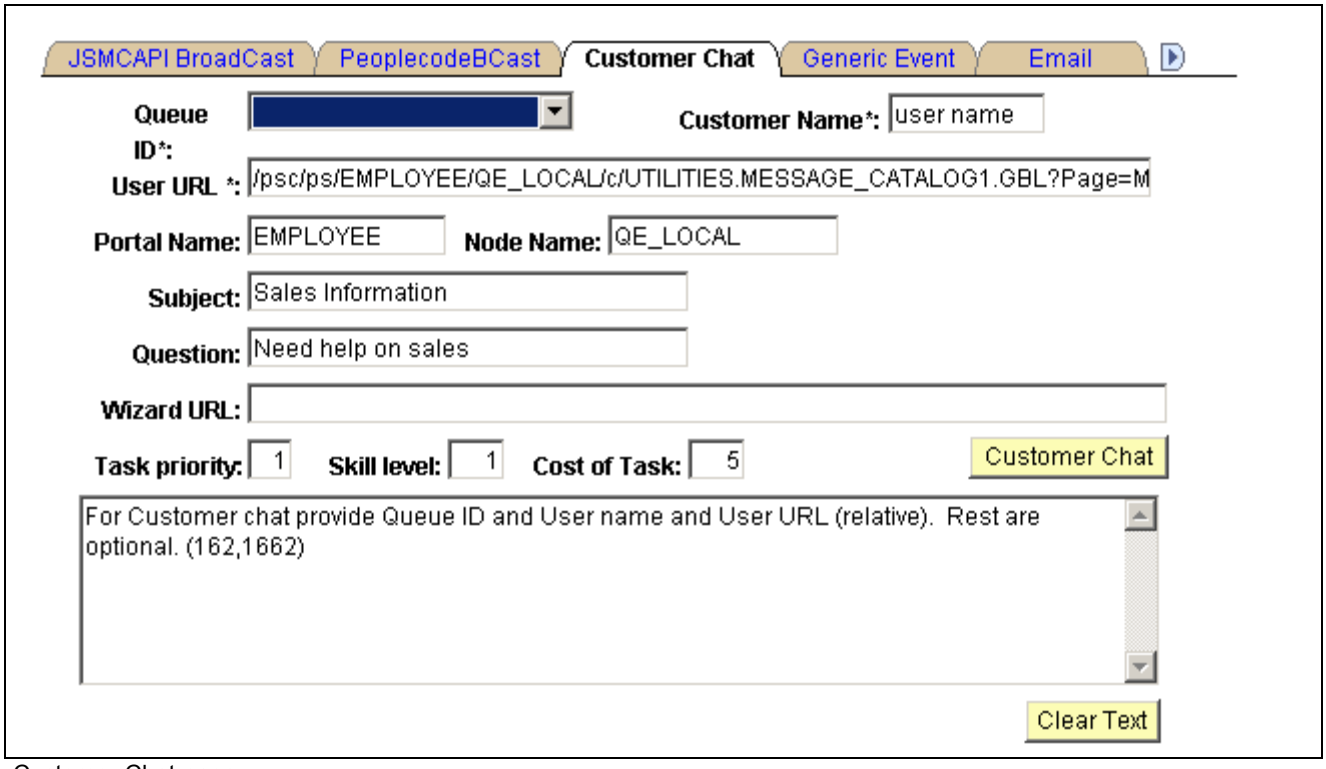
Pages Used to Work with Sample Pages

Page Name	Object Name	Navigation	Usage
Customer Chat	MCF_DEMO_PG	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, Customer Chat	View a customer chat session and built-in functions.
Generic Event	MCF_DEMO_NOTIFY_PG	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, Generic Event	View a generic event notification and built-in functions.
Email	MCF_DEMOEM_NTFY_PG	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, Email	View email retrieval, built-in functions, and application package classes.

Using the Customer Chat Sample Page

Access the Customer Chat page.

Note. This page is available using both queue server and third-party routing server.



Customer Chat page

This page demonstrates the InitChat() built-in function, including required and optional parameters.

Copyright © 1988-2007, Oracle. All rights reserved.

This page is not intended for production use. On a typical application page, the developer includes a Live Help or Customer Chat button that calls the InitChat() built-in function and passes required parameters, including the context of the chat request. The sample customer chat page demonstrates values that can be included in the InitChat() parameters. The user may not be prompted for any information.

To run a sample chat:

1. Access the MultiChannel Console.
2. Open the Customer Chat page.
3. Open the Queue ID drop-down list box and select the queue that you are logged on to.

   The other fields are automatically populated with values that generate a sample chat session. You can change the values as long as valid required values are included.

   The User URL is a required field that contains the application page URL to send to the agent chat browser.

   The optional Wizard URL field represents a grab URL; the actual page that appears is determined by InitChat() parameters.

   The text box displays error messages and additional information for using the sample page.
4. Click Customer Chat.

   The customer chat window appears.
5. On the MultiChannel Console, click the flashing chat notification icon to accept the chat session.

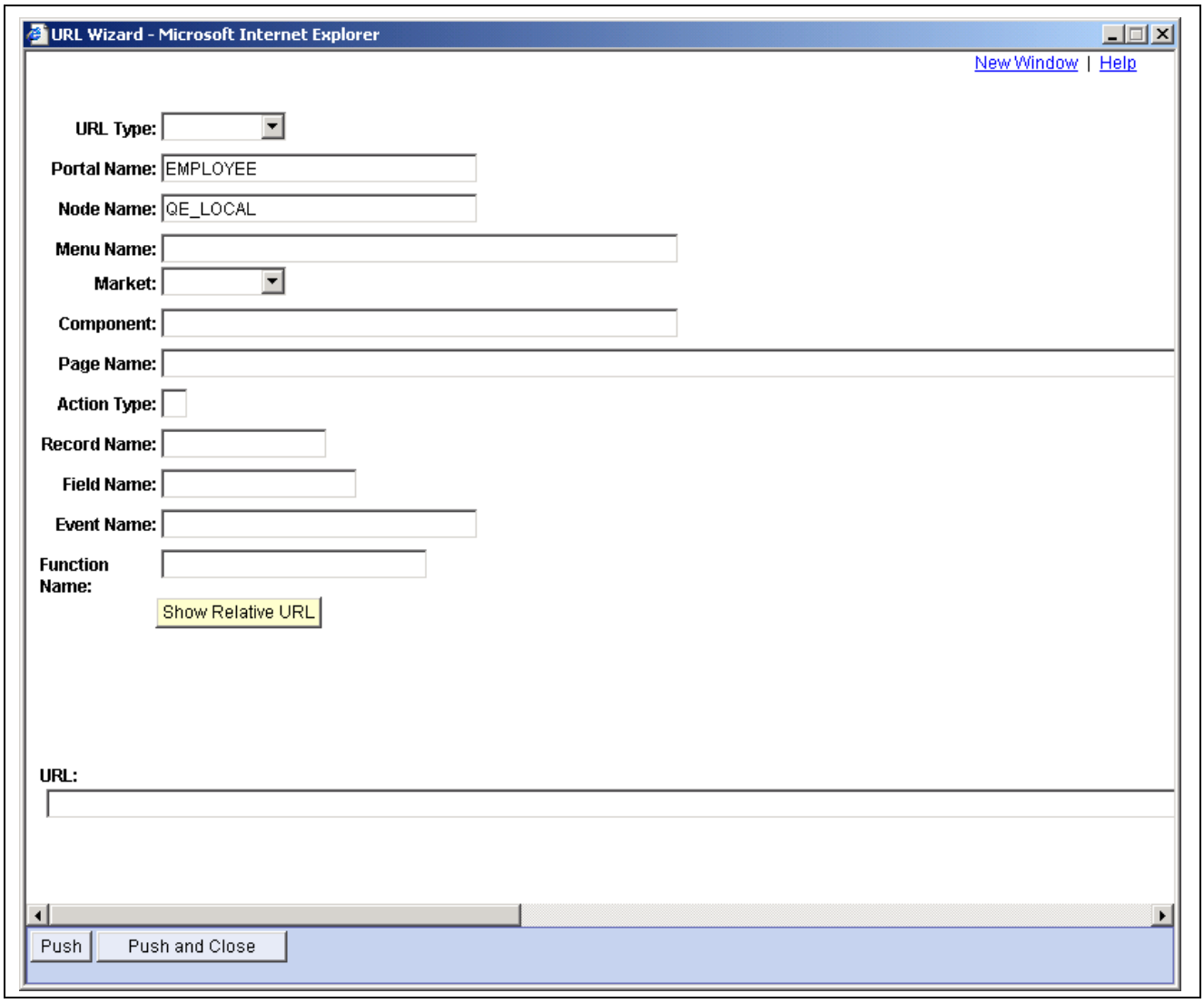
   The agent chat window appears.

   You can enter text as both agent and customer to demonstrate chat functionality.

Using the URL Wizard

Click Select from the agent chat window during a chat session to open the URL wizard:

Copyright © 1988-2007, Oracle. All rights reserved.



URL Wizard page

Use the wizard to form a URL to send to the customer. The URL Wizard page is a template that demonstrates constructing a URL and sending it to a customer. The URL wizard can be used as is or as the basis for developing URL generation for your application.

URL Type Select the type of URL that is being accessed. Values are:

- *Component:* Generates a PeopleSoft Pure Internet Architecture component URL.

  Component URL generation uses the PeopleCode built-in function **GenerateComponentContentRelURL.**

  See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions".
- *External:* Not currently used.
- *Gen URL:* Not currently used.
- *iScript:* Generates a PeopleSoft Pure Internet Architecture iScript URL.

 Copyright © 1988-2007, Oracle. All rights reserved.

iScript URL generation uses the PeopleCode built-in function **GenerateScriptContentRelURL.**

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions".

Fields that are not required by the selected URL type are not available.

Show Relative URL Click to display the relative URL that is generated by the values that you have entered into the page's fields.

URL Displays the generated relative URL.

Push Click to send the generated relative URL to the customer's chat window.

The Push button demonstrates functionality that must be included in an application-specific URL wizard.

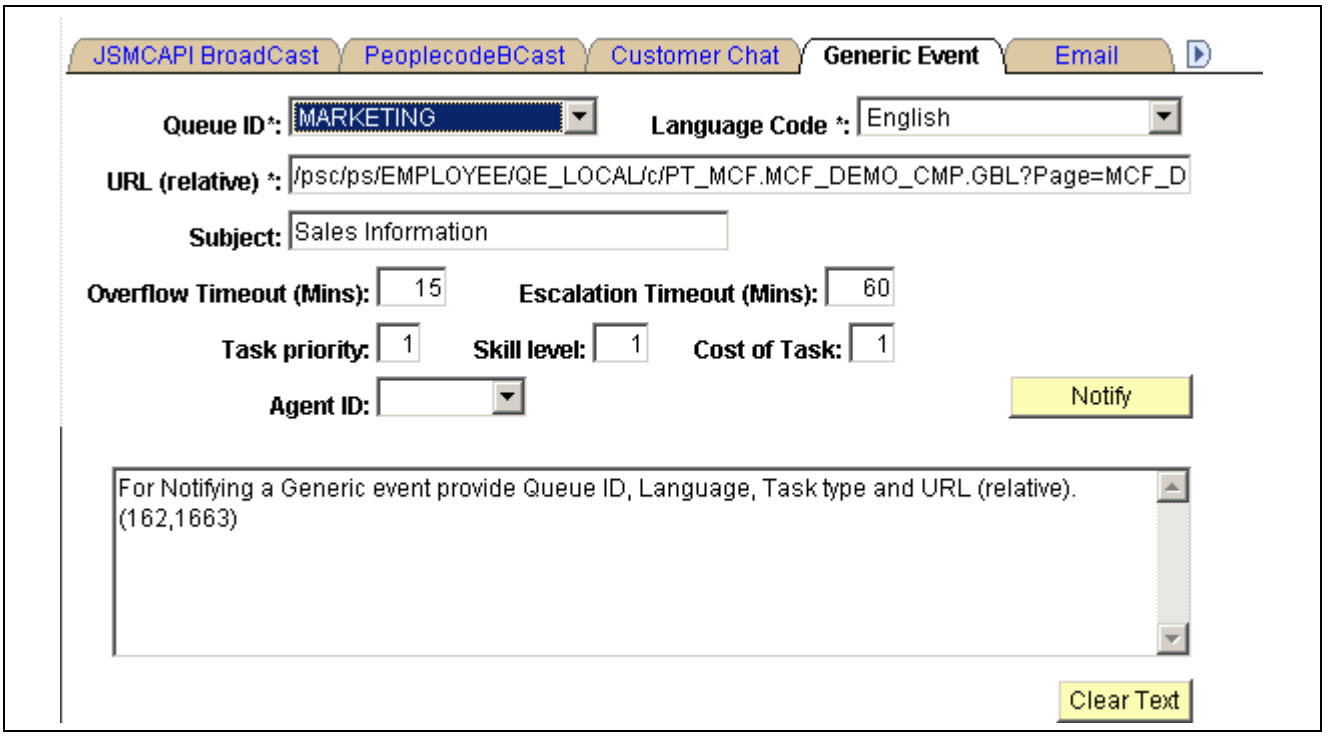
Push and Close Click to send the generated relative URL to the customer's chat window and close the URL wizard.

The Push and Close button demonstrates functionality that must be included in an application-specific URL wizard.

Using the Generic Event Sample Page

Access the Generic Event page.

Note. This page is available using both queue server and third-party routing server.



Generic Event page

This page demonstrates a generic persistent event using the EnQueue() and NotifyQ() built-in functions, including required and optional parameters.

Copyright © 1988-2007, Oracle. All rights reserved.

This page is not intended for production use. On a typical application page, the developer includes an Enqueue or similar button that calls the EnQueue() built-in function and passes required parameters, including the context of the request. The sample customer chat page demonstrates values that can be included in the EnQueue() parameters. The user may not be prompted for any information.

To run a sample generic event:

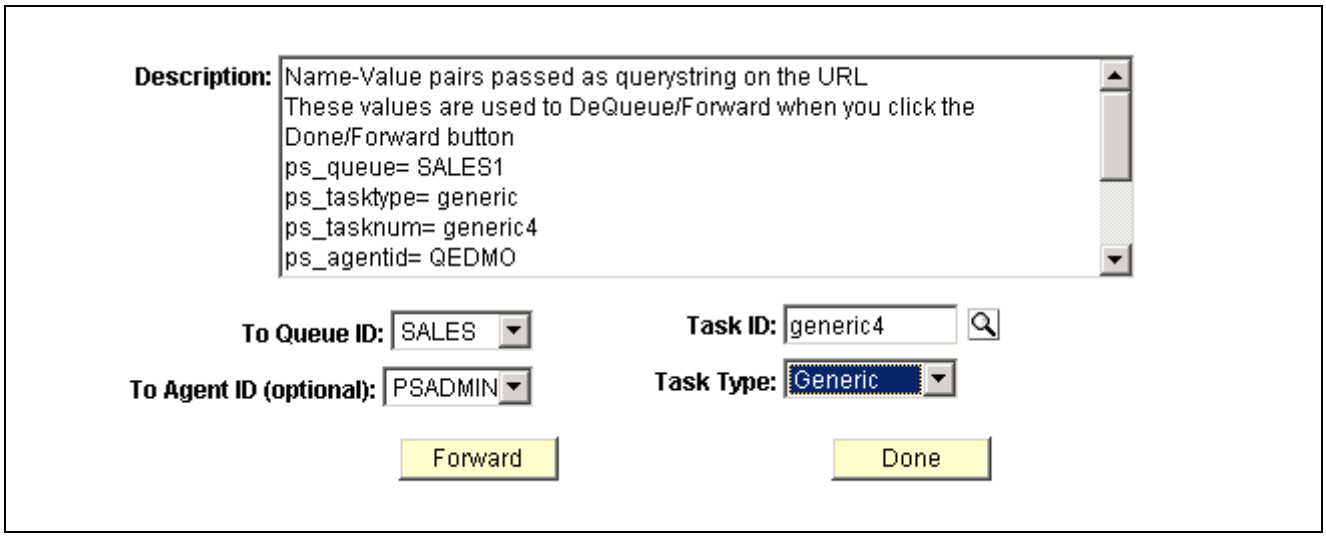
1. Access the MultiChannel Console.
2. Open the Generic Event page.
3. Open the Queue ID drop-down list box, and select the queue that you are logged on to.
4. Open the Agent ID drop-down list box, and select the agent ID that you are logged on with.
5. Select *Generic* for the task type.
6. Click Notify.
7. On the MultiChannel Console, click the flashing event notification icon to accept the event.

   The Generic Event window appears.

Using the Generic Event Window

The format of the generic event window is determined by application developers. PeopleSoft supplies a sample generic event window to demonstrate the available functionality, including the DeQueue() built-in.

Configure the size and initial location of the generic event window on the Agent Window Configuration page:



Generic Event window

Description	Displays text from the generic event.
To Queue ID	Select a queue to which the generic event will be forwarded.
Task ID	Select the task ID of the generic event to forward.
To Agent ID	Select the agent ID to whom the generic event will be forwarded.
Task Type	Select the task type of the generic event to be forwarded.
Forward	Click to send the generic event to the selected agent or queue.

Copyright © 1988-2007, Oracle. All rights reserved.

This button demonstrates the functionality of the Forward() built-in function.

Done Click to notify the real-time event notification (REN) server that you are done with this task. A message asks if you want to close the window.

The Done button demonstrates the functionality of the DeQueue() built-in function.

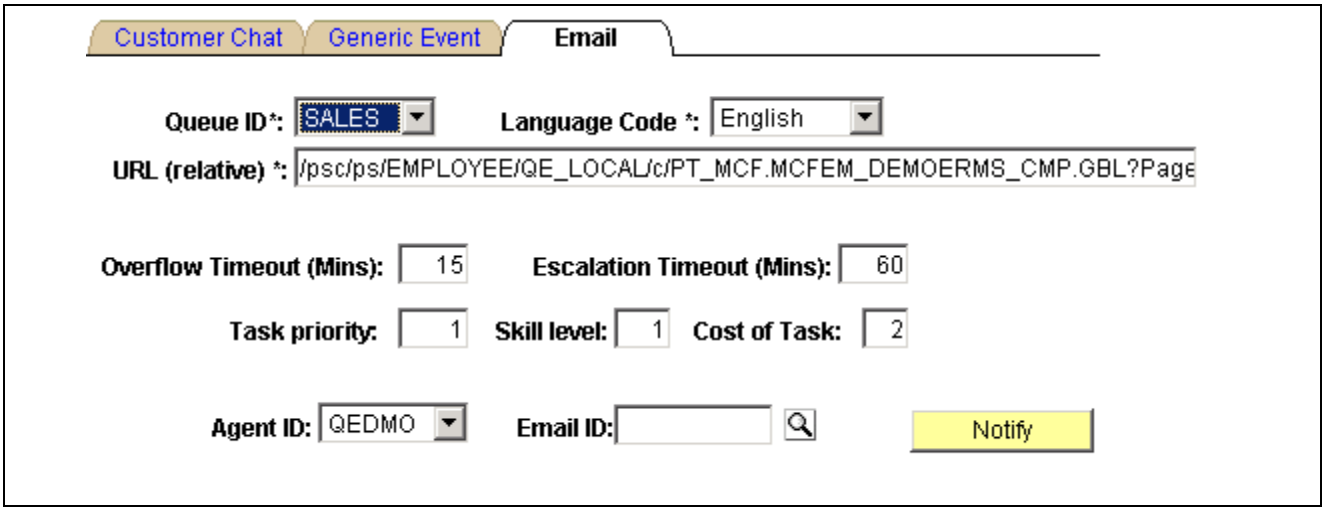
See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, "PeopleCode Built-in Functions"

Using the Email Sample Page

Access the Email page.

Note. This page is available using both queue server and third-party routing server.



Email page

The email sample pages are intended for demonstration purposes and should not be used in production.

EnQueue() and NotifyQ() can also be called from PeopleSoft Application Engine batch programs.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions".

Note. For you to fully demonstrate the functionality of the Email sample page, an email must be read and written to the email database using the GetMail - Server sample page.

To process a sample email:

1. Open the Queue ID drop-down list box and select the queue that you are logged on to.
2. Open the Agent ID drop-down list box and select the agent ID that you are logged on with.
3. Click Notify.
4. On the MultiChannel Console, click the flashing email notification icon to accept the email.

   The email window appears.

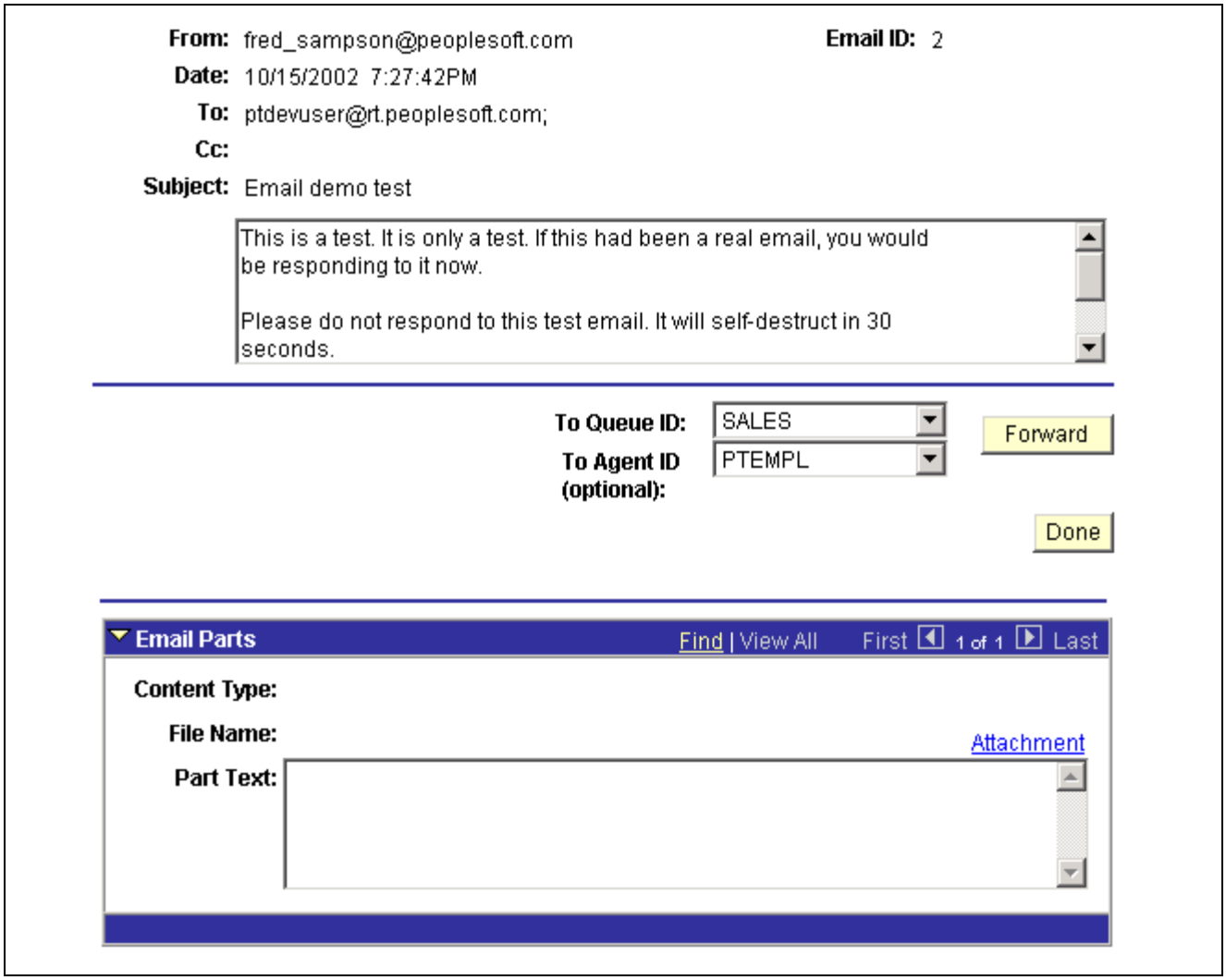
Copyright © 1988-2007, Oracle. All rights reserved.

Using the Email Window

The format of the email window is determined by application developers. PeopleSoft supplies a sample email event window to demonstrate available functionality.

Configure the size and initial location of the email event window on the agent's Window Configuration page.

The following example shows an email window:



Email window

To Queue ID To forward the email to another queue, select a queue ID from the drop-down list box and click Forward.

To Agent ID To forward the email to another agent, select an agent ID from the drop-down list box and click Forward.

Forward Click to forward the email to the specified queue or agent.

This button demonstrates the functionality of the Forward() built-in function.

Done Click to quit the email and remove it from the queue.

The Done button demonstrates the functionality of the DeQueue() built-in.

Copyright © 1988-2007, Oracle. All rights reserved.

Email Parts

If an email has been divided into parts stored in the email database, or has an attachment, they can be accessed here.

See Also

Chapter 14, “Configuring the Email Channel,” Demonstrating the Email Channel, page 410

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference, “Mail Classes”

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “PeopleCode Built-in Functions”

Using and Demonstrating JSMCAPI

To demonstrate MCF consoles and tools that use JSMCAPI, use the CTI Sample Pages (PT_CTI_DEMOOUTB) and MCF Sample Pages (MCF_DEMO_CMP) components.

This section provides an overview of JSMCAPI and discusses how to:

- Use the CTI Sample Console page and Sample CTI Console.
- Use the Agent Console page and Sample UQ Agent Console.
- Use the Monitor Agents page and Sample Monitor - Agent States page.
- Use the Monitor Queues page Sample Monitor - Queue Statistics page.

Understanding JSMCAPI

The JSMCAPI enables custom configuration of MCF consoles (including the CTI console), MCF functionality on a PeopleSoft Pure Internet Architecture page, and queue and agent monitoring. For example, developers can create supervisor dashboards with which to monitor activity on their channels of interest, or developers can modify consoles according to their company’s business requirements. The JSMCAPI builds on the REN JavaScript client. JSMCAPI uses standard JavaScript.

PeopleSoft provides sample pages demonstrating the functionality that is enabled by JSMCAPI. These pages are for demonstration purposes only, and should not be used in a production environment.

JSMCAPI is delivered with PeopleTools; you do not need a separate installation. The jsmcapi.js file is located in the <PS_HOME>\webserv\<domain>\applications\peoplesoft\PORTAL\ps\pMCF folder.

Understanding Reason Objects and Reason Codes

PeopleSoft MCF provides reason codes to explain any error or an event that has happened. The third party vendors uses the same reason codes as provided by PeopleSoft MCF to create reason objects that they send with the events. PSMCAPI internally parses the event, gets the reason code, and sends it to JSMCAPI with the event. The CTI sample console or third-party multichannel console looks up a mapper table that maintains the mapping of the reason codes to the Message Catalog entries and displays the reason code messages associated with the error or event that was sent by PSMCAPI.

To display the reason code messages, the sample console picks up all the reason codes and reason messages from the Message Catalogue. Whenever an error occurs, the JavaScript function will extract the reasoncode from the reason object in event and look for the reason message. Then, the JavaScript code prepares the reason message from the fields of reason object and displays the reason message in a new small window.

Copyright © 1988-2007, Oracle. All rights reserved.

Reason codes can also be send with requests from JSMCAPI side. The JavaScript function will prompt for the reason codes and any extra data as part of reason.

The following table lists the reason codes and their corresponding reason messages:

Reason Code	Description/Message Entry
0	System or general error has occurred.
1	See provider for details.
2	Extension or agent is busy.
3	No answer.
4	Task is in transfer.
5	Task is in conference.
6	Task is abandoned.
7	Searching for an agent.
8	Incoming task.
9	Task is assigned.
10	Connection is established.
11	User or Task data is updated.
12	Request is completed successfully.
13	State of the agent, group, or extension changed.
14	New task is initiated.
15	Conference is not successful.
16	Transfer is not successful.
17	Retrieve is not successful.
18	Forward is not successful.
19	Reject is not successful.
20	Revoke is not Successful.
21	Task is revoked.
22	Task is withdrawn.
23	Do Not Disturb is turned on.

Copyright © 1988-2007, Oracle. All rights reserved.

Reason Code	Description/Message Entry
24	Do Not Disturb is turned off.
25	Forwarding of calls is set.
26	Forwarding of calls is canceled.
27	Phone is on hook.
28	Phone is off hook.
29	New call is initiated.
30	Call is put on hold.
31	Call is parked.
32	Call is cleared.
33	Call is alternated between hold and active states.
34	Call is taken off hold.
35	Hold is not successful.
36	Invalid session.
37	Invalid State.
38	Incorrect Data.
39	Unsupported Feature.
40	Configuration Error.
41	Server is on.
42	Server is off.
43	Agent is Busy.
44	Unexpected Error.

The following table maps the reason codes to the entries in the Message Catalog:

Message Number MCFMSGNUM	Message Set Number MCFMSGSETNUM	Reason Code Number REASONCODENO
2790	162	0
2791	162	1
2792	162	2

Copyright © 1988-2007, Oracle. All rights reserved.

Message Number MCFMSGNUM	Message Set Number MCFMSGSETNUM	Reason Code Number REASONCODENO
2793	162	3
2794	162	4
2795	162	5
2796	162	6
2797	162	7
2798	162	8
2799	162	9
2800	162	10
2801	162	11
2802	162	12
2803	162	13
2804	162	14
2805	162	15
2806	162	16
2807	162	17
2808	162	18
2809	162	19
2810	162	20
2811	162	21
2812	162	22
2813	162	23
2814	162	24
2815	162	25
2816	162	26
2817	162	27
2818	162	28

Copyright © 1988-2007, Oracle. All rights reserved.

Message Number MCFMSGNUM	Message Set Number MCFMSGSETNUM	Reason Code Number REASONCODENO
2819	162	29
2820	162	30
2821	162	31
2822	162	32
2823	162	33
2824	162	34
2825	162	35
2826	162	36
2827	162	37
2828	162	38
2829	162	39
2830	162	40
2831	162	41
2832	162	42
2833	162	43
2834	162	44

See Also

Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Pages Used to Use and Demonstrate JSMCAPI, page 160

Chapter 13, “Understanding JSMCAPI Classes,” page 213

Appendix A, “JSMCAPI Quick Reference,” page 421

Common Elements Used in This Section

This section discusses common elements used on the CTI sample pages.

Common Elements Used in CTI Sample Pages

The following common elements are used in CTI sample pages.

Agent Id Select the CTI or MCF agent’s agent ID.

In the case of a CTI agent, the agent ID may be different from the agent’s user ID.

Copyright © 1988-2007, Oracle. All rights reserved.

g/e/c (generic, email, chat) Represents three task types: generic, email, chat.

Go Click to initiate the action that is selected in the associated drop-down list box.

MCF Cluster ID Select the MCF cluster to be monitored or on which to test the sample console.

Physical Queue Select the physical queue to be monitored.

Each MCF logical queue can comprise one or more physical queues. Because each physical queue is associated with an MCF cluster, only physical queues can be monitored.

State Displays a value indicating the state of the associated element.

Statistics Display statistics that are generated by the CTI server.

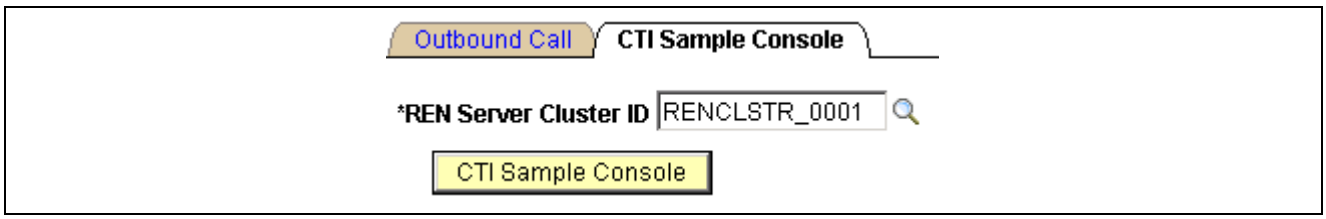
User Id Displays the CTI agent's PeopleSoft user ID.

Pages Used to Use and Demonstrate JSMCAPI

Page Name	Object Name	Navigation	Usage
Agent Console	PT_SAMPLEAGNTCONSL	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, Agent Console	View a sample MultiChannel Console that is generated using JSMCAPI.
CTI Sample Console	CTI_SAMPLECONSOLE	PeopleTools, MultiChannel Framework, CTI Configuration, Sample Pages, CTI Sample Console	View a sample CTI console that is generated using JSMCAPI.
Monitor Agents	PT_UQSAMPLECNSL	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, Monitor Agents	View a sample agent statistics monitoring page.
Monitor Queues	PT_UQSAMPLECNSQ	PeopleTools, MultiChannel Framework, Universal Queue, Sample Pages, Monitor Queues	View a sample queue statistics monitoring page.

Using the CTI Sample Console

Access the CTI Sample Console page.



CTI Sample Console page

REN Server Cluster ID Select the REN server cluster on which to test the sample console.

CTI Sample Console Click to initiate the sample console.

Copyright © 1988-2007, Oracle. All rights reserved.

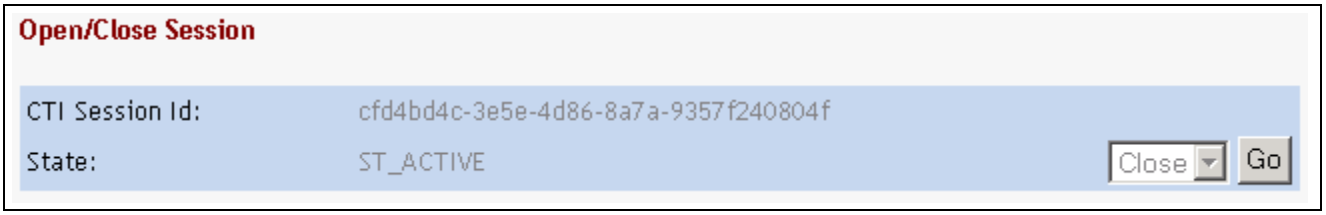
Note. To demonstrate the CTI sample console, you must have a CTI server running and communicating with the specified REN server cluster.

After clicking CTI Sample Console, a new browser window appears that displays the sample console.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.

Open/Close Session

Access the Open/Close Session group box.



Open/Close Session group box

The Open/Close Session group box displays information about the session.

CTI Session Id This field displays a value identifying the session.

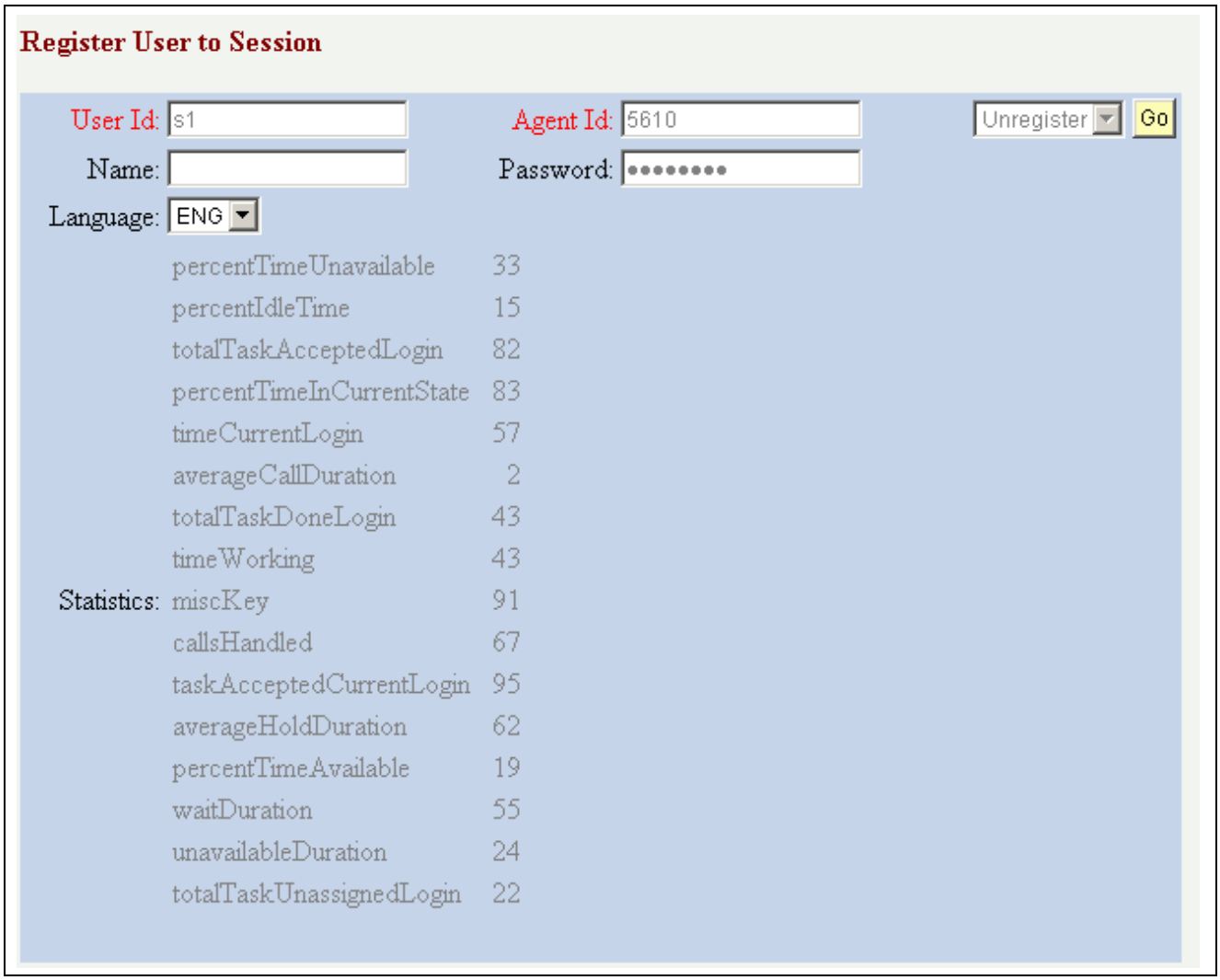
If no session ID is displayed, check to ensure that the CTI server and REN server are both running.

You can close an active session by clicking the Go button. You cannot make a selection from the drop-down list box.

Register User to Session

Access the Register User to Session group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Register User to Session group box

Register a user to the session in the Register User to Session group box.

Because a CTI agent can have an agent ID different from the agent's user ID, both values are required. If the agent has a password, the password is also required.

Language Reserved for future use.

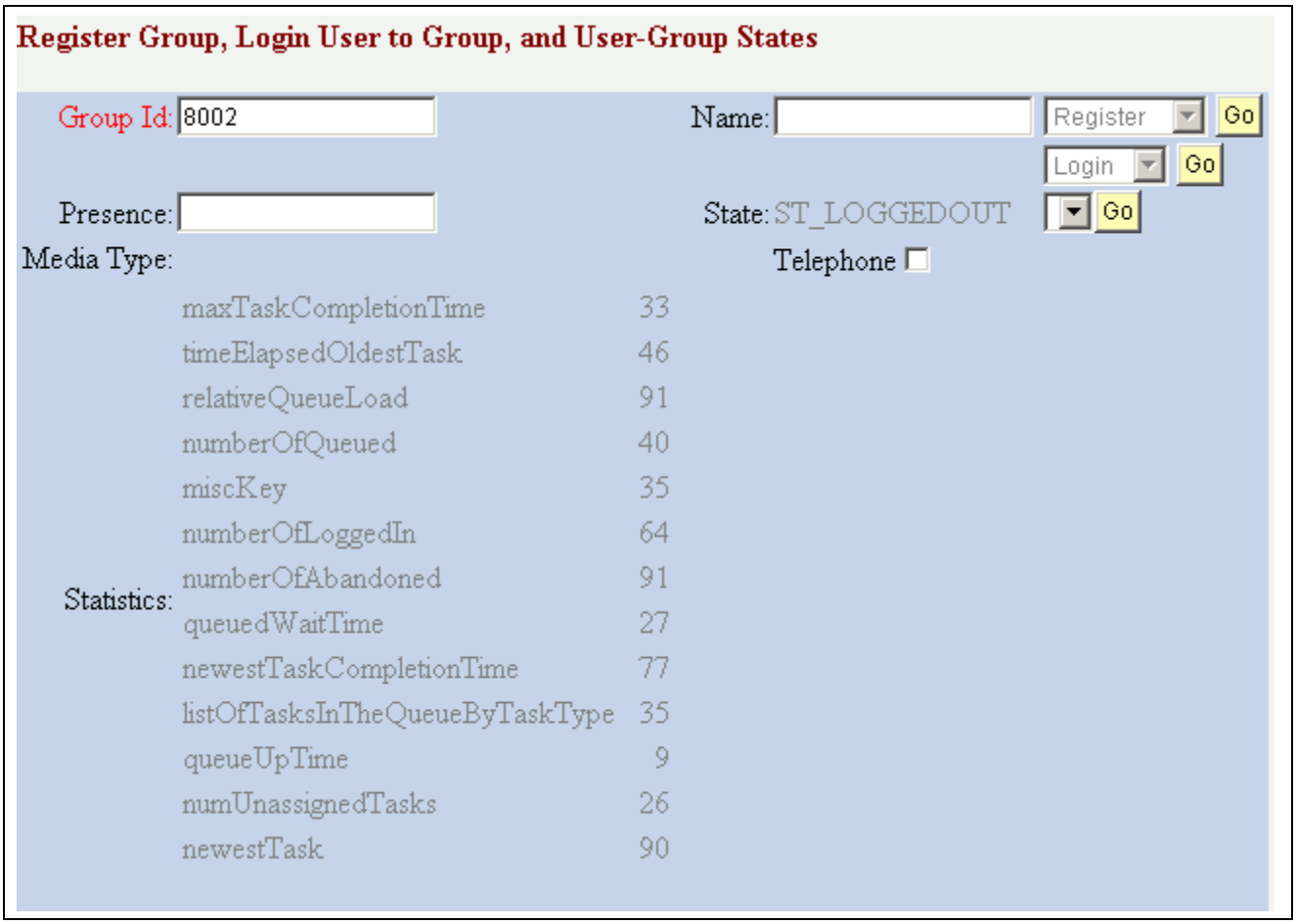
The default language is English.

You can register or unregister an agent by clicking the Go button. You cannot make a selection from the drop-down list box.

Register Group, Login User to Group, and User-Group States

Access the Register Group, Login User to Group, and User-Group States group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Register Group, Login User to Group, and User Group States group box

Group Id

Enter the group ID to which the agent will be registered.

To register the agent with the specified group, click the Go button. You cannot make a selection from the drop-down list box.

Login

Click the Go button to log in if *Login* appears in the drop-down list box.

Click the Go button to log out if *Logout* appears in the drop-down list box.

Note. The agent cannot receive calls until the agent is in the Ready state.

Note. After you login to the group, a new pop-up window opens prompting for reason code data. If you register to an invalid group, the system displays an error message.

See Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Understanding JSMCAPI, page 155.

Presence

Enter a presence description to associate with the state that is selected in the drop-down list box.

This presence value is different from any presence values that are predefined in the agent configuration. The agent cannot select from the pre-defined values.

Copyright © 1988-2007, Oracle. All rights reserved.

State

Select an agent state.

State options are:

- *Ready*
- *Not Ready*
- *Work Ready*
- *Work Not Ready*

Each state value has an associated state code that appears in the State field.

To enable incoming calls, select *Ready* from the drop-down list box and click the Go button.

Media Type

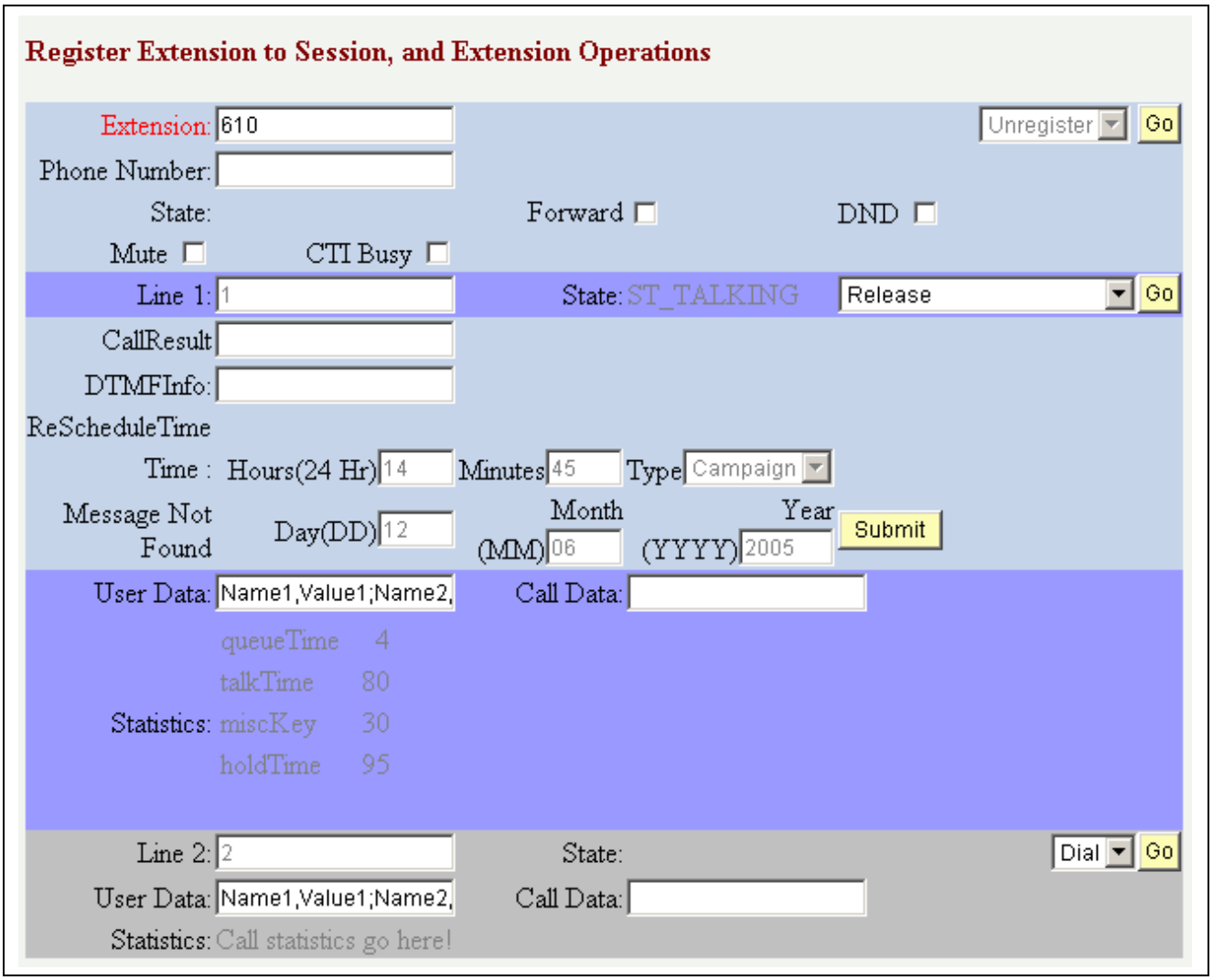
Reserved for future use.

The sample console supports only the telephone (voice) channel. However, JSMCAPI can support other channels.

Register Extension to Session and Extension Operations

Access the Register Extension to Session, and Extension Operations group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Register Extension to Session and Extension Operation group box

Extension Enter a valid telephone extension number.

Phone Number Enter a valid telephone number for use in subsequent operations, such as dial, forward, or do-not-disturb (DND).

Mute Select to mute a call.

Note. This is enabled only when the agent is on a call.

CTI Busy This flag is checked by the queue server for an agent handling a voice task.

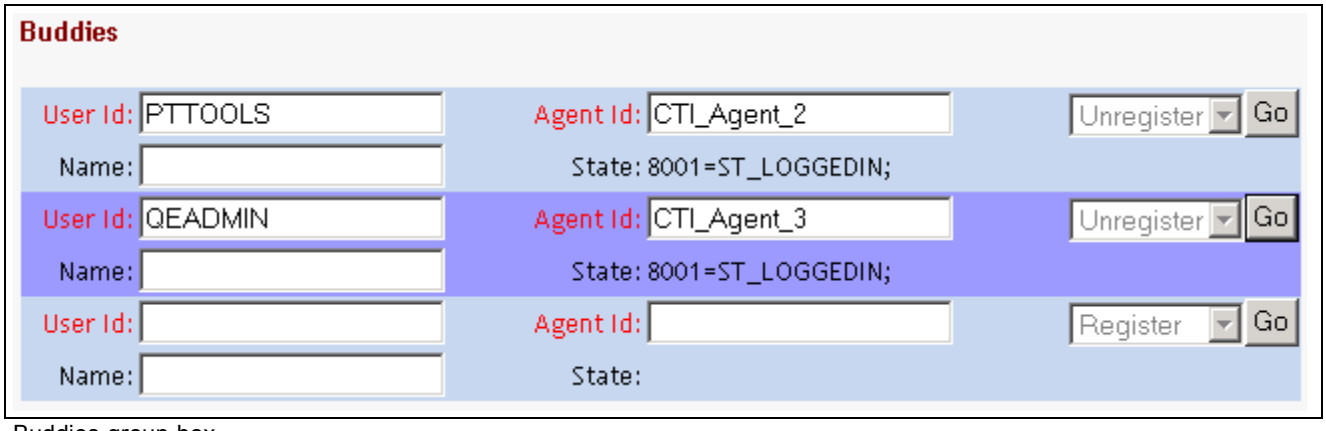
Note. If the flag is checked, the queue server will not assign new tasks to such an agent. The CTI Busy flag allows the application to implement a request which conveys the status of the agent to the queue server. Because the queue server stores the task list and the status in its temporary cache, a recovered or a rebooted queue server will not have the information of a agent's previous CTI status. The agent can send its CTI status to the queue server to indicate that it is busy with a CTI task.

Copyright © 1988-2007, Oracle. All rights reserved.

Forward	Select to forward incoming calls to the specified number.
DND (do not disturb)	Select to put the selected extension in do-not-disturb status.
Line 1 or **Line 2**	Displays a name or value for each line. On the sample console, the value will either be *1* or *2*.
DTMFInfo (Dual-tone Multi frequency information)	Select to send DTMF digits on the line
CallResult	Select to set call result on the line
ReScheduleTime	Select to set reschedule time on the line
Type	Select to set reschedule type on the line.
User Data	Enter name-value pairs representing user data to be attached to a call.
Call Data	Enter any call data to be attached to the call. Call data includes: • *ANI* • *DNIS* • *THISDN* • *OTHERDN*

Buddies

Access the Buddies group box.



Buddies group box

Before an agent can chat with a buddy, the buddy must be registered in the Buddies group box.

Because a CTI agent can have an agent ID that is different from the agent's user ID, both values are required.

Note. If one agent is logged on the multichannel console and another agent is logged on the sample pages, the agents may see an incorrect buddy state for each other.

Error Messages/Information

Access the Error Messages/Information group box.

Copyright © 1988-2007, Oracle. All rights reserved.

Error Messages/Information group box.

Any error messages associated with the process display in this section.

Clear Click the button to clear a displayed error message.

Server State and Broadcasts

Access the Server State and Broadcasts group box.

Server State and Broadcasts group box

CTI State Displays the status of the CTI server.

REN State Displays the status of the REN server.

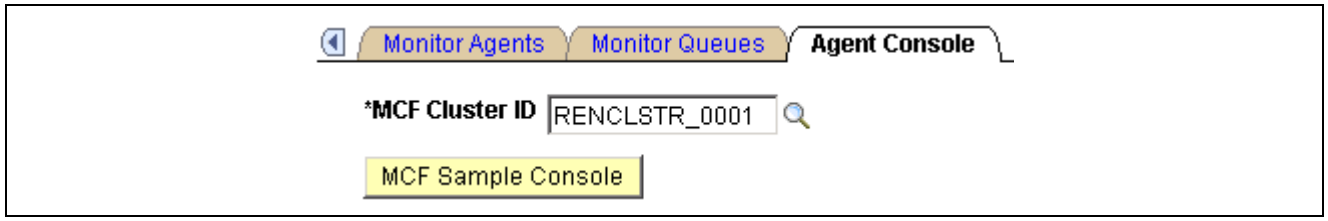
Info (information) Displays the text of a broadcast message from the CTI server.

Tracer

The tracer displays all requests between the CTI server and the console, which can be useful for debugging.

Using the Agent Console Page

Access the Agent Console page.



Agent Console page

MCF Sample Console Click to initiate the sample console.

Note. To demonstrate the MCF sample console, you must have an MCF cluster running and communicating with the specified REN server cluster.

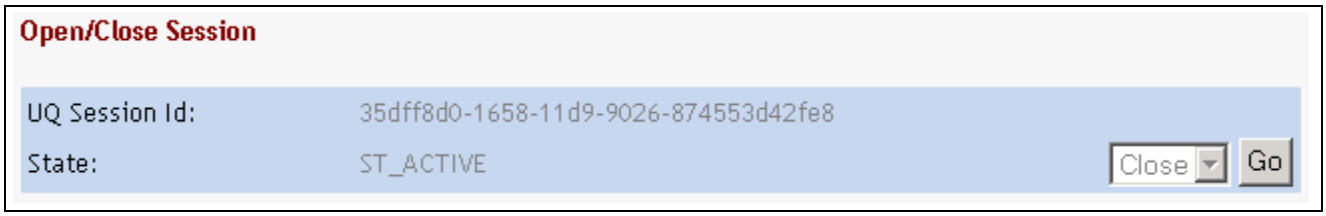
After you click MCF Sample Console, a new browser window appears that displays the sample console.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.

Copyright © 1988-2007, Oracle. All rights reserved.

Open/Close Session

Access the Open/Close Session group box.

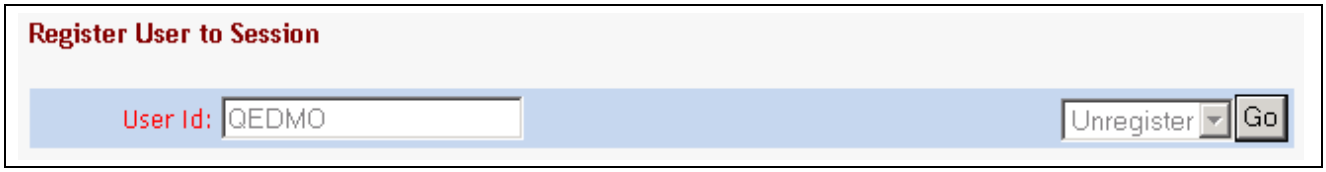


Open/Close Session group box

UQ Session Id The universal queue session ID.

Register User to Session

Access the Register User to Session group box.

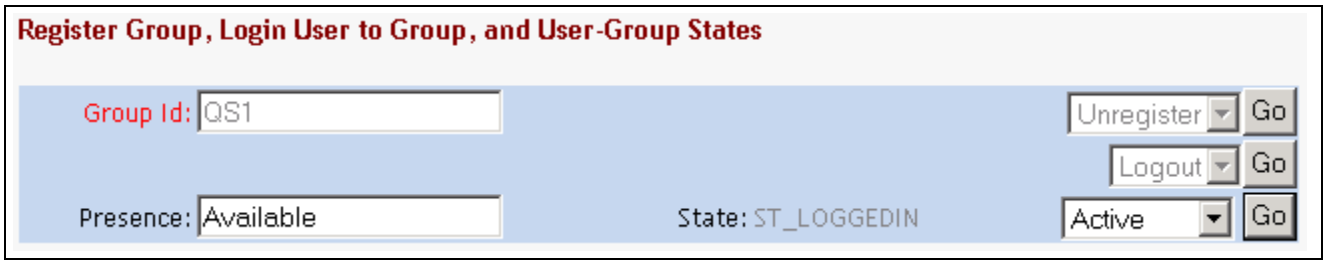


Register User to Session group box

Register Group, Login User to Group, and User-Group States

Access the Register Group, Login User to Group, and User-Group States group box.

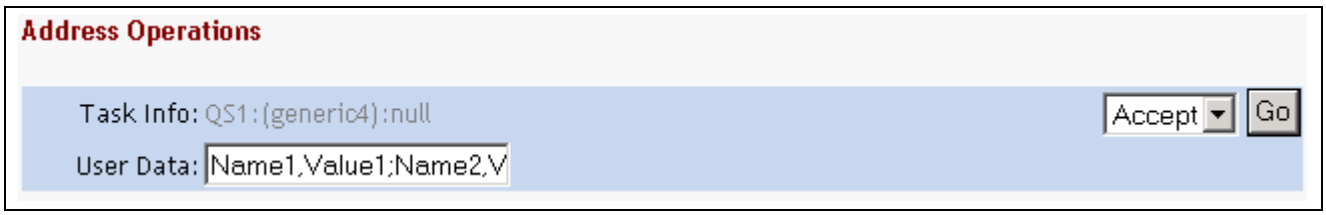
Note. The presence change happens only when you change from active to inactive or from inactive to active. When you log in, the default value is active and therefore no presence change occurs.



Register Group, Login User to Group, and User-Group States group box

Address Operations

Access the Address Operations group box.

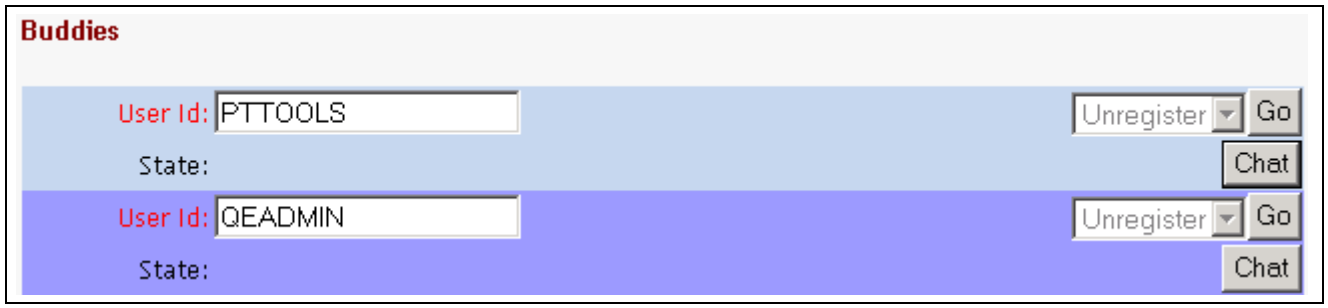


Address Operations group box

Buddies

Access the Buddies group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Buddies group box

Note. If one agent is logged on the multichannel console and another agent is logged on the sample pages, the agents may see an incorrect buddy state for each other.

Note. The buddy states on this page are not automatically displayed. You must monitor an agent first using the Monitor Agents page before the buddy states are displayed.

See Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using the Monitor Agents Page and Sample Monitor - Agent States Page, page 170.

Click to initiate an agent-to-agent chat session.

Error Messages/Information

Access the Error Messages/Information group box.

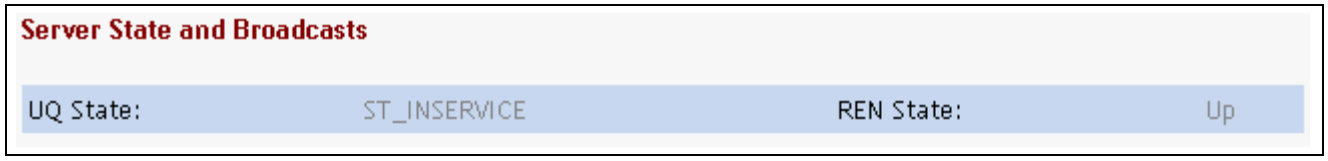
Error Messages/Information group box

Any error messages associated with the process display here.

Clear Click the button to clear a displayed error message.

Server State and Broadcasts

Access the Server State and Broadcasts group box.



Server State and Broadcasts group box

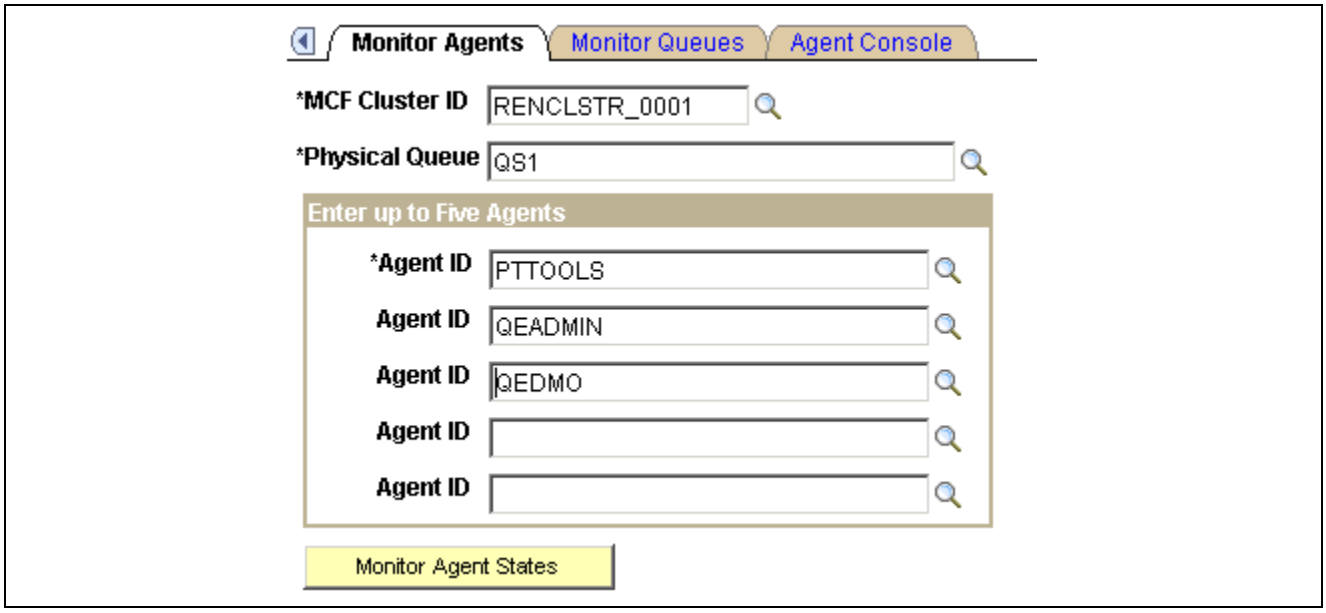
UQ State Displays the state of the queue server.

REN State Displays the state of the REN server.

Copyright © 1988-2007, Oracle. All rights reserved.

Using the Monitor Agents Page and Sample Monitor - Agent States Page

Access the Monitor Agents page.

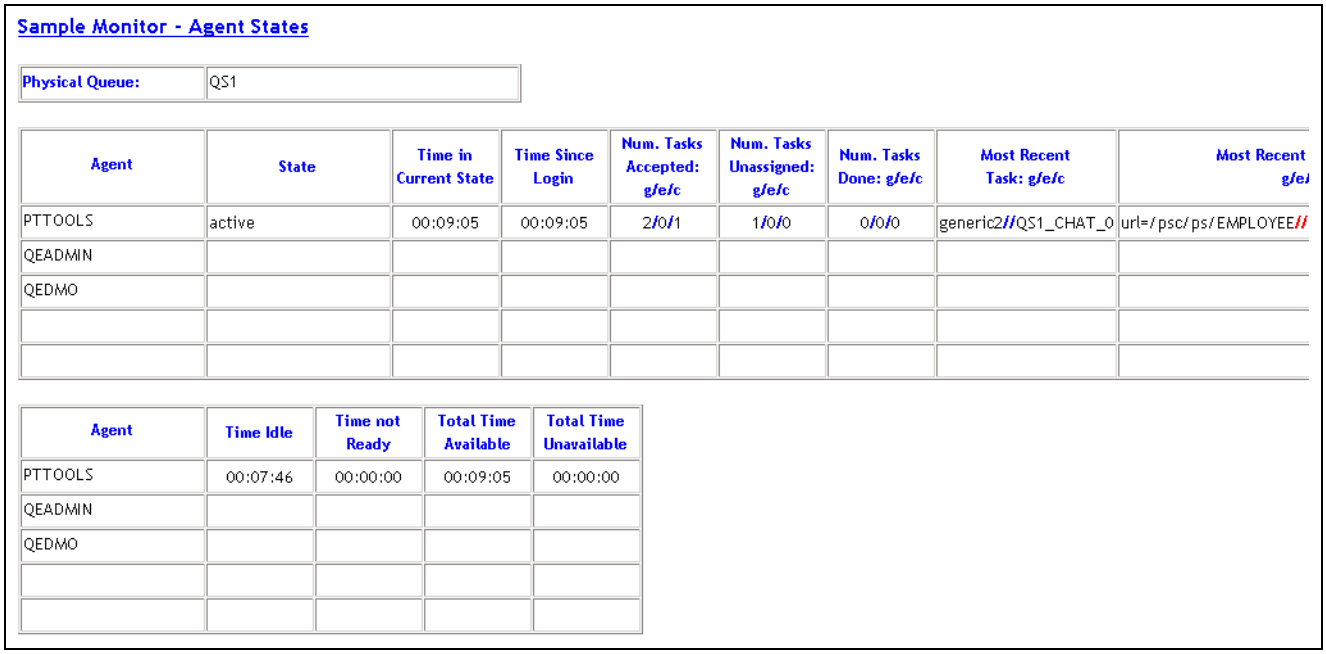


Monitor Agents page

Select the MCF cluster, physical queue on that cluster, and up to five agents on that queue to be monitored.

After you click Monitor Agent States, a new browser window appears.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.



Sample Monitor - Agent States page

Copyright © 1988-2007, Oracle. All rights reserved.

Physical Queue

The ID of the physical queue that is being monitored appears in the first table.

Agent Statistics Table 1

The first agent statistics table displays information that is published by the JSMCAPI user.onStat1 event.

Time in Current State	Displays the duration of the agent's current state. The value returned is `timeInCurrentState`.
Time Since Login	Denotes the duration since the agent logged in. The value returned is `timeLogin`.
Num. Tasks Accepted: g/e/c (number of tasks accepted)	Displays the number of tasks in each task type that the agent has accepted since login. The value returned is `numTaskAccepted`.
Num. Tasks Unassigned: g/e/c (number of tasks unassigned)	Displays the number of tasks in each task type that have been unassigned from the agent. Returned by `numTasksUnassigned`.
Num. Tasks Done: g/e/c (number of tasks done)	Displays the number of tasks in each task type that the agent has completed. Returned by `numTasksDone`.
Most Recent Task: g/e/c	Identifies the most recent task that was accepted since login for each task type. Returned by `mostRecentTaskId`.
Most Recent Task Data: g/e/c	Displays data on the most recent task that was accepted since login for each task type. Returned by `mostRecentTaskInfo`

Agent Statistics Table 2

The second agent statistics table displays information that is published by the JSMCAPI user.onStat2 event.

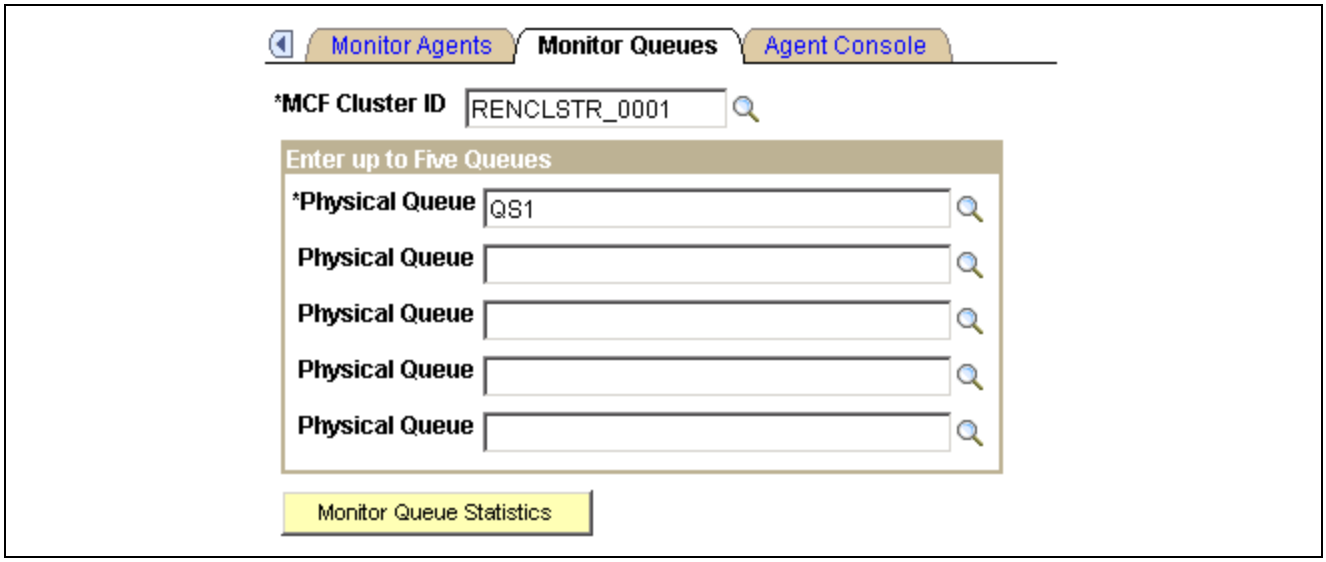
Time Idle	Displays the duration of time during which the agent has been idle. An agent is considered to be idle from the time the agent logs on until the agent accepts a task, and during the time between completing the last accepted task and accepting another task. The value returned is `timeIdle`.
Time not Ready	Displays the duration of time during which the queue could not assign tasks to the agent because the agent's maximum workload has been reached. The value returned is `timeNotReady`.
Total Time Available	Displays the total amount of time that the agent has been in an available status. The value returned is `totalTimeAvailable`.
Total Time Unavailable	Displays the total amount of time that the agent has been in an unavailable status.

Copyright © 1988-2007, Oracle. All rights reserved.

The value returned is `totalTimeUnavailable`.

Using the Monitor Queues Page and Sample Monitor - Queue Statistics Page

Access the Monitor Queues Page.



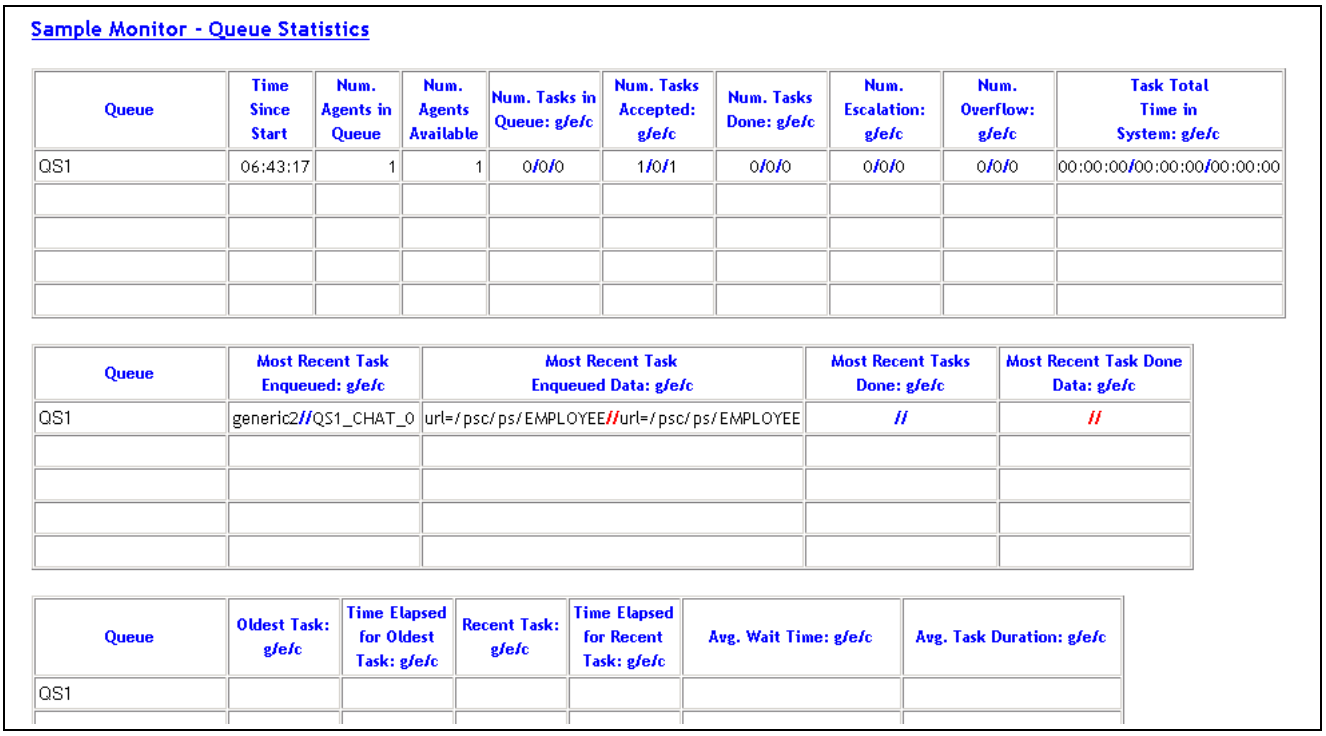
Monitor Queues page

Select the MCF cluster and up to five physical queues on that cluster to be monitored.

After you click Monitor Queue Statistics, a new browser window appears.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.

Copyright © 1988-2007, Oracle. All rights reserved.



Sample Monitor - Queue Statistics

Queue	Time Since Start	Num. Agents in Queue	Num. Agents Available	Num. Tasks in Queue: g/e/c	Num. Tasks Accepted: g/e/c	Num. Tasks Done: g/e/c	Num. Escalation: g/e/c	Num. Overflow: g/e/c	Task Total Time in System: g/e/c
QS1	06:43:17	1	1	0/0/0	1/0/1	0/0/0	0/0/0	0/0/0	00:00:00/00:00:00/00:00:00

Queue	Most Recent Task Enqueued: g/e/c	Most Recent Task Enqueued Data: g/e/c	Most Recent Tasks Done: g/e/c	Most Recent Task Done Data: g/e/c
QS1	generic2//QS1_CHAT_0	url=/psc/ps/EMPLOYEE//url=/psc/ps/EMPLOYEE	//	//

Queue	Oldest Task: g/e/c	Time Elapsed for Oldest Task: g/e/c	Recent Task: g/e/c	Time Elapsed for Recent Task: g/e/c	Avg. Wait Time: g/e/c	Avg. Task Duration: g/e/c
QS1						

Sample Monitor - Queue Statistics page

Queue Statistics 1

The first queue statistics table displays data that does not change frequently. The table displays information that is published by the JSMCAPI group.onStat1 event.

Queue
Displays an identifier for this queue.

The value returned is `Queue_Id`.

Time Since Start
The duration since this queue was started (application server domain start).

The value returned is `timeSinceStart`.

Num. Agents in Queue (number of agents in queue)
The number of agents that are currently logged in to this queue.

The value returned is `numAgentsLoggedIn`.

Num. Agents Available (number of agents available)
The number of agents that are currently logged in to this queue with an available status.

The value returned is `numAgentsAvail`.

Num. Tasks in Queue: g/e/c (number of tasks in queue)
The number of tasks that are currently on this queue, by task type.

The value returned is `numTasksQueued`.

Num. Tasks Accepted: g/e/c (number of tasks accepted)
The number of accepted tasks that are currently on this queue, by task type.

The value returned is `numTaskAccepted`.

Num. Tasks Done: g/e/c (number of tasks done)
The number of tasks on this queue that have been completed, by task type.

The value returned is `numTaskDone`.

Copyright © 1988-2007, Oracle. All rights reserved.

Num. Escalation: g/e/c (number of escalation)
The number of escalated tasks on this queue, by task type.
The value returned is `numEscalation`.

Num. Overflow: g/e/c (number of overflow)
The number of overflow tasks on this queue, by task type.
The value returned is `numOverflow`.

Task Total Time in System: g/e/c
Displays the total duration of the most recently completed task in the system. The total time is the difference between the time that the task was enqueued and the time at which it was marked done.
The value returned is `taskTotalTimeInSystem`.

Queue Statistics 2

The second queue statistics table displays data that changes frequently. The table displays information that is published by the JSMCAPI group.onStat1 event.

Most Recent Task Enqueued: g/e/c
Identifies the task that was most recently enqueued on this queue, by task type.
The value returned is `mostRecentTaskEnqueued`.

Most Recent Task Enqueued Data: g/e/c
Displays data for the most recently enqueued task on this queue, by task type.
The value returned is `mostRecentTasksEnqueuedData`.

Most Recent Tasks Done: g/e/c
Identifies the task that was most recently completed, by task type.
The value returned is `mostRecentTaskDone`.

Most Recent Task Done Data: g/e/c
Displays data for the most recently completed tasks, by task type.
The value returned is `mostRecentTaskDoneData`.

Queue Statistics 3

The table displays information that is published by the JSMCAPI group.onStat2 event.

Oldest Task: g/e/c
Identifies the oldest enqueued and accepted task on this queue, by task type.
The value returned is `oldestTask`.

Time Elapsed for Oldest Task: g/e/c
Displays the duration that the oldest task on this queue has been enqueued, by task type.
The value returned is `timeElapsedOldestTask`.

Recent Task: g/e/c
Identifies the most recent task that was enqueued and accepted on this queue, by task type.
The value returned is `recentTask`.

Time Elapsed for Recent Task: g/e/c
Displays the duration that the most recent task on this queue has been enqueued, by task type.
The value returned is `timeElapsedRecentTask`.

Avg. Wait Time: g/e/c (average wait time)
Displays the average time between a task being enqueued and being accepted, by task type.
The value returned is `averageWaitTime`.

Copyright © 1988-2007, Oracle. All rights reserved.

Avg. Task Duration: g/e/c (average task duration)

Displays the average duration before a task is completed.

The value returned is `averageTaskDuration`.

The average task duration is calculated by referencing a list of completed tasks. The size of that list is determined by the donelistsize parameter that is set on the Cluster Tuning page.

See Chapter 7, "Configuring PeopleSoft MCF Queues and Tasks," Tuning Cluster Parameters, page 94.

Using PeopleCode Built-in Functions with PeopleSoft MultiChannel Framework

PeopleSoft MultiChannel Framework uses several PeopleCode built-in functions. Application developers use these functions to communicate task requests and parameters to the queue server. For example, use InitChat code behind a button on an application page to initiate a chat session with an agent.

The built-in functions are:

- DeQueue

  Use DeQueue to notify the queue server that an enqueued task has been completed and to remove the task from the queue.

- EnQueue

  Use EnQueue to add a task to an active physical queue belonging to the specified logical queue.

- Forward

  Use Forward to transfer a task from one agent to another agent or another logical queue.

- InitChat

  Use InitChat to place a customer chat request on a queue.

- NotifyQ

  Use NotifyQ to notify the queue server of a task enqueued by EnQueue.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions".

Using Universal Queue Classes

The universal queue classes provide the means by which applications can inspect objects that are processed by the queue server, such as tasks, agents, and queues. The classes also enable tasks to reenter the queue with their original task ID, as well as keep task times based on their original entry into the system.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Universal Queue Classes".

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 12

Configuring PeopleSoft MCF for Third-Party Routing Systems

In previous releases, PeopleSoft CTI required a third-party middleware in the form of the Genesys CTI Framework, a Cisco ICM system, or CTI middleware that implements PSMCAPI.

Beginning with PeopleTools 8.48, PeopleSoft MultiChannel Framework (MCF) uses third-party multichannel routing systems to offer a wide range of channels of communication to empower PeopleSoft applications, like CRM. In the third-party system, the queue is the logical storage unit for representation of a work item and there is no difference between logical and physical queue. The third party develops the universal routing system to route email, voice, chat, and generic tasks.

PeopleSoft customers may choose either the pre 8.48 features of queue server and CTI, or the new third-party routing system to use email, chat, voice, and generic channels.

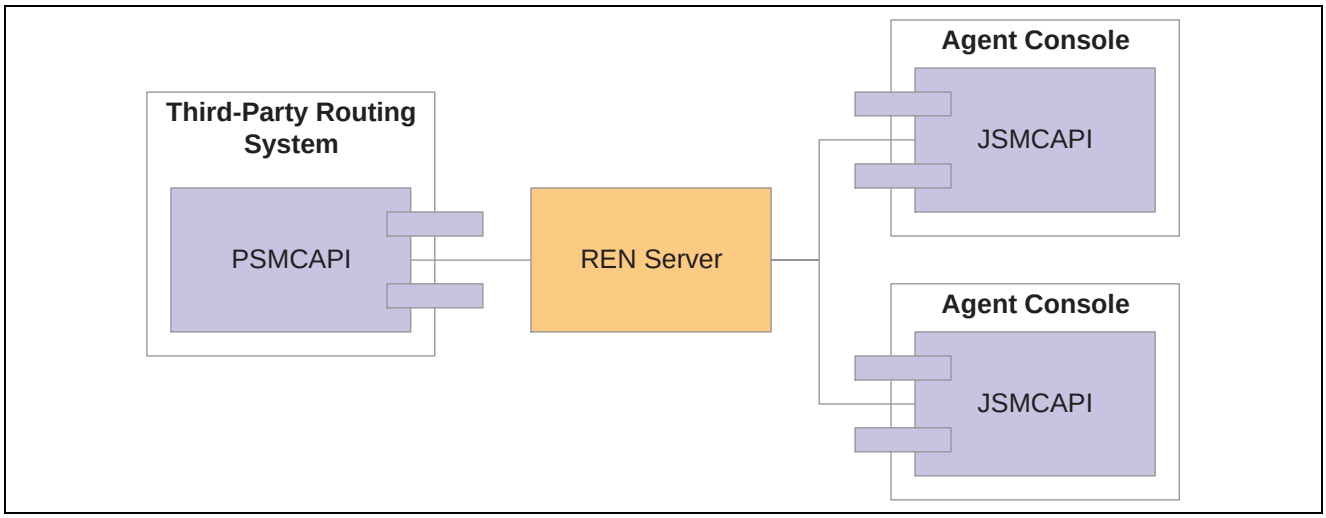
This chapter provides an overview of third-party routing systems and discusses how to configure PeopleSoft MCF for third-party routing systems.

Understanding Third-Party Routing Systems

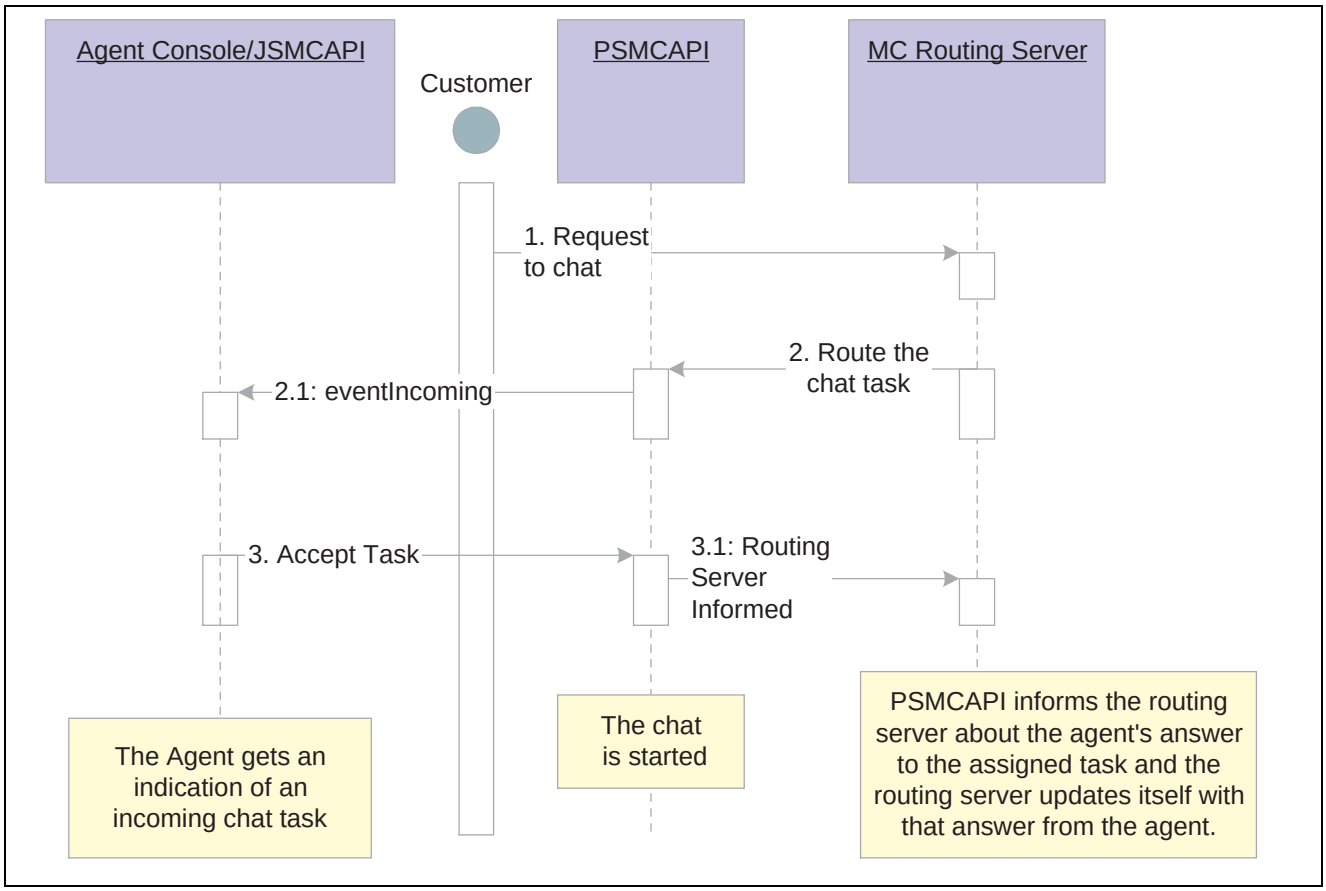
PSMCAPI provides an interface to the third-party routing system that enables communication between various PeopleSoft components and the third party. An event, like an incoming task from the third-party routing system is received by PSMCAPI and pushed to JSMCAPI (MCF console) using the REN server. Similarly, a request from MCF console or JSMCAPI is sent to PSMCAPI using the REN server and is eventually passed on to the third-party routing system. The REN server routes requests and events internally to the PeopleSoft system including an agent desktop or browser.

The following diagram illustrates how PSMCAPI acts as an interface to the third-party routing system and communicates with JSMCAPI through REN server.

Copyright © 1988-2007, Oracle. All rights reserved.



PSMCAPI interface with PeopleSoft MultiChannel Framework

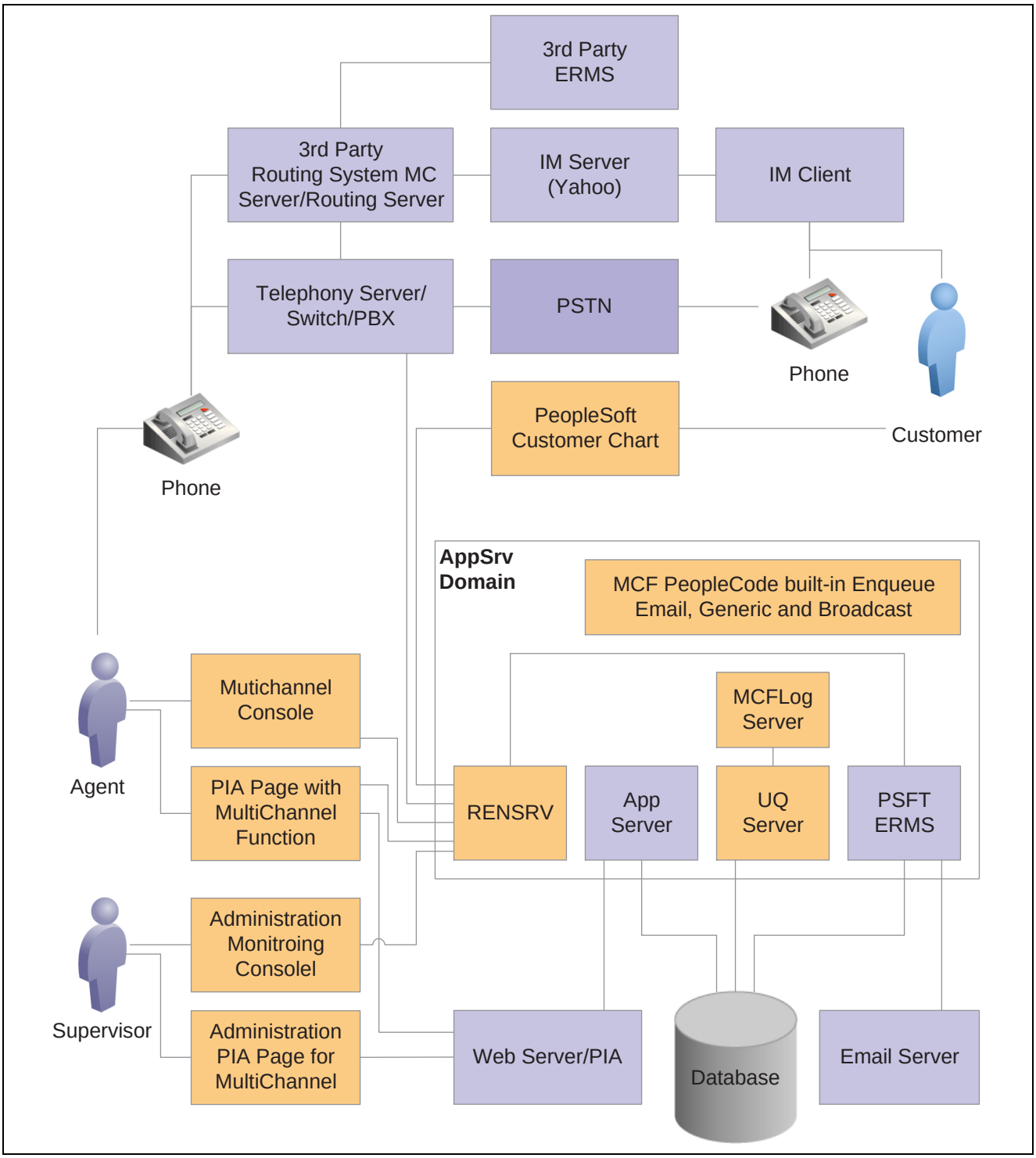


Interaction of PSMCAPI with third party and JSMCAPI

PSMCAPI and JSMCAPI supports multiple channels of communication, queue, and agent statistics information that enables the leverage of third-party routing systems.

The following diagram illustrates how third-party routing system interacts with PeopleSoft MCF.

Copyright © 1988-2007, Oracle. All rights reserved.



Third-party routing system with PeopleSoft MultiChannel Framework

The third-party routing system connects to all kinds of media servers, and routes the task to the agent based on routing rules. The third-party routing system connects to the REN server through HTTP or HTTPS using PSMCAPI. All requests sent by the agents to third-party system such as login, logout, accept incoming call, and forward email use the above connection. All media requests sent by PeopleCode built-in functions to the REN server such as InitChat or Enqueue also use the above connection. All events from the routing system such as assignment, broadcast, and statistics, go to the agent or the supervisor through this connection.

Copyright © 1988-2007, Oracle. All rights reserved.

In the third-party routing system, the queue server acts as an overflow or escalation adapter that writes the overflow or escalated tasks from the third-party routing system to the PeopleSoft Database. When using a third-party routing system for escalated and overflowed task, the queue server is used for task insertions to the database. The third-party routing system processes and maintains task properties and timers. On a task time-out, the routing system sends an event to the queue server to insert the task in the database. The routing system creates the events to remove the task from the agents and recalculates their workload.

Defining Third-Party Routing System Requirements

In general, the third-party router:

- Allows business users to define routing strategies.
- Defines escalation and timeout routing paths.
- Allows agents to receive any type of work from multiple channels and queues simultaneously or in parallel.
- Allows pushed and pulled events simultaneously or in parallel.
- Defines intra- and extra-channel escalations.

Defining Third-Party Routing Rules

The third-party routing rules are based on:

- Task type, for example email, chat, voice, scheduled callback, fax, and so on.
- Time against SLA (Service Level Agreement) in form of overflow and escalation timers.
- Priority of the event expressed as a positive single-digit number.
- Skill within organization or task type.
- Primary language required to resolve any issue.
- Cost of each event type assigned by the administrator to avoid overloading.

The agents or the supervisors must have the following capabilities:

- Change the skills, language, priority, or cost of an item to affect its subsequent routing.
- Assign an item to a specific queue or agent.
- Agents may accept, reject, forward, mark as resolved, or mark as awaiting additional information.

Configuring PeopleSoft MCF for a Third Party

This section outlines the basic steps to configure PeopleSoft MCF for a third party. The following sections describe each step in details.

To configure PeopleSoft MCF for a third party:

- Define the third-party flag.

  See Chapter 12, “Configuring PeopleSoft MCF for Third-Party Routing Systems,” Defining the Third-Party Flag, page 181.

- Define PeopleSoft MCF clusters for a third party.

 Copyright © 1988-2007, Oracle. All rights reserved.

See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Defining the PeopleSoft MCF Cluster Page for a Third Party, page 182.

- Tune PeopleSoft MCF clusters for a third party.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Tuning PeopleSoft MCF Cluster Parameters for a Third Party, page 184.

- Notify PeopleSoft MCF of changed parameters for a third party.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying PeopleSoft MCF Clusters of Changed Parameters for a Third Party, page 187.

- Define PeopleSoft MCF queues for a third party.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Defining PeopleSoft MCF Queues for a Third Party, page 188.

- Define PeopleSoft MCF agents for a third party.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Defining PeopleSoft MCF Agents for a Third Party, page 191.

- Configure PeopleSoft MCF tasks for a third party.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Configuring PeopleSoft MCF Task for a Third Party, page 200.

- Configure CTI.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Configuring CTI for a Third Party, page 201.

- View event logs.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Viewing Event Logs for a Third Party, page 201.

- View broadcast logs.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Viewing Broadcast Logs for a Third Party, page 203.

- Work with third-party sample pages.

  See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Working with Third-Party Sample Pages, page 203.

The following sections describe each of the steps in the preceding list in details.

Defining the Third-Party Flag

This section discusses how to define the third-party flag.

To define the third-party flag, use the Third-party Flag (MCF_TP_FLAG_CMP) component.

Copyright © 1988-2007, Oracle. All rights reserved.

Page Used to Define the Third-Party Flag

Page Name	Object Name	Navigation	Usage
Third-Party Flag	MCF_TP_FLAG_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Third-Party Flag	Select this flag to use the third-party routing system to configure MCF.

Defining the Third-Party Flag

Access the Third Party Flag page.

Third-Party Flag page

The default value of the third-party flag is *False.* If the flag is set to *False,* the queue server routes chat, generic, and email and uses PeopleSoft CTI to route voice. If the flag is set to *True*, all events, chat, generic, email, and voice are routed by the third-party routing system and you can configure MultiChannel Framework using third-party pages only. The queue server:

- Stops listening to all routing requests.
- Starts up new listeners for overflow and escalation.

Separate database tables are created for queue server and third-party routing server that combine all fields from MCF and CTI. When the third-party flag is set to *True*, all pages using queue server are disabled and grayed out. When the third-party flag is set to *False*, all pages using third-party routing system are disabled and grayed out.

Note. Whenever you change the value of this flag, you must reboot the queue server and MCF Log server.

Defining the PeopleSoft MCF Cluster Page for a Third Party

This section discusses how to define the PeopleSoft MCF Cluster page for the third party. To view the Cluster page, use the Cluster (MCF_TP_RENCFG_CMP) component.

Page Used to Define the PeopleSoft MCF Cluster Page for a Third Party

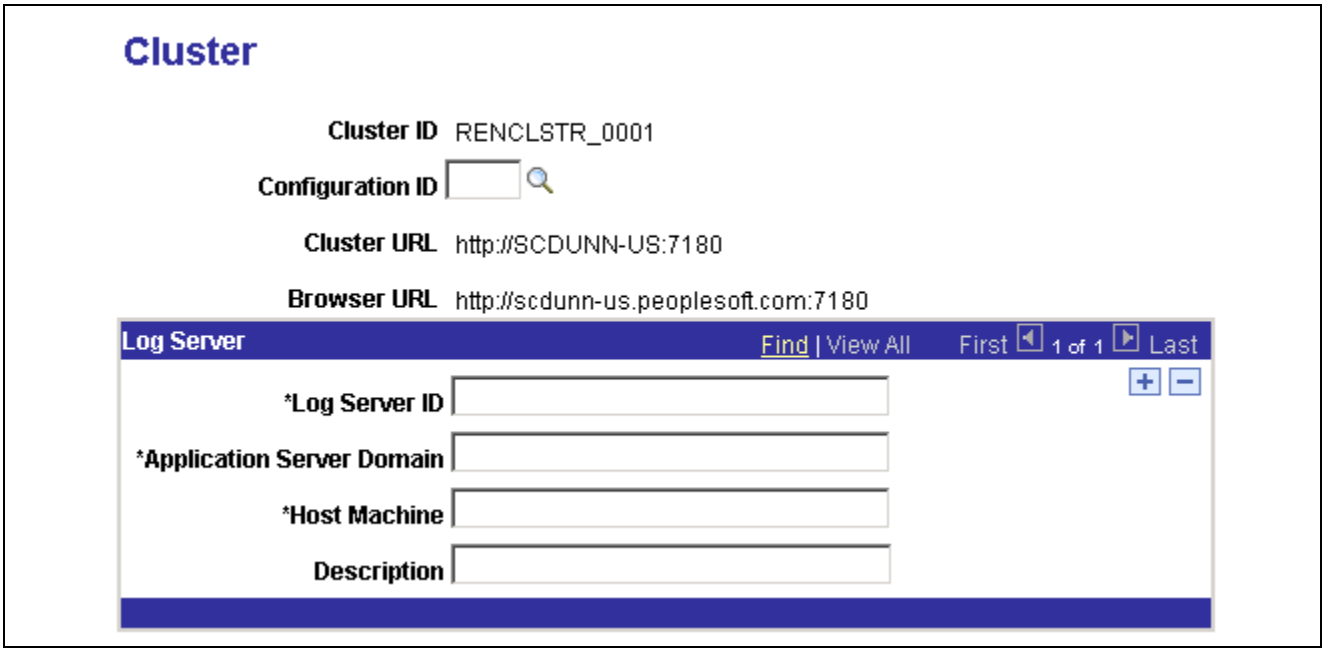
Page Name	Object Name	Navigation	Usage
Cluster	MCF_TP_UQCLUSTER	PeopleTools, MultiChannel Framework, Third-Party Configuration, Cluster	View information about third-party clusters.

Copyright © 1988-2007, Oracle. All rights reserved.

Defining the PeopleSoft MCF Cluster Page for a Third Party

The Cluster page displays the associated PeopleSoft MCF cluster URLs and log server details for the selected cluster.

Access the Cluster page.



Cluster page

Cluster ID Displays the cluster ID of the cluster containing the queue server and log server.

Configuration ID Displays the configuration ID.

Note. This is required only for CTI configuration. To select the configuration ID from lookup, define the configuration ID when you are configuring CTI.

See Chapter 3, "Configuring PeopleSoft CTI," Creating CTI Configurations, page 22.

Cluster URL Displays the URL for the REN server that serves this cluster.

Note. The Cluster URL is same as defined on the REN Server Definitions page.

See Chapter 5, "Configuring REN Servers," Defining REN Servers, page 66.

Browser URL Displays the browser URL.

Note. The browser URL may be different from the cluster URL, which should not have to go through any firewall, reverse proxy server, or other outward-facing security barrier.

The browser URL is same as defined in the REN Server Definitions page.

See Chapter 5, "Configuring REN Servers," Defining REN Servers, page 66.

Copyright © 1988-2007, Oracle. All rights reserved.

Log Server

A cluster consists of a primary log server and any number of backup servers. Each cluster requires a minimum of one log server and one queue server. The primary log server is the first log server started and the remaining log servers are backups. If the primary log server fails, the system determines the subsequent primary log server among the backups.

Note. You can save a cluster without creating a log server or a queue server. But, for MCF to run, you need at least one log server and one queue server.

You can add a log server to a cluster by adding a new row. Before removing a log server, ensure that it is not the master, then shut down its domain. Then click Delete (the minus sign).

Log Server ID Enter a unique identifier for each log server to identify its entries in the log cluster table.

The queue server process paired with this log server uses this same ID to identify its entry in the database table.

Application Server Domain Enter the application server domain of which this log server is a member.

Host Machine Enter the name of the application server host machine.

Description Enter the description of the host machine.

Tuning PeopleSoft MCF Cluster Parameters for a Third Party

This section discusses how to tune PeopleSoft MCF cluster parameters for a third party. Tuning cluster parameters may give you better performance. To tune cluster parameters, use the Cluster Tuning (MCF_TP_SYS_NV_CMP) component.

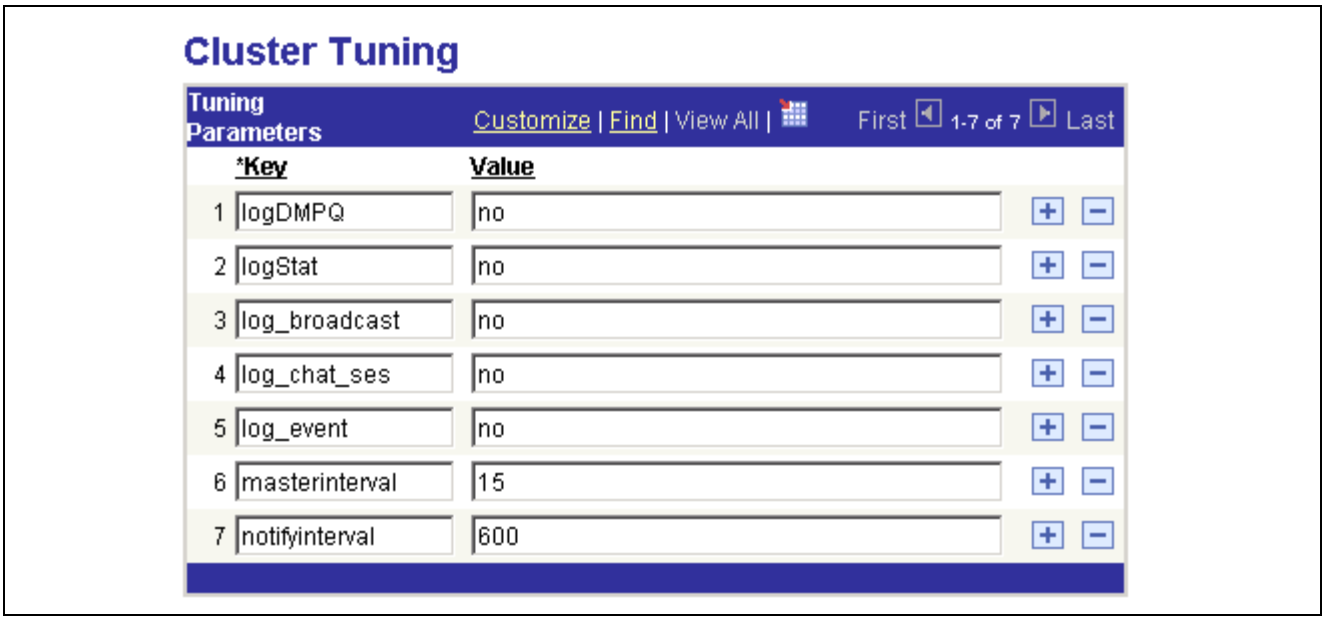
Page Used to Tune PeopleSoft MCF Cluster Parameters for a Third Party

Page Name	Object Name	Navigation	Usage
Cluster Tuning	MCFTP_SYSTEM_KV_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Cluster Tuning	Modify cluster parameters to tune performance.

Tuning PeopleSoft MCF Cluster Parameters for a Third Party

Access the Cluster Tuning page.

 Copyright © 1988-2007, Oracle. All rights reserved.



Cluster Tuning page

Use the Cluster Tuning page to set MCF cluster parameters to optimize performance or enable logging for a cluster.

If you make changes to a cluster parameter, you must use the third-party Notify Cluster page to propagate the changes.

See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188.

The following table lists the cluster tuning parameters you can modify and describes the default values and usage of each:

Key	Default value	Usage
logDMPQ	*No*	Select *Yes* if you want PSMCFLOG to log REN server event notifications resulting from bcastinterval broadcasts. Only the event is logged, not its contents. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the third-party Cluster Notify page to notify clusters of the changed parameter See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Event Logs, page 123.

Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
logStat	*No*	Select *Yes* to log the statistics returned by the queue server for the onStat1 user and group events to the database. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the third-party Cluster Notify page to notify clusters of the changed parameter See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188 and Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Using and Demonstrating JSMCAPI, page 155.
log_broadcast	*No*	Select *Yes* to turn on logging of the broadcast messages that are sent. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the third-party Cluster Notify page to notify clusters of the changed parameter See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Broadcast Logs, page 120.
log_chat_ses	*No*	Select *Yes* to turn on logging of the contents of chat sessions. This is a logging parameter. If you change the value of this parameter you must use the Refresh Logging Parameters button on the third-party Cluster Notify page to notify clusters of the changed parameter See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188 and Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Chat Logs, page 121.

 Copyright © 1988-2007, Oracle. All rights reserved.

Key	Default value	Usage
masterinterval	*15*	Interval, in seconds, after which a cluster master updates its timestamp in its cluster tables. Slave clusters check the timestamp to determine that the master cluster is still running. A lower value enables rapid discovery of a failed master server, but increases log server overhead. A higher value reduces log server overhead, but delays discovery of a failed master server. If a only one log server is configured for an MCF cluster, this value can be large. The masterinterval value also acts as a heartbeat interval for the master log server connection to user consoles. This is a timing parameter. If you change the value of this parameter you must use the Refresh Timing Parameters button on the third-party Cluster Notify page to notify clusters of the changed parameter. See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188.
notifyinterval	600	Interval, in seconds, after which the database is checked for any pending enqueued tasks. The third party will be notified if any enqueued task is pending in the database. This is a timing parameter. If you change the value of this parameter you must use the Refresh Timing Parameters button on the third-party Cluster Notify page to notify clusters of the changed parameter. See Chapter 12, "Configuring PeopleSoft MCF for Third-Party Routing Systems," Notifying Third-Party Clusters of Changed Parameters, page 188.

Notifying PeopleSoft MCF Clusters of Changed Parameters for a Third Party

This section describes how to notify PeopleSoft MCF clusters of changed parameters for a third party. To notify clusters of changed parameters, use the Cluster Notify (MCF_TP_NOTIFY_CMP) component.

Copyright © 1988-2007, Oracle. All rights reserved.

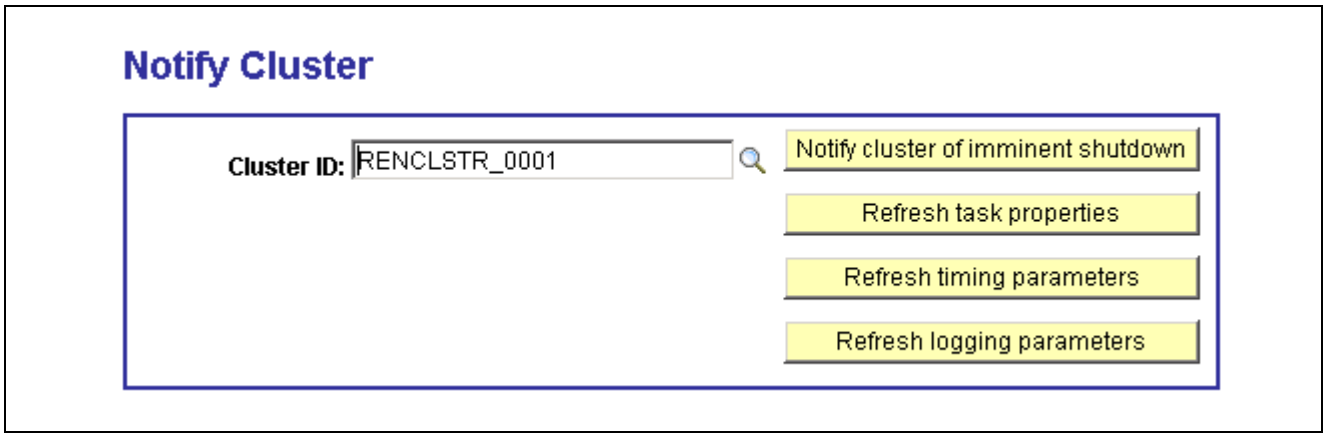
Page Used to Notify Third-Party Clusters of Changed Parameters

Page Name	Object Name	Navigation	Usage
Notify Cluster	MCFTP_CL_NOTIFY_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Cluster Notify	Notify a cluster of changes to parameters, configuration, or of shutdown.

Notifying Third-Party Clusters of Changed Parameters

Use the Notify Cluster page to notify a cluster of certain changes to its parameters, constituent queues, or that its application servers are being shut down.

Access the Notify Cluster page.



Cluster Notify page

Notify cluster of imminent shutdown Click to send a message to every agent logged on to the selected cluster that they have been logged off.

Send this notification if the cluster's application servers are being shut down.

Refresh task properties Click to load task properties that have been changed on the Tasks page for the selected cluster.

Refresh timing parameters Click to reload parameters, other than logging parameters, changed on the Cluster Tuning page for the selected cluster.

Refresh logging parameters Click to reload logging parameters changed on the Cluster Tuning page for the selected cluster.

Defining PeopleSoft MCF Queues for a Third Party

To define PeopleSoft MCF queues for the third-party routing system, use the third-party queue (MCF_TP_Q_CFG_CMP) component. This section discusses how to:

- Define queues.
- Define canned queue messages.

Copyright © 1988-2007, Oracle. All rights reserved.

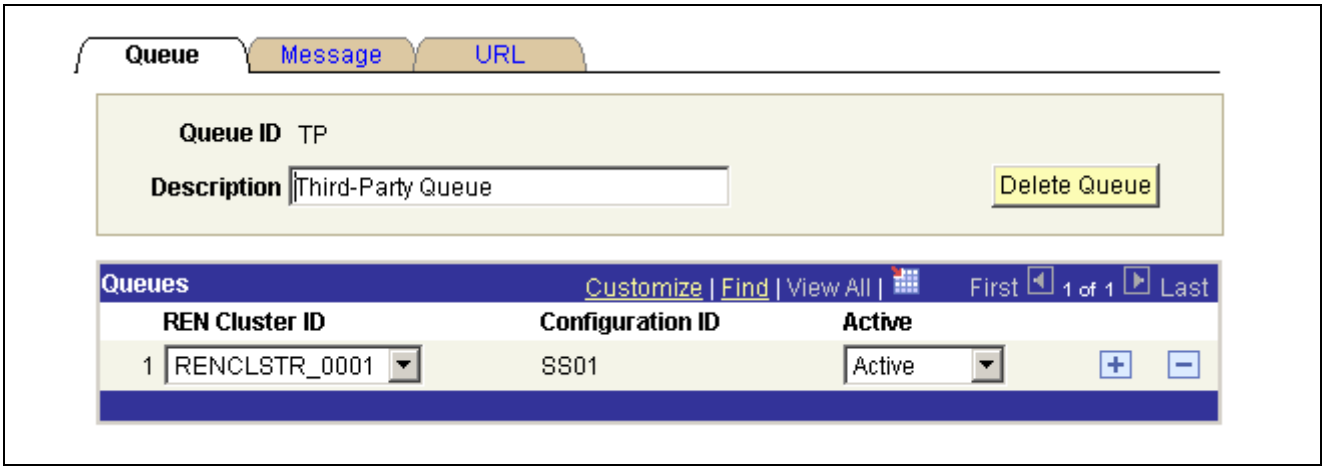
- Define canned queue URLs.

Pages Used to Define PeopleSoft MCF Queues for a Third Party

Page Name	Object Name	Navigation	Usage
Queue	MCF_TP_QUEUE_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Queue	Add a queue or view the details of an existing queue.
Message	MCFTPSYSMSG_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Queue, Message	Define messages that every agent can use.
URL	MCFTPQUEUEURL_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Queue, URLs	Define URLs that each agent can push to a client.

Defining PeopleSoft MCF Queues For Third Party

Access the Queue page.



Queue page

Queue ID Queue IDs must be alphanumeric, but may include underscore characters.

Configuration ID Displays the configuration ID.

Note. This is required only for CTI configuration. To select the configuration ID from lookup, define the configuration ID when you are configuring CTI.

See Chapter 3, "Configuring PeopleSoft CTI," Creating CTI Configurations, page 22.

Delete Queue Click to remove this queue.

Deleting a queue means that no work or agents can be assigned to the queue and the queue is removed from all agents' available queues.

Cluster ID The Cluster ID field identifies the cluster that services this queue.

Copyright © 1988-2007, Oracle. All rights reserved.

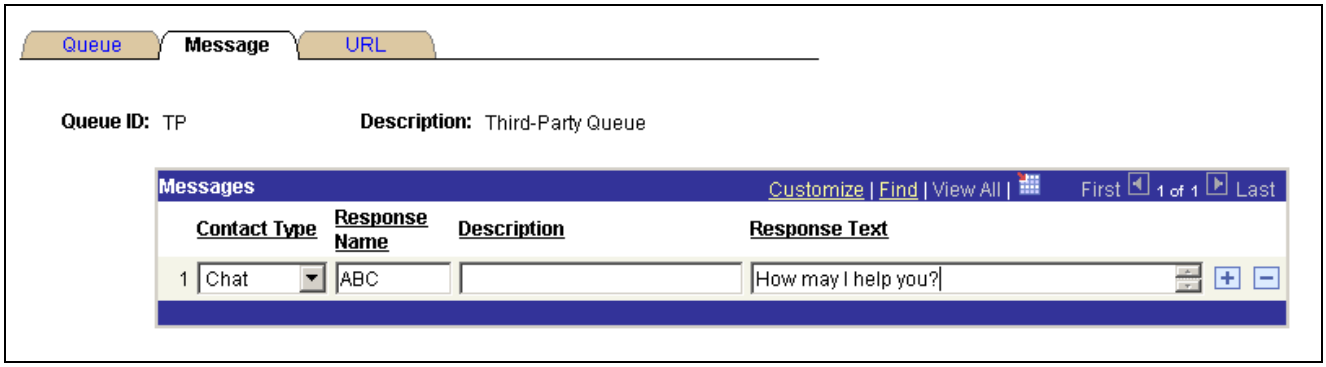
Active Select *Active* or *Inactive* from the drop-down list box.

The queue server does not send new tasks to an inactive queue. Agents and existing tasks remain on an inactive queue. To receive new queued tasks, the queue must be active.

Active and inactive status support "follow-the-sun" practices. For example, SALES1 could be supported in the London office, and SALES2 could be supported in the San Francisco office when the London office is closed by activating and inactivating the appropriate queues.

Defining Canned Queue Messages

Access the Message page.



Message page

Contact Type Select one of the following contact types:

- *Chat*
- *Email*

Response Name The response name appears in the agent's Template Messages drop-down list box for all agents who belong to this queue.

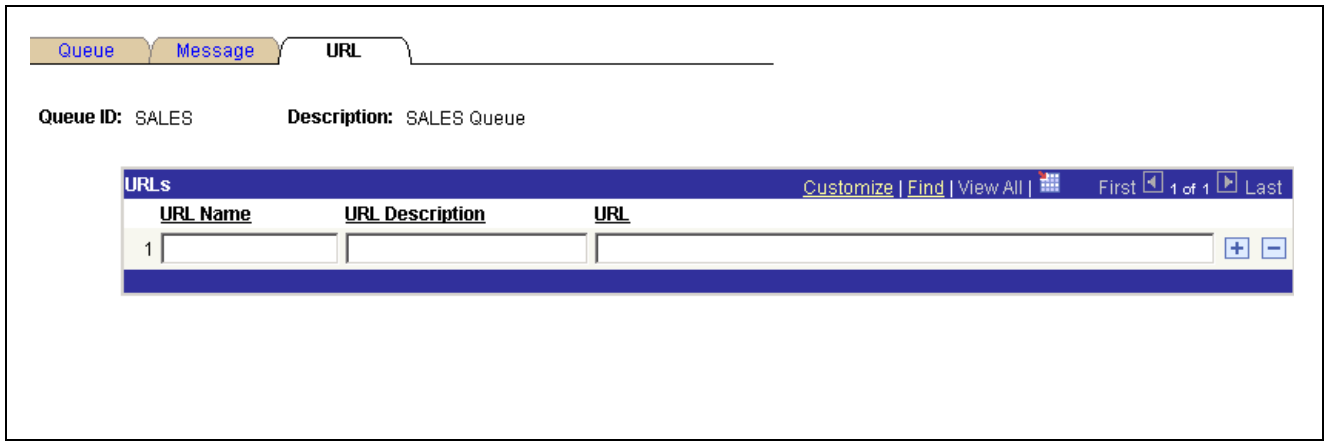
All messages are downloaded when the agent launches the agent chat console by accepting a customer chat. Template messages are not available in collaborative chat.

Response Text The specified message appears in the client chat window when selected by the agent.

Defining Canned Queue URL

Access the URL page.

Copyright © 1988-2007, Oracle. All rights reserved.



URL page

Canned queue URLs enable the agent to push a predefined URL to the customer, which appears in a pop-up window for the customer. The agent can edit these URLs in the chat window after selecting them.

URL Name Enter the name of the URL.

The URL name appears in the agent's URL drop-down list box for all agents that belong to this queue.

URL Description Enter a description of the URL.

This description appears only on this page to further describe this URL or, for example, its reason for inclusion.

URL Enter the URL. The URL must include the protocol (http://) and any required parameters.

All canned queue URLs defined for the queue are downloaded when the agent launches the agent chat console by accepting a customer chat. These URLs are not available in collaborative chat.

If you send a canned queue URL that is a PeopleSoft Pure Internet Architecture URL, be sure that the recipient has permissions to access that portal, node, or page.

Defining PeopleSoft MCF Agents for a Third Party

To define PeopleSoft MCF agents for a third party, use the (MCF_TP_AGENT_CMP) component. This section discusses how to:

- Create agents.
- Set up buddy lists.
- Customize windows.
- Specify messages.
- Specify agent-specific URLs.
- Specify media.
- Specify languages that an agent supports.

Copyright © 1988-2007, Oracle. All rights reserved.

- Customize an agent's presence.
- Specify miscellaneous parameters.

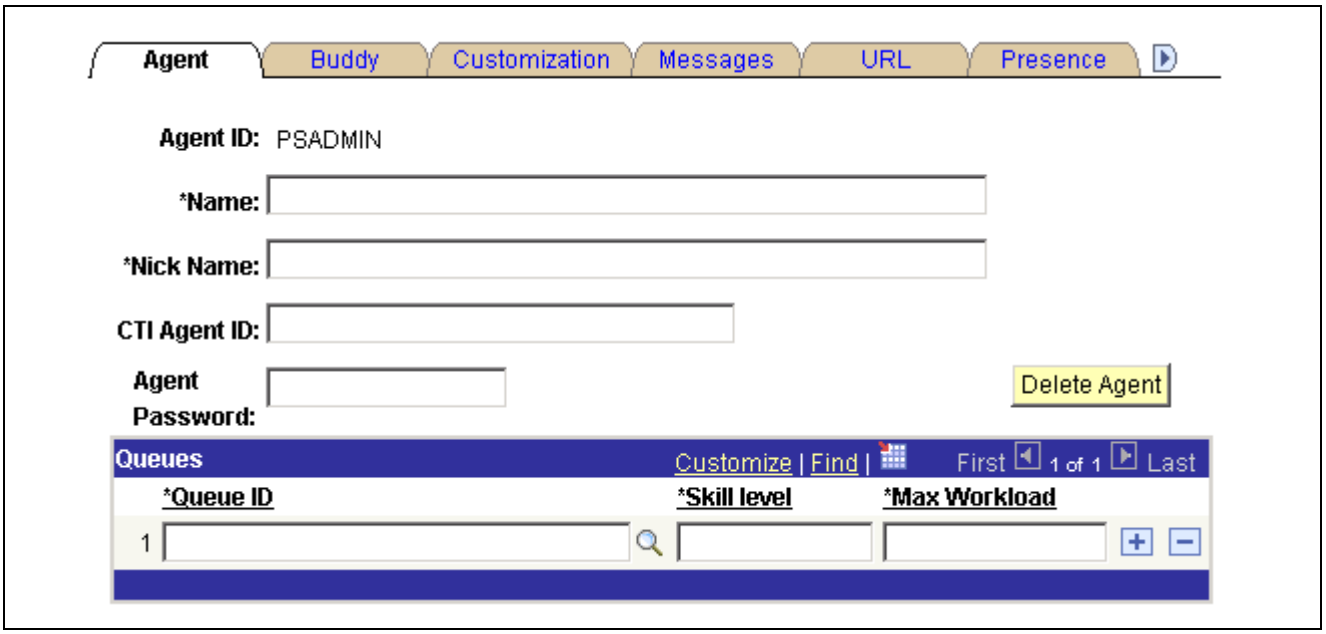
Pages Used to Define PeopleSoft MCF Agents for a Third Party

Page Name	Object Name	Navigation	Usage
Agent	MCFTP_AGENT_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent	Enter name, skill level, maximum workload, and queues for each agent.
Buddy	MCFTPAGENTBUDDY_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Buddy	Enter a list of other agents.
Customization	MCFTPAGENTCUST_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Customization	Configure the specified agent's window.
Messages	MCFTPAGENTMSG_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Messages	Enter agent-specific messages, in addition to available system messages.
URL	MCFTPAGENTURL_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, URL	Enter agent-specific URLs, in addition to canned queue URLs.
Presence	MCFTPAGENTPRES_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Presence	Enter an agent's presence options.
Media	MCFTPAGENTMEDIA_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Media	Enter an agent's capabilities.
Languages	MCFTP_AGENTLANG_PG	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Languages	Enter the languages that an agent is qualified to use. Enter at least one language.
Miscellaneous	MCFTP_AGENTMISC	PeopleTools, MultiChannel Framework, Third Party Configuration, Agent, Miscellaneous	Enter miscellaneous behaviors.

Creating PeopleSoft MCF Agents for a Third Party

Access the Agent page.

 Copyright © 1988-2007, Oracle. All rights reserved.



Agent page

Agent ID Specifies the agent ID.

Name Enter the full name of this agent in (lastname,firstname) format. The agent name appears in other agents' buddy lists.

Nick Name Enter a short name for this agent. The agent nickname identifies this agent in chat sessions and logs.

CTI Agent ID Enter a CTI agent ID for a CTI agent.

Agent Password Enter a password for the agent.

Queue ID Enter the ID of a queue to which this agent is assigned. Each agent may be assigned to more than one queue. An agent can log on to only one queue at a time from the MultiChannel Console.

Skill Select the skill level of this agent for the tasks assigned for this queue. This is a required field.

This option is used by third parties for task assignment. Third parties define skill levels based on business needs, requirements, policy and so on.

Maximum Workload Select the maximum load that this agent can be assigned before tasks are held or assigned to other agents. This is a required field.

This option is used by third parties for task assignment. Third parties define maximum workload based on business needs, requirements, policy and so on.

Note. Do not delete a queue from an agent's list unless that agent has no open accepted tasks on that queue.

Setting Up Buddy Lists

Access the Buddy page.

Copyright © 1988-2007, Oracle. All rights reserved.

Buddy page

The agent's buddy list facilitates collaborative chat and chat conferencing.

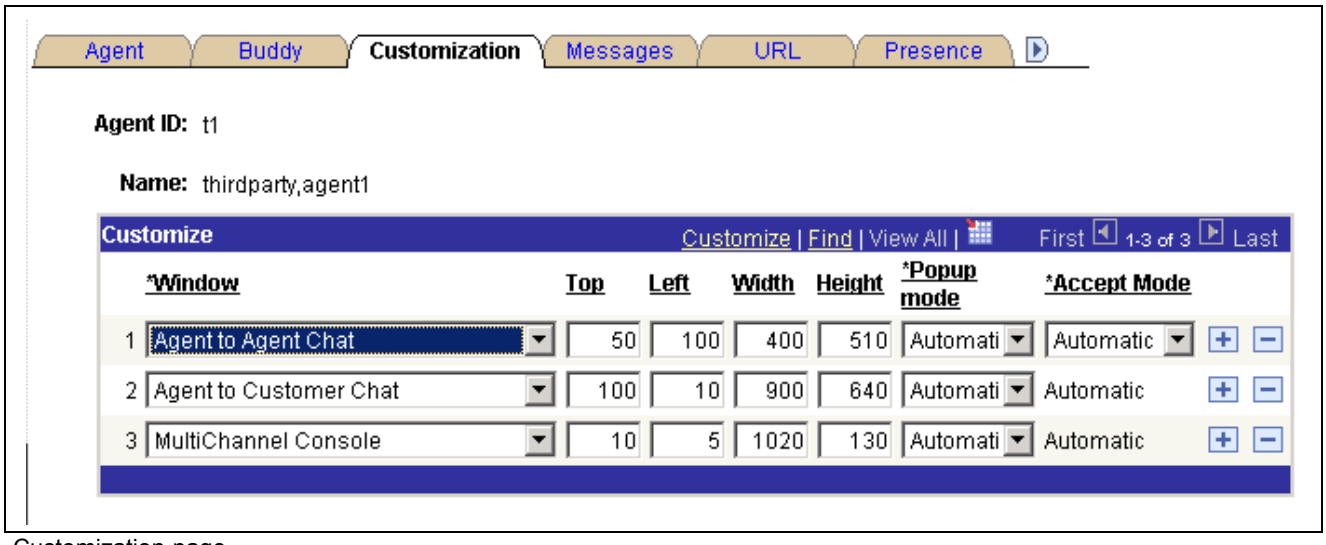
Agent Buddy Select another agent with whom this agent can have a chat session or can ask to conference onto another chat.

Name Displays the buddy agent's nickname. The name appears in the buddy list on the MultiChannel Console or agent chat window.

An agent's presence as shown in the buddy list on the MultiChannel Console or on the Invite Agent list on the chat console indicates the agent's availability for chat or conference.

Customizing Windows

Access the Customization page.



Customization page

Set the initial agent window placement and size by specifying parameters on this page. The windows may be resized and moved by an agent.

Window Select the window to which the specified configuration applies.

Select from:

Copyright © 1988-2007, Oracle. All rights reserved.

- *Agent to Agent Chat*
- *Agent to Customer Chat*
- *E-mail*
- *Generic Alert*
- *Grab URL*
- *MultiChannel Console*

Top and **Left** — Enter the distance in pixels from the top and left edge of the screen when the window first appears.

Width and **Height** — Enter the width and height, in pixels, of the window when it first appears.

Popup Mode — Select from:

Automatic: The window appears automatically. For customer-initiated chat or tasks initiated from the EnQueue built-in function, the task is automatically accepted as well. For agent-initiated chat, the agent can elect not to accept the task; in effect, the agent can preview the task. If this is the desired behavior, select *Manual* from the Accept Mode drop-down list box. If you want agent-to-agent tasks to function like customer-initiated tasks, select *Automatic* from the Accept Mode drop-down list box.

Manual: The window does not appear until the agent clicks the task on the agent MultiChannel Console. For customer-initiated chat or tasks initiated from the EnQueue() built-in function, clicking the task means that the task is automatically accepted as well. For agent-initiated chat, the behavior depends on the setting for the Accept Mode field.

Accept Mode — Select from:

Automatic: Agent-to-agent chats are automatically accepted without requiring the agent to click the icon.

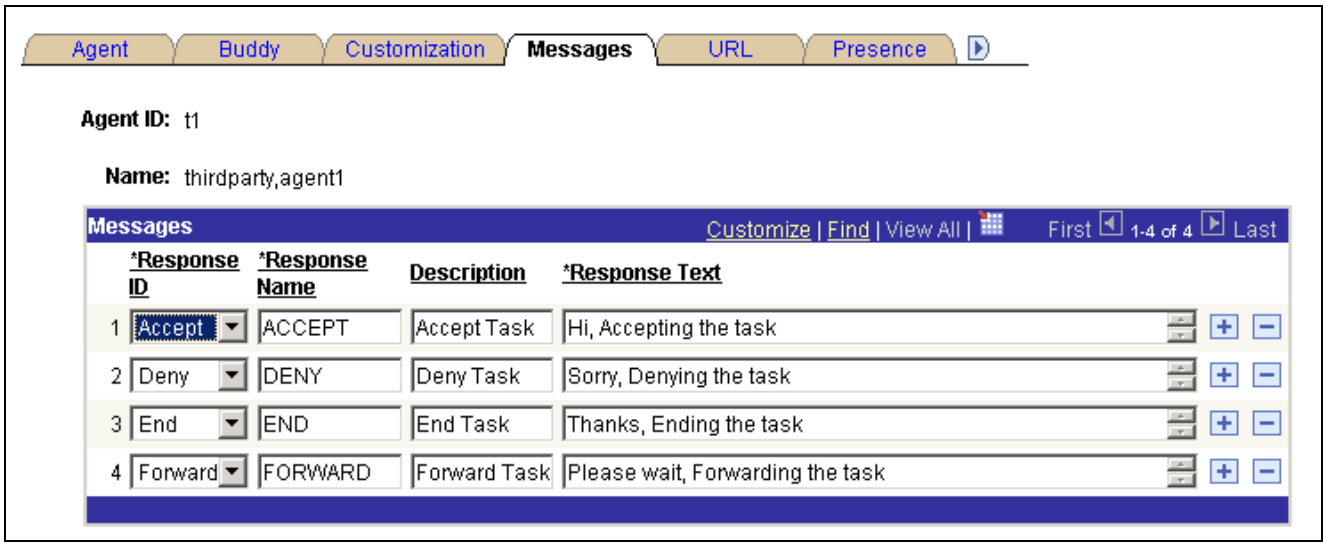
Manual: Agent-to-agent chats require the agent to click the icon.

Accept mode affects only collaborative chat.

Defining Messages

Access the Messages page.

Copyright © 1988-2007, Oracle. All rights reserved.



Messages page

An agent can create personalized responses in addition to the system responses defined for each queue.

Response ID

Responses, except those identified as Other, are linked to specific events. These responses, except for those identified Other, are always sent on these events from this agent. If an agent does not have a customized response for a specific event, it is read from a default response set in the Message Catalog. The response text set here overrides the default text set in the Message Catalog.

Select from:

- *Abandon:* A chat is abandoned when a chat initiator closes the chat window before the chat is accepted by an agent. This response appears when the agent accepts the abandoned chat.
- *Accept:* This response is automatically sent to the chat initiator when an agent accepts a chat in response to a chat request that does not include a question. If the chat request includes a question, the agent's Answer Question text is sent in response instead of the Accept response.
- *Answer Question:* This response is automatically sent to the chat initiator when an agent accepts a chat in response to a chat request that includes a question.
- *Deny:* Only applies to collaborative chat. If an agent elects not to accept a chat, this response is automatically sent to the chat initiator.
- *End:* If either party exits a chat after the chat is accepted, this response is displayed from the agent.
- *Forward:* If the agent forwards a chat session to another queue, this response is sent to the customer.
- *Other:* These responses are never automatically sent in a chat session. Their message names appear in the Template Messages drop-down list box on the agent chat page. These messages are appended to the template messages (chat responses) defined for the queue.

Response Name

This name appears in the agent's template response drop-down list box.

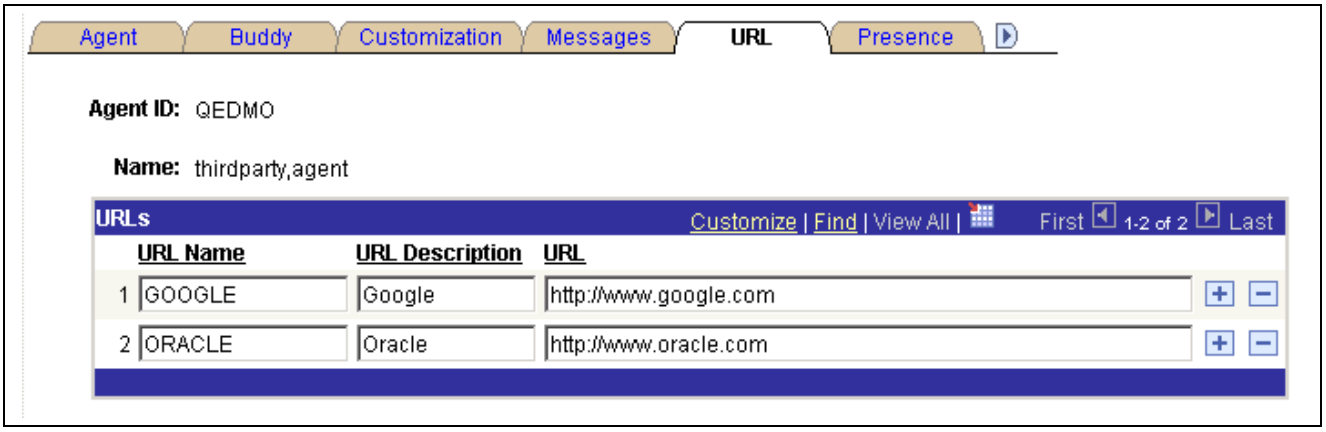
Response Text

Enter the response text to appear in the chat window.

Copyright © 1988-2007, Oracle. All rights reserved.

Specifying Agent-Specific URLs

This page defines URLs that this agent can send to a client browser. Access the URL page.



URL page

URL Name The URL name appears in the agent's URL drop-down list box.

URL Description This description appears only on this page, to further describe this URL or, for example, its reason for inclusion.

URL Enter the URL. The URL must include the protocol (http://) and any required parameters.

All URLs defined for the agent are downloaded when the agent launches the agent chat console by accepting a customer chat. These URLs are not available in collaborative chat.

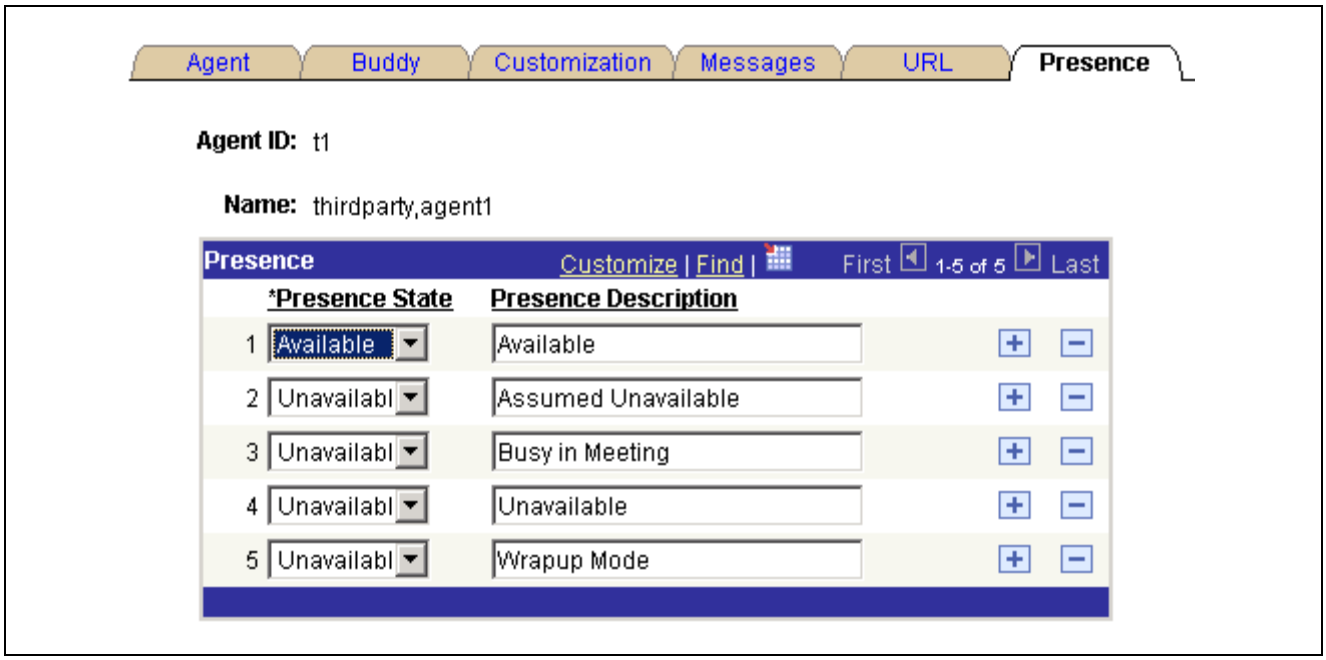
If you send a PeopleSoft Pure Internet Architecture URL, be sure that the recipient has permissions to access that portal, node, or page.

Defining Agent's Presence

Each agent can customize the presence description displayed when the agent is available or unavailable. The queue server only understands the presence state, available or unavailable, but you can enter more specific presence descriptions when displaying or logging an agent's presence. If you do not enter presence descriptions, default values are used.

Access the Presence page.

Copyright © 1988-2007, Oracle. All rights reserved.



Presence page

Presence State Select *Available* or *Unavailable.*

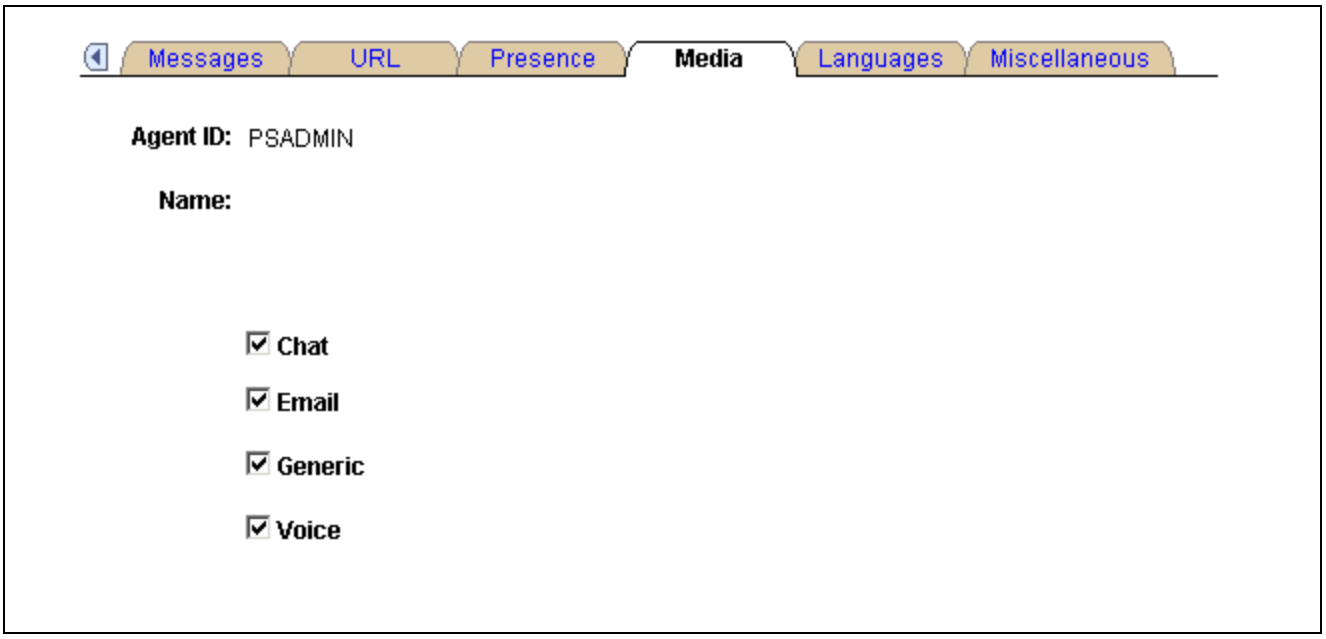
Presence Description Enter a description for each agent state. The description appears in logs of agent activity and when agent presence is displayed.

Note. There is only one presence description, *Available* for an available state. For an unavailable state, you can enter any presence description, such as tea break, coffee break, lunch, and so on.

Specifying the Media

The media page defines the capability of each agent to perform a task. Access the Media page.

Copyright © 1988-2007, Oracle. All rights reserved.

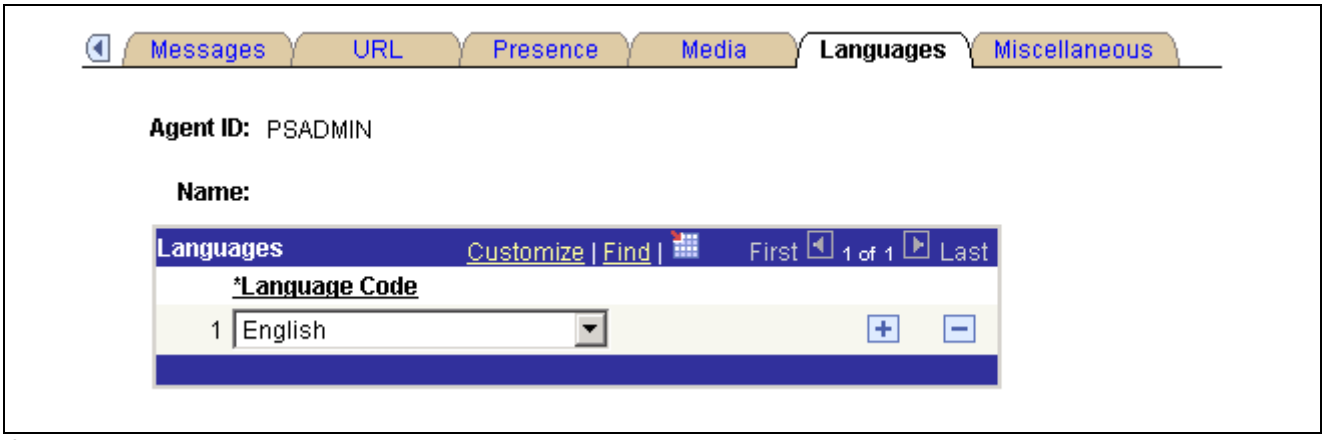


Media page

Chat	Select this if the agent has a capability to chat.
Email	Select this if the agent has a capability to email.
Generic	Select this if the agent has a capability to perform a generic task.
Voice	Select this if the agent has a capability to perform a voice task.

Specifying Languages That an Agent Supports

Access the Language page.



Languages page

Enter the languages that each agent is qualified to support.

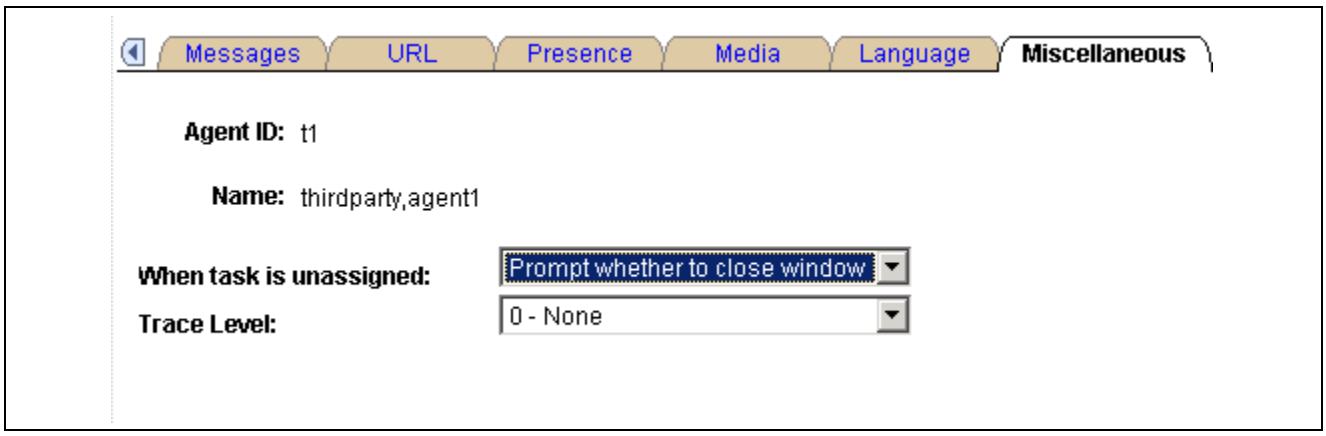
The agent is assigned only tasks that have been enqueued with a language code in the agent's language list. For the EnQueue built-in function, the language code is specified as a parameter. For the InitChat built-in function, the language code is determined by the user profile of the initiator.

If you do not enter a language code for a new agent, the default value is English.

Copyright © 1988-2007, Oracle. All rights reserved.

Specifying Miscellaneous Parameters

Access the Miscellaneous page.



Miscellaneous page

When task is unassigned Select from the following options the action that occurs when a task assigned to an agent is unassigned:

- *Prompt whether to close window (default).*
- *Close the task window.*
- *Do not close the task window.*

Trace Level Select from the following log trace levels:

- *0 - None*
- *1 - Information*
- *2 - Debug*

Configuring PeopleSoft MCF Task for a Third Party

This section discusses how to configure tasks. To configure tasks, use the MCF Task (MCF_TP_TASKCFG_CMP) component.

Page Used to Configure PeopleSoft MCF Task for a Third Party

Page Name	Object Name	Navigation	Usage
Task Configuration	MCF_TASKCFG_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Task	Configure tasks such as chat, email, voice, and generic alerts.

Configuring PeopleSoft MCF Task for a Third Party

See Chapter 7, “Configuring PeopleSoft MCF Queues and Tasks,” Configuring Tasks, page 89.

 Copyright © 1988-2007, Oracle. All rights reserved.

Configuring CTI for a Third Party

This section discusses how to configure CTI using third-party routing system. To configure CTI, use the MCF_TP_CTI_CONFIG component.

See Chapter 3, "Configuring PeopleSoft CTI," Configuring PeopleSoft CTI Using CTI Server or an Integrated Third-Party Routing System, page 19.

Viewing Event Logs for a Third Party

This section discusses how to view event logs. To view event logs, use the (MCF_TP_EVNTLOG_CMP) component.

Domain	Time event logged to DBMS	Event type	Task Type	Queue ID	Agent ID	CTI Agent ID	ANI	DNIS	This DN	Other DN	Call ID
MCF1	12/07/2005 5:02:30PM	Restrt Ack	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/12/2005 12:09:40PM	User RQ	Voice	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/12/2005 12:09:40PM	User Event	Voice	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/15/2005 2:01:29PM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/15/2005 2:01:29PM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/15/2005 2:01:34PM	Restrt Ack	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/19/2005 1:33:14PM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/19/2005 1:33:14PM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/19/2005 1:34:02PM	Restrt Ack	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/19/2005 2:23:06PM	Accept	Generic	MARKETING	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/19/2005 2:23:06PM	DB Cntct	Generic	MARKETING	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/21/2005 11:05:00AM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/21/2005 11:05:00AM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)
MCF1	12/28/2005 11:28:28AM	Q Shutdown	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)	(blank)

Event Log page

Page Used to View Event Logs for a Third Party

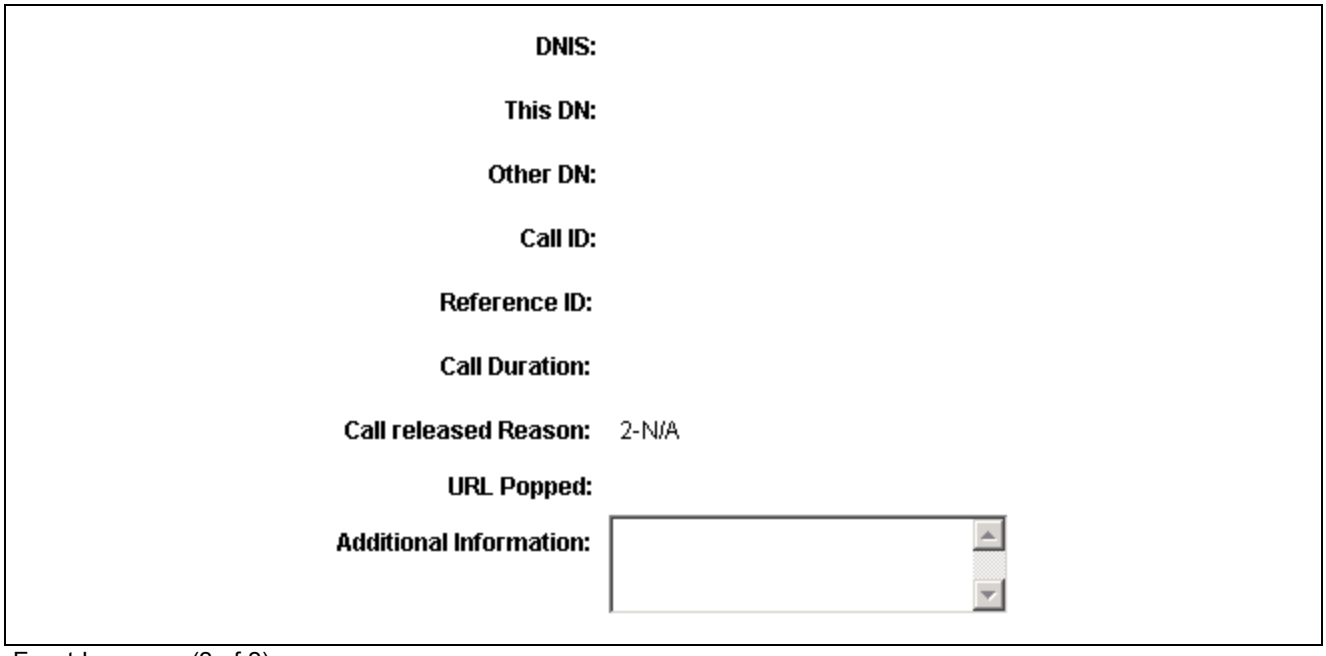
Page Name	Object Name	Navigation	Usage
Event Log	MCFTPEVENTLOG_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Event Log	Displays the details of any event.

Viewing Event Logs for a Third Party

Select any event from the Event Log page to display the details of any event.

Copyright © 1988-2007, Oracle. All rights reserved.

Event Log page (1 of 2)



Event Log page (2 of 2)

 Copyright © 1988-2007, Oracle. All rights reserved.

Viewing Broadcast Logs for a Third Party

To view the broadcast log, access the broadcast log (MCF_TP_BCASTLG_CMP) component.

Page Used to View Broadcast Logs For Third Party

Page Name	Object Name	Navigation	Usage
Broadcast Log	MCF_BCASTLOG_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Broadcast Log	View broadcast logs.

Viewing Broadcast Logs for a Third Party

See Chapter 9, "Administering Queues, Logs, and Tasks," Viewing Broadcast Logs, page 120.

Working with Third-Party Sample Pages

To demonstrate tools and functionality, use the Sample Pages (MCF_TP_DEMO_CMP) component. These sample pages demonstrate how JSMCAPI can be used with the PeopleCode behind it. The third party may develop their own interface, which may look significantly different from the sample pages included in the section.

This section discusses how to:

- Use the Customer Chat sample page.
- Use the Generic Event sample page.
- Use the Email sample page.
- Use the Sample Console page.
- Use JSMCAPI Broadcast page.
- Use PeopleCode Broadcast page.

Copyright © 1988-2007, Oracle. All rights reserved.

Pages Used to Work with Third-Party Sample Pages

Page Name	Object Name	Navigation	Usage
Customer Chat Page	MCFTP_DEMO_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Sample Pages, Customer Chat Page	Send a customer chat session and use built-in functions.
Generic Page	MCFTP_DEMO_NTFY_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Sample Pages, Generic Page	Send a generic event notification and use built-in functions.
Email Page	MCFTP_DEMOEM_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Sample Pages, Email Page	Send an email, use built-in functions, and application package classes.
Sample Console Page	PT_SAMPLETPAGTCNSL	PeopleTools, MultiChannel Framework, Third-Party Configuration, Sample Pages, Sample Console Page	Launch a sample MultiChannel Console generated using JSMCAPI and perform all tasks.
JSMCAPI Page	MCFTP_BROADCAST_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Sample Pages, JSMCAPI Page	Send a broadcast message using JSMCAPI.
PCodeBroadcast Page	MCFTP_PCODEBCST_PG	PeopleTools, MultiChannel Framework, Third-Party Configuration, Sample Pages, PCodeBroadcast Page	Send a broadcast message using PeopleCode.

Using the Customer Chat Sample Page

See Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using the Customer Chat Sample Page, page 148.

Using the Generic Event Sample Page

See Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using the Generic Event Sample Page, page 151.

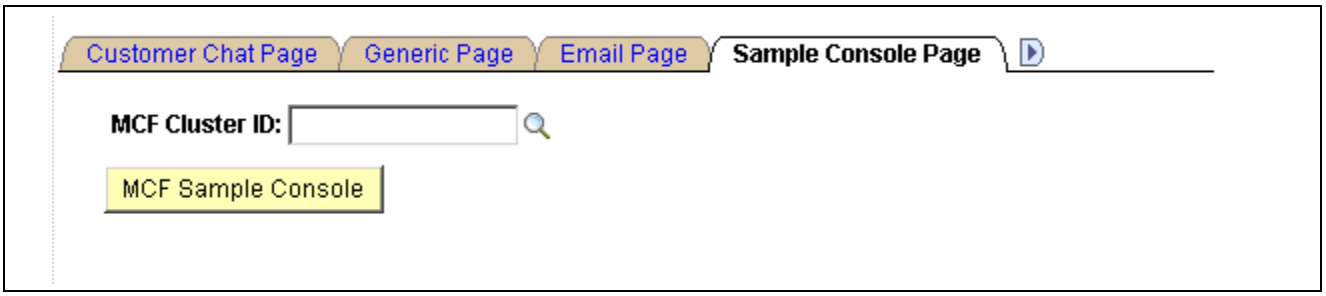
Using the Email Sample Page

See Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using the Email Sample Page, page 153.

Using the Sample Console Page

Access the Sample Console page. Click the lookup button and select the REN cluster.

 Copyright © 1988-2007, Oracle. All rights reserved.



Sample Console page

MCF Sample Console Click to initiate the sample console.

Note. To demonstrate the MCF sample console, you must have an MCF cluster running and communicating with the specified REN server cluster.

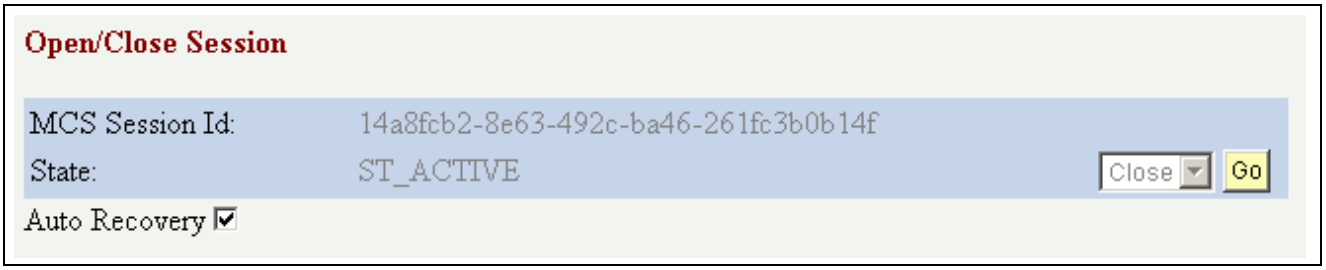
After clicking MCF Sample Console, a new browser window appears that displays the sample console.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.

Before accessing the Open/Close Session group box, enable the third-party routing server.

Open/Close Session

Access the Open/Close Session group box.



Open/Close Session group box

MCS Session Id Displays the MultiChannel server's session ID.

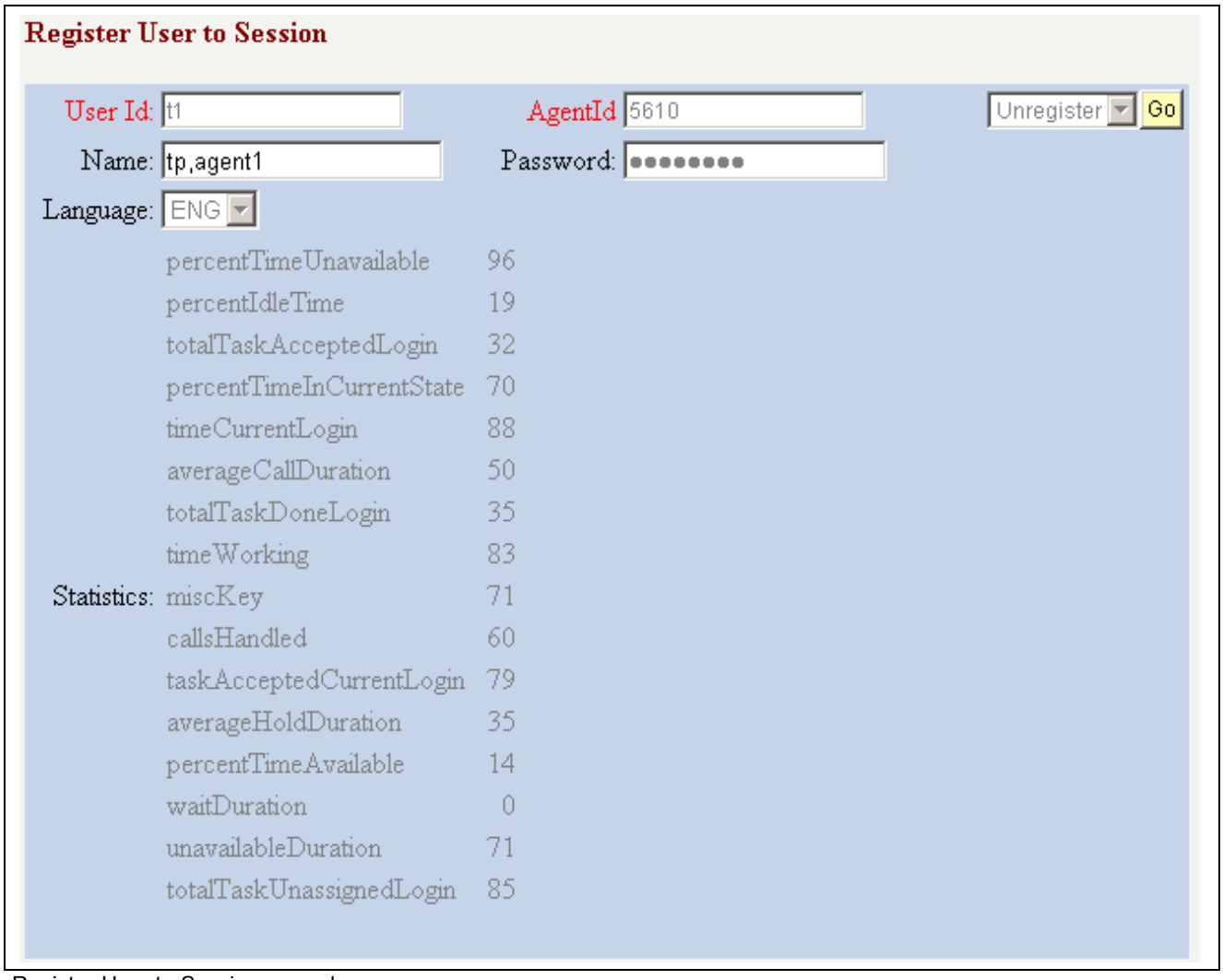
State Displays the state of the session.

Note. If the state is *ST_FAILED*, check the third-party routing server. It must be up and running.

Register User to Session

Access the Register User to Session group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Register User to Session group box

You can register a user to the session in the Register User to Session group box.

User Id Enter user ID of the agent.

Agent Id Enter agent's ID.

Note. Because a CTI agent can have an agent ID different from the agent's user ID, both values are required. If the agent has a password, the password is also required.

Name Enter the name of the agent.

Password Enter the password if the agent has a password.

Language Select the language.

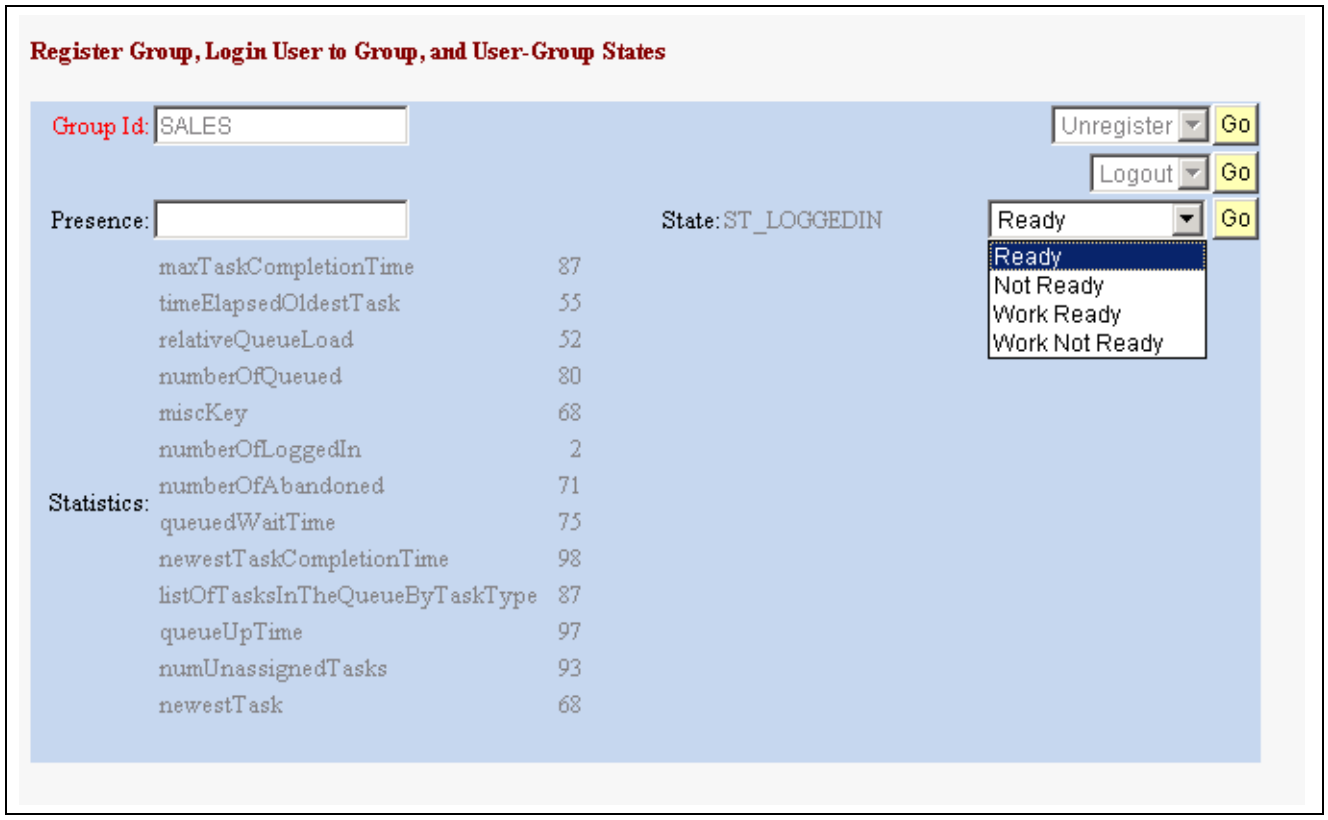
The language field is reserved for future use. The default language is English.

Statistics Displays the agent statistics.

Copyright © 1988-2007, Oracle. All rights reserved.

Register Group, Login User to Group, and User-Group States

Access the Register Group, Login User to Group, and User-Group States group box.



Register Group, Login User to Group, and User-Group States group box

Group Id

Enter the name of the group to which the agent will be registered. The group ID is same as the name of the queue to which the agent is logged in.

To register the agent with the specified group, click the Go button next to the Register field.

Note. After you click Go button next to *Register* field, the field changes to *Unregister*.

Login

Click the Go button to log in if *Login* appears in the drop-down list box.

Click the Go button to log out if *Logout* appears in the drop-down list box.

Note. After you login to the group, a new pop-up window opens prompting for reason code data. If you register to an invalid group, a pop-up window opens showing the error messages.

See Chapter 11, "Using PeopleSoft MCF Broadcast and Working with Sample Pages," Understanding JSMCAPI, page 155.

Presence

Enter a presence description to associate with the state selected in the drop-down list box.

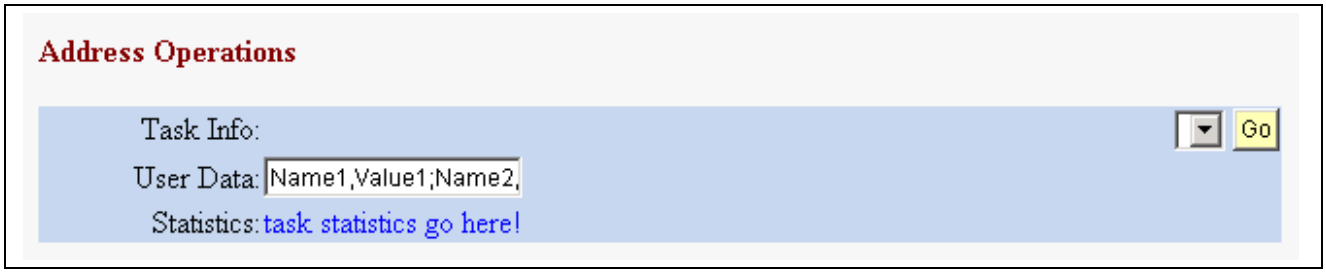
This presence value is different from any presence values predefined in the agent configuration. The agent cannot select from the predefined values.

Copyright © 1988-2007, Oracle. All rights reserved.

State	Select an agent state from the following options: • *Ready* • *Not Ready* • *Work Ready* • *Work Not Ready* Each state value has an associated state code that appears in the State field. To enable incoming calls, select *Ready* from the drop-down list box and click the Go button.
Statistics	Displays the group statistics.

Address Operations

Access the Address Operations group box.



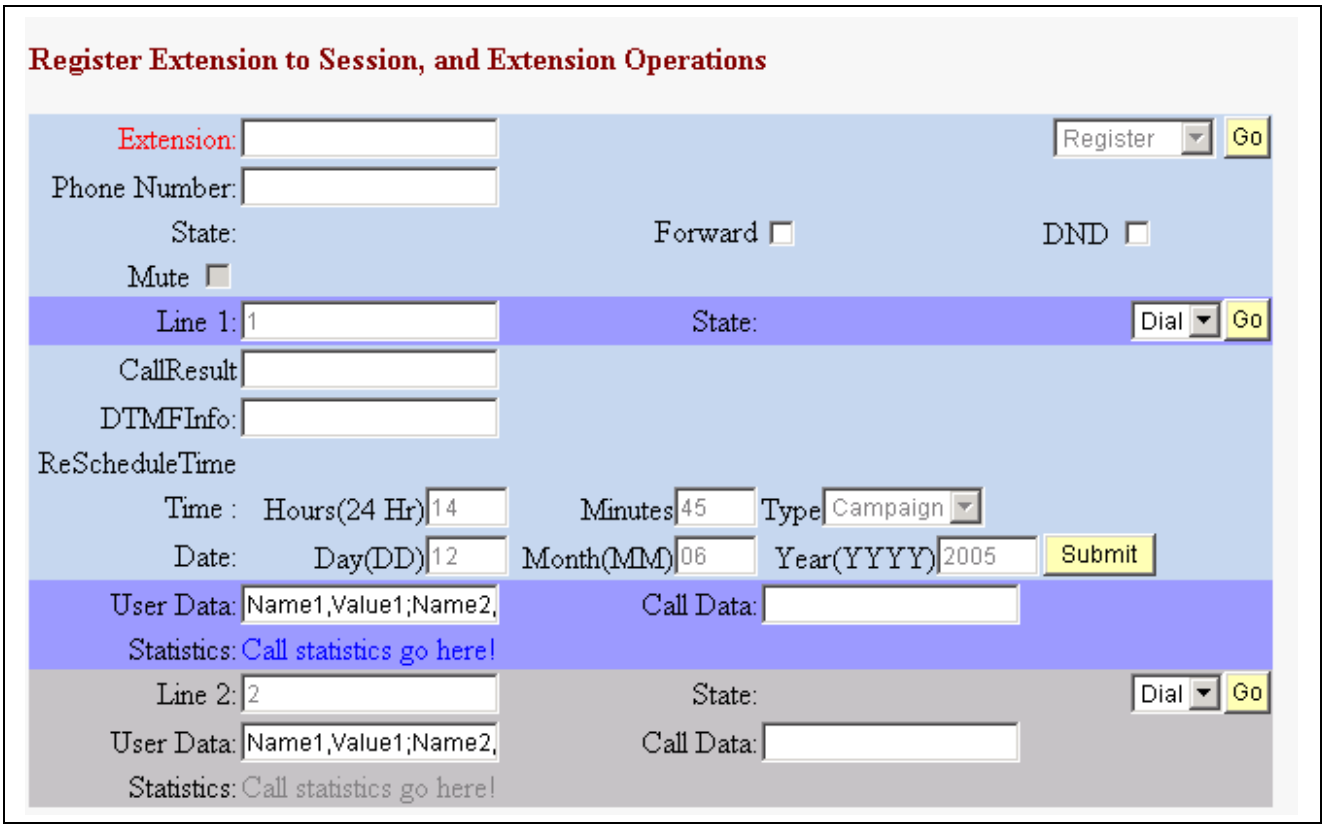
Address Operations group box

Task Info (task information)	Displays the task information.
User Data	Displays the user data.
Statistics	Displays the task statistics.

Register Extension to Session, and Extension Operations

Access the Register Extension to Session, and Extension Operations group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Register Extension to Session, and Extension Operations group box

Extension	Enter a valid telephone extension number.
Phone Number	Enter a valid telephone number for use in subsequent operations, such as dial, forward, or do-not-disturb (DND).
Mute	Select to mute the call.
	Note. Mute is enabled only when the agent is on the call.
Forward	Select to forward incoming calls to the specified number.
DND (do not disturb)	Select to put the selected extension in do-not-disturb status.
Line 1 or **Line 2**	Displays a name or value for each line. On the sample console, the value will either be *1* or *2*.
DTMF Info (Dual Tone Multi-Frequency information)	Select to send DTMF digits on the line.
CallResult	Select to set call result on the line.
ReScheduleTime	Select to set reschedule time on the line.
Type	Select to set reschedule type on the line.
User Data	Enter name-value pairs representing user data to be attached to a call.
Call Data	Enter any call data to be attached to the call.

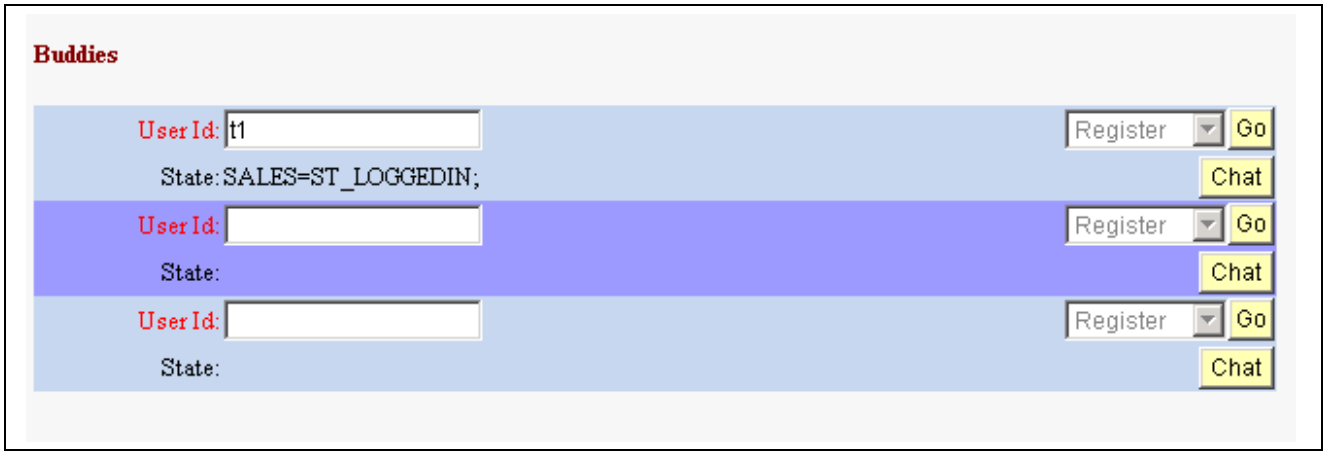
Copyright © 1988-2007, Oracle. All rights reserved.

Call data includes:

- *ANI*
- *DNIS*
- *THISDN*
- *OTHERDN*

Buddies

Access the Buddies group box.



Buddies group box

User Id Enter the user ID of the buddy you want to chat with.

State Displays the state of the buddy.

Click to initiate an agent-to-buddy chat session.

Error Messages/Information

Access the Error Messages/Information group box.

Error Messages/Information group box

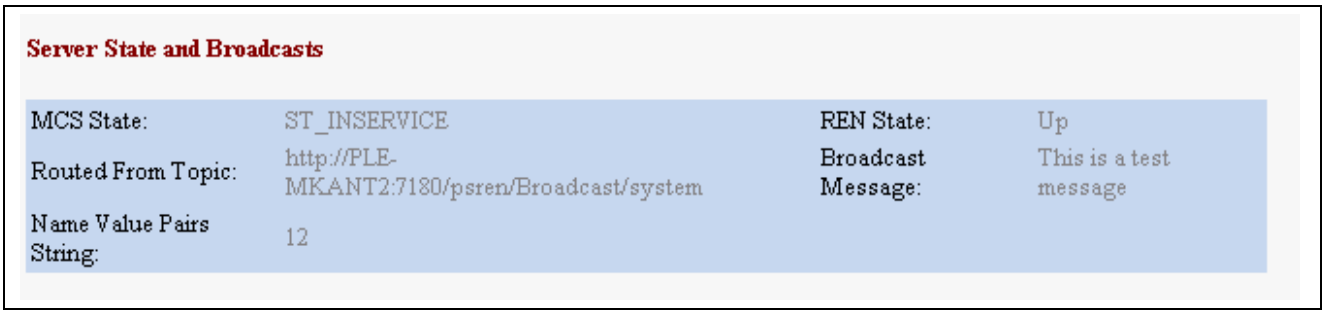
Any error messages associated with the process display here.

Clear Click the button to clear a displayed error message.

Server State and Broadcasts

Access the Server State and Broadcasts group box.

Copyright © 1988-2007, Oracle. All rights reserved.



Server State and Broadcasts group box

MCS State Displays the MultiChannel server's state.

Routed From Topic Displays the machine name from where the broadcast message was sent.

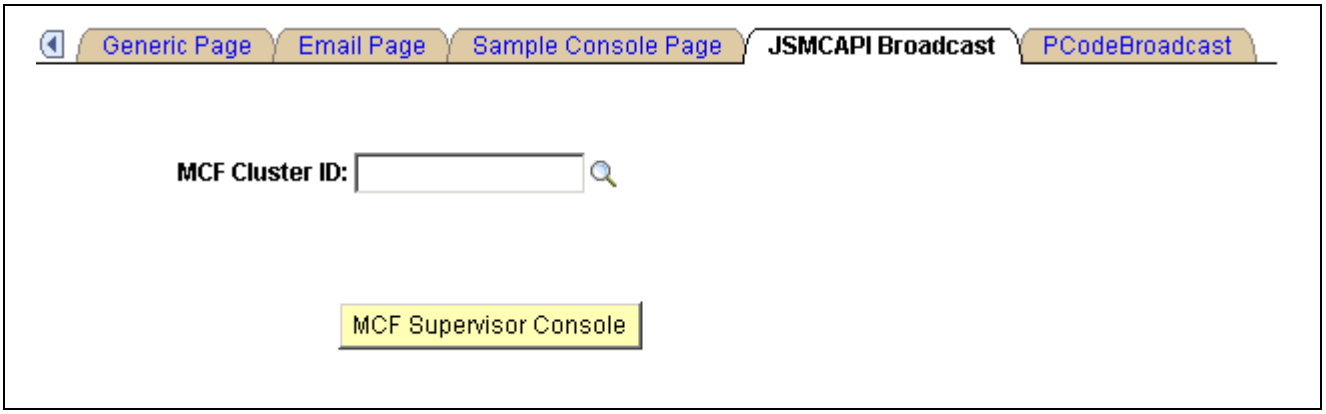
Name Value Pairs Displays the name value pair string.

REN State Displays the status of the REN server.

Broadcast Message Displays the message that was broadcasted by the supervisor.

Using the MCF Broadcast Page

This section discusses how to use the JSMCAPI Broadcast page for the third party.



JSMCAPI Broadcast page

MCF Server Cluster ID Select the REN server cluster on which to test the MCF Supervisor console.

Note. The MCF Supervisor console is specific to the REN cluster Id selected.

MCF Supervisor Console Click to initiate the supervisor console.

Note. To demonstrate the MCF Supervisor console, you must have MCF servers, both queue and log servers, running and communicating with the specified REN server cluster.

After you click MCF Supervisor console, a new browser window appears that displays the sample supervisor console.

Note. To enable the new browser window to appear, disable any pop-up blocking software for your browser.

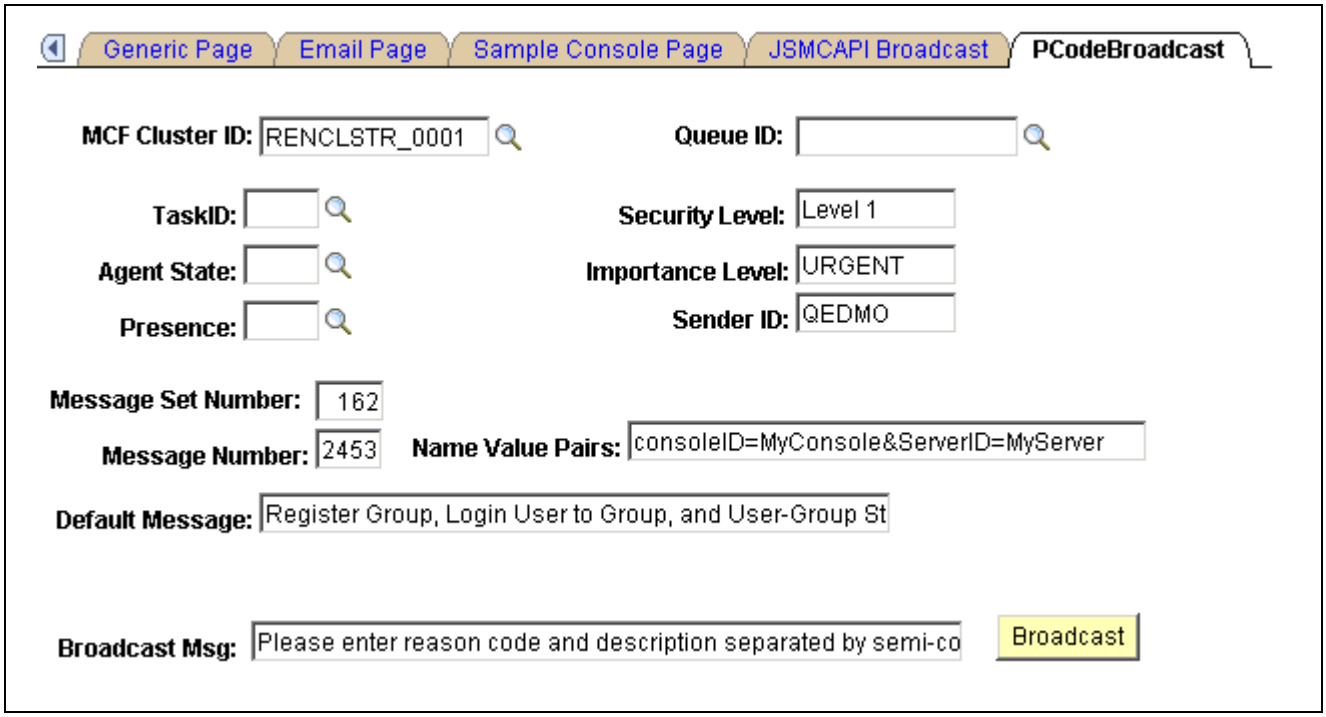
Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using JSMCAPI Broadcast with MCF Supervisor Console, page 139

Using the PCodeBroadcast Page

This section discusses how to use the PCodeBroadcast page.



PCodeBroadcast Page

See Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using PeopleCode Broadcast, page 144.

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 13

Understanding JSMCAPI Classes

This chapter discusses the JavaScript MultiChannel Application Programming Interface (JSMCAPI) classes.

Understanding JSMCAPI

This section introduces the JSMCAPI, a JavaScript-based API used to build custom applications, such as MultiChannel consoles, or to enable MultiChannel functionality on the PeopleSoft Pure Internet Architecture page. The API is built upon the REN server JavaScript client and is a pure JavaScript API.

JSMCAPI is the interface through which the application developer accesses the JSMCAPI functionalities. JSMCAPI provides a set of objects such as User, Address, Group, and so on. The PSMC, a global object, provides all those methods called to send agent requests to the server; it also provides an event handler callback method for the PeopleSoft MultiChannel Application Programming Interface (PSMCAPI) call backs for events coming from the CTI server.

The JSMCAPI communicates with the PSMCAPI through the REN server using MCP (Multi-Channel Protocol). MCP includes topic and event definition. In general, JSMCAPI:

- Sends agent requests to the MultiChannel Server.

  The requests can be agent state requests such as login, logout, set ready, and so on. Requests can also be call requests such as dial-out, answer the call, transfer/conference, and so on.

- Receives PBX/Switch events from the CTI server.

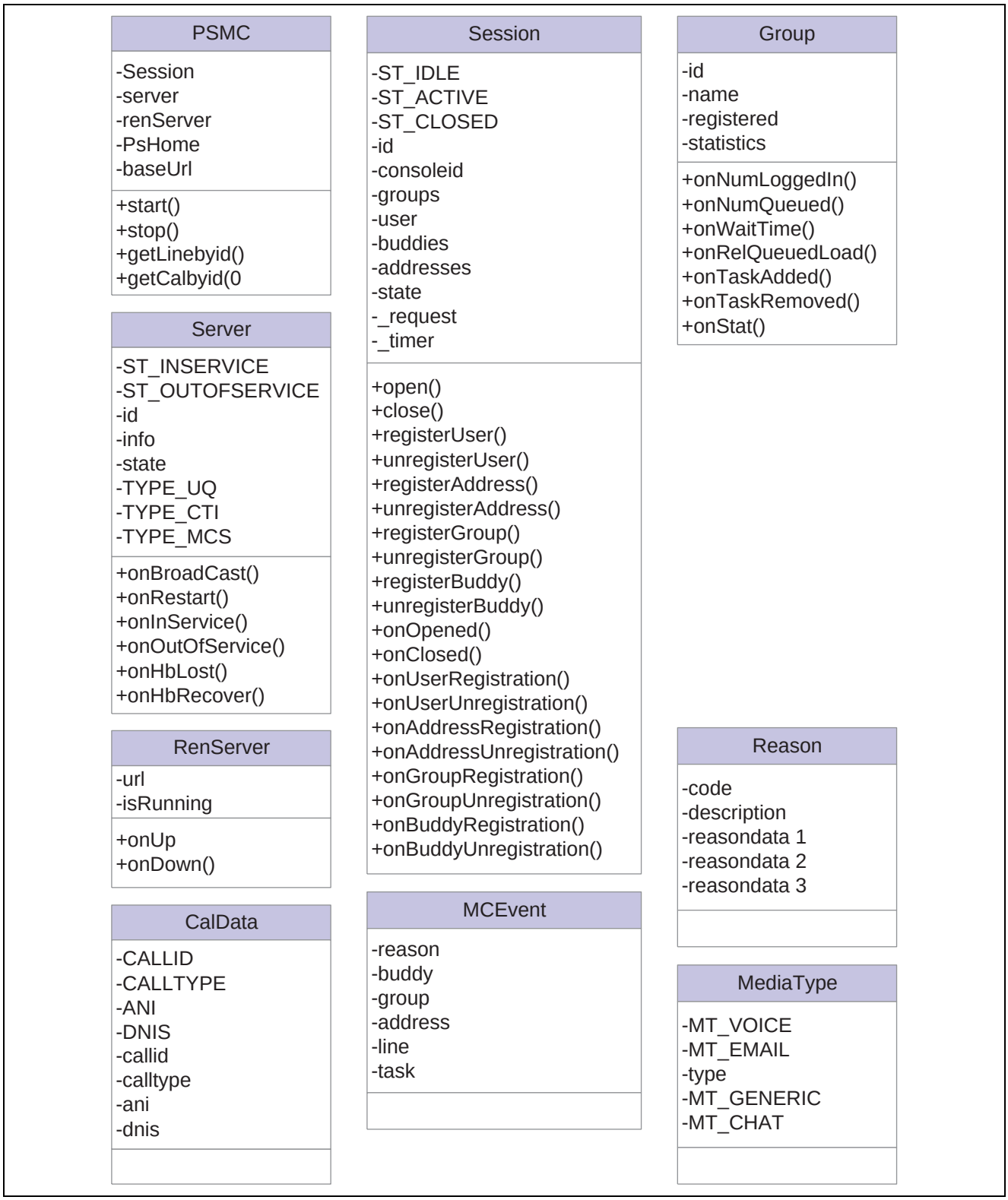
  Events can be the agent state change such as event ready, not ready, and so on. Events can also be call events such as incoming call, call released, call transferred, and so on.

Understanding JSMCAPI Classes

This sections gives an overview of all the JSMCAPI classes.

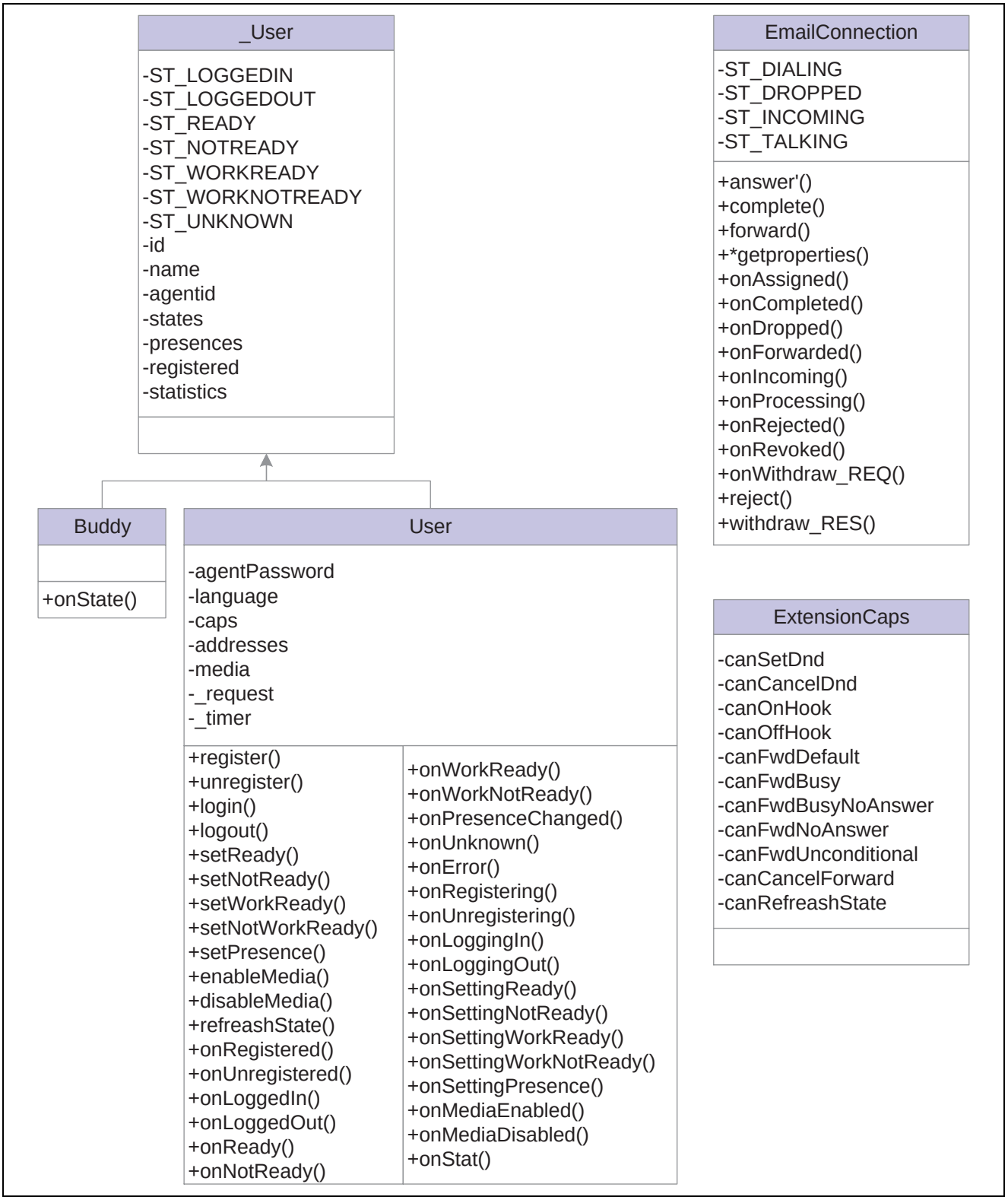
The following class diagrams explain all the classes, fields, and the methods.

Copyright © 1988-2007, Oracle. All rights reserved.



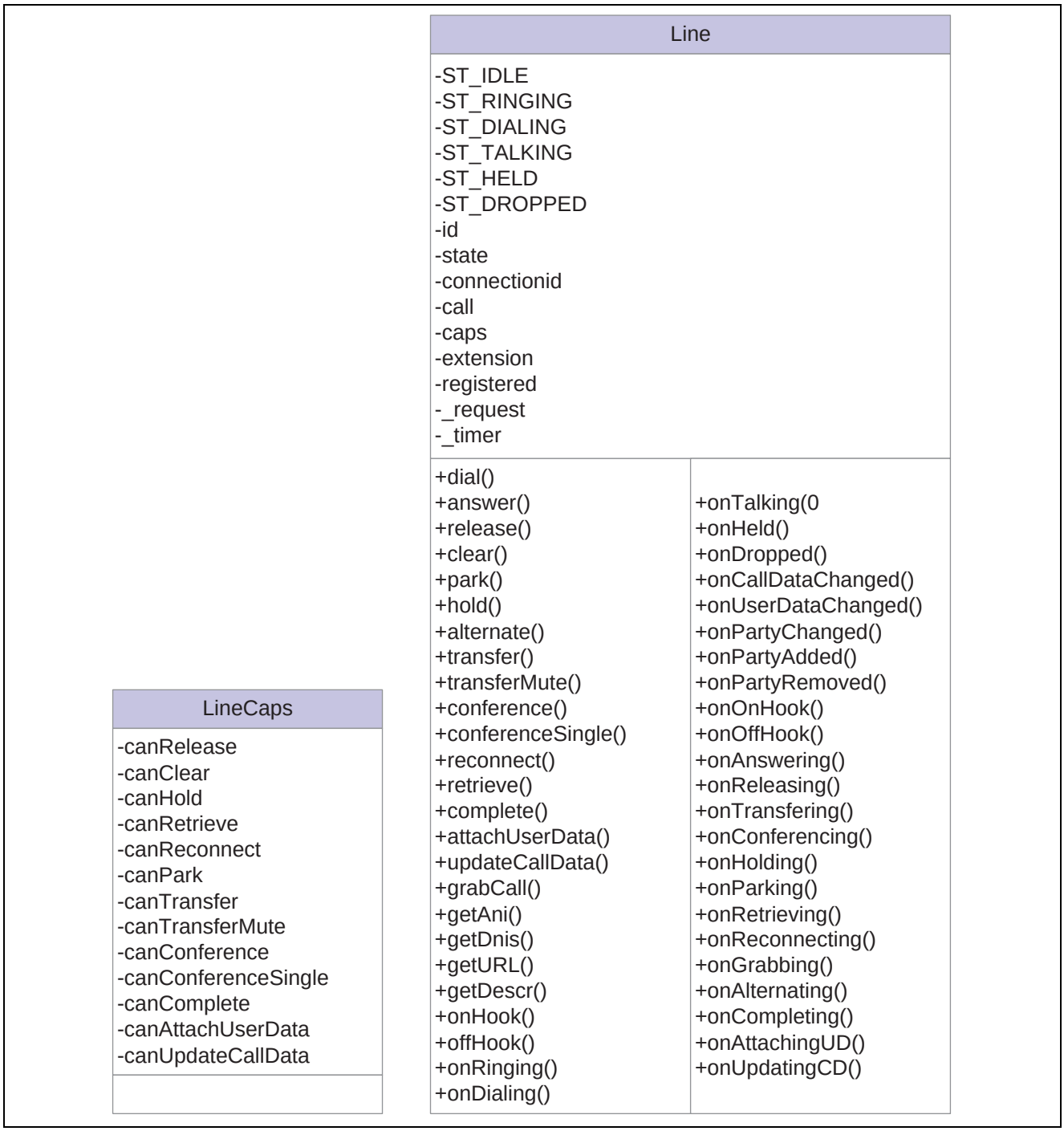
Class diagram Part 1

Copyright © 1988-2007, Oracle. All rights reserved.



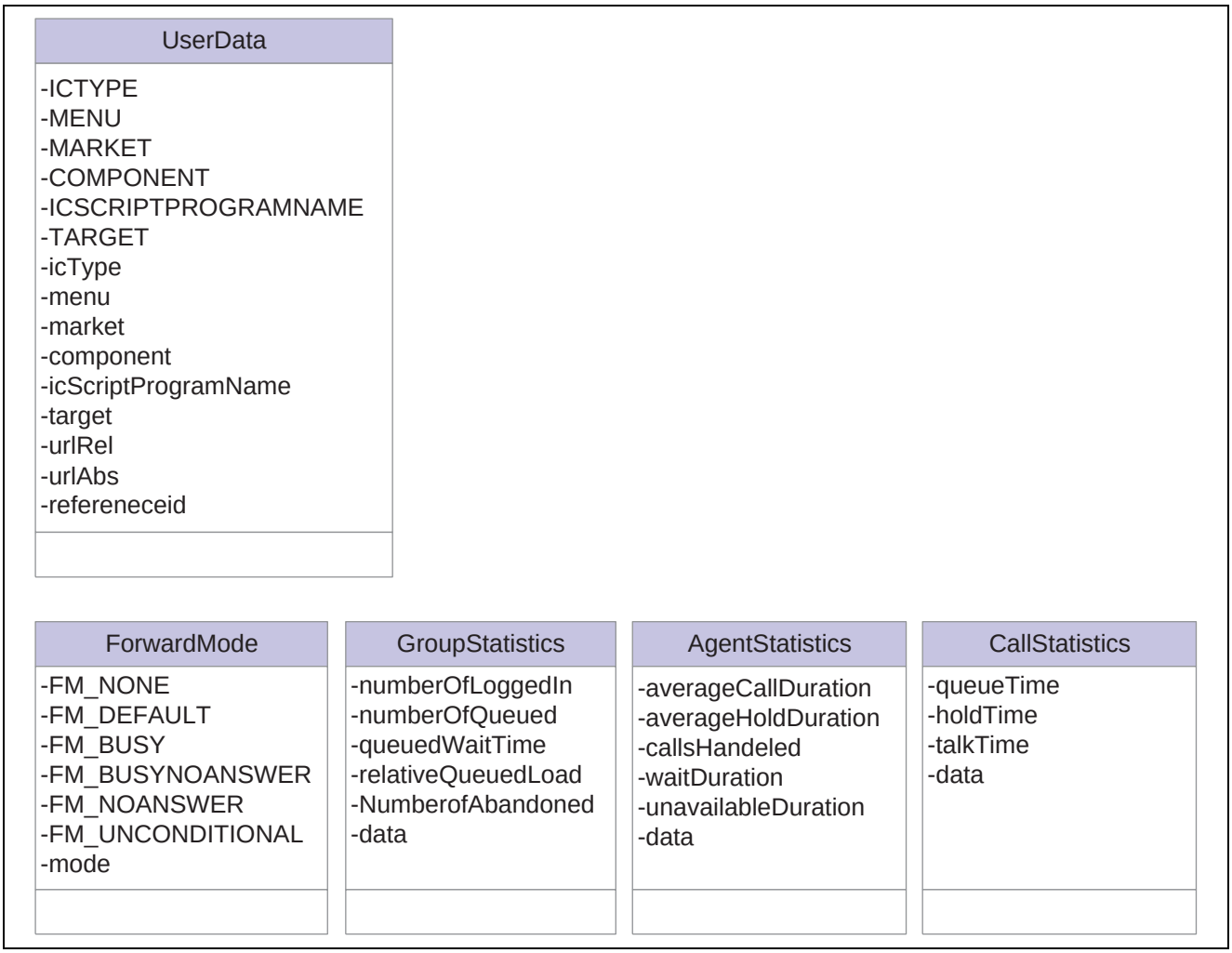
Class diagram Part 2

Copyright © 1988-2007, Oracle. All rights reserved.



Class diagram Part 3

Copyright © 1988-2007, Oracle. All rights reserved.



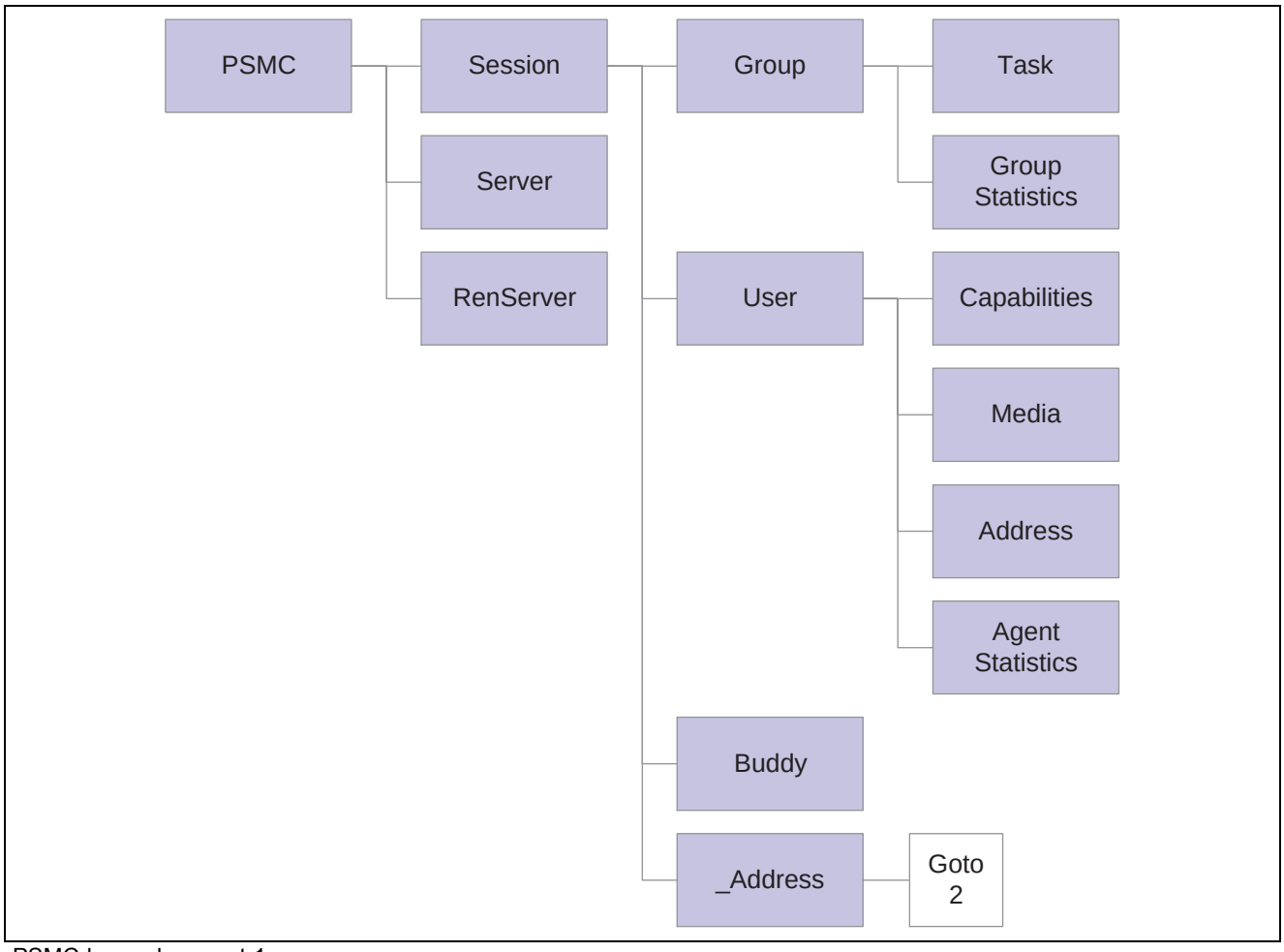
Class diagram Part 4

PSMC

PSMC is the base class that an application accesses to start an instance of JSMCAPI. The application can access all JSMCAPI functionality through this object. A session object is created from this class.

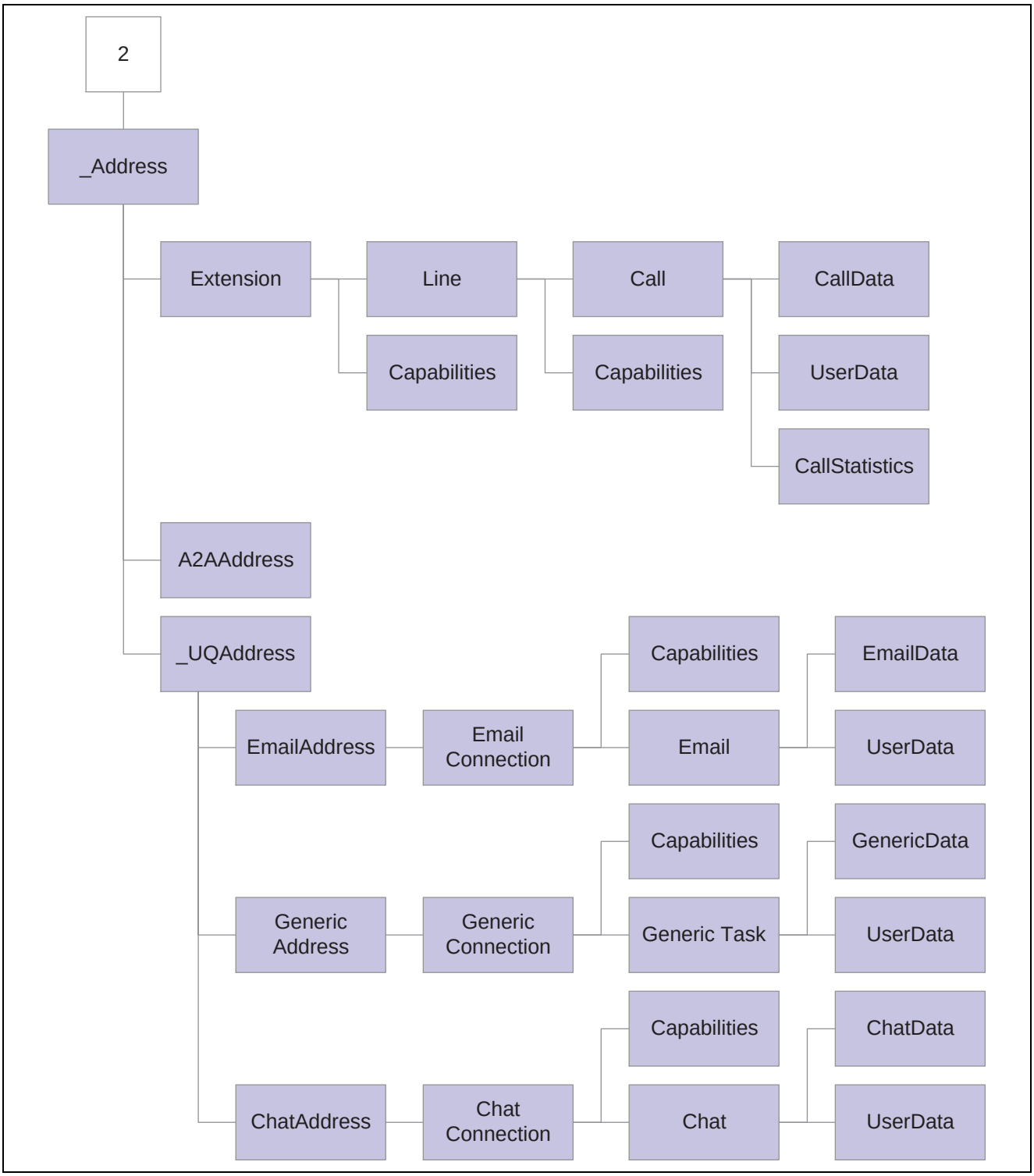
The following diagram explains the relationship of PSMC with all other classes.

Copyright © 1988-2007, Oracle. All rights reserved.



PSMC base class-part 1

Copyright © 1988-2007, Oracle. All rights reserved.



PSMC base class-part 2

See Also

Chapter 13, “Understanding JSMCAPI Classes,” PSMC Class Constructor, page 353

Copyright © 1988-2007, Oracle. All rights reserved.

Server

The Server class refers to the routing server and can execute events for specific server states. Three types of servers are available, CTI, queue server, and MultiChannel server. The constants are CTI, UQ and MCS respectively.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Server Class Constructor, page 360

RENServer

The RENServer cluster is represented in the RENServer class. It provides the URL and the server status (active or shutdown).

See Also

Chapter 13, "Understanding JSMCAPI Classes," RenServer Class Constructor, page 359

Session

Sessions are set for users registering with the Server. Addresses, buddies, and groups are registered for the user with the session. The connection between the server and JSMCAPI is a session. There is only one session object per PSMC object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Session Class Constructor, page 363

_Address

The _Address class identifies the user with a unique ID. The ID identifies the user to the routing system and is unique for each channel or mediatype.

The subclasses of _Address class are:

- Extension
- _UQAddress
- A2AchatAddress

Further delineation of address by media type is provided by:

- ChatAddress
- EmailAddress
- GenericAddress

ChatAddress, EmailAddress, and GenericAddress classes extend _UQAddress.

 Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Extension Class Constructor, page 287

Chapter 13, “Understanding JSMCAPI Classes,” _UQAddress Class Constructor, page 227

Chapter 13, “Understanding JSMCAPI Classes,” A2AChatAddress Class Constructor, page 237

Chapter 13, “Understanding JSMCAPI Classes,” ChatAddress Class Constructor, page 255

Chapter 13, “Understanding JSMCAPI Classes,” EmailAddress Class Constructor, page 275

Chapter 13, “Understanding JSMCAPI Classes,” GenericAddress Class Constructor, page 297

Line

Line class describes the line of the extension for a call task. This version supports one extension with two lines and two extensions with one line in each.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Line Class Constructor, page 325

Connection

Tasks are routed to agents through a connection. Email, chat and generic have a dedicated connection class, like EmailConnection, ChatConnection, and GenericConnection. These connections provide task specific manipulation functions.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” ChatConnection Class Constructor, page 258

Chapter 13, “Understanding JSMCAPI Classes,” EmailConnection Class Constructor, page 278

Group

The group class defines the group or queue information. There are one or more groups objects per session.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Group Class Constructor, page 316

Task

This is an abstract base class that defines a unit of work. The classes that extend task per media type are:

- Call
- Email
- Chat
- A2AChat
- GenericTask

Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Call Class Constructor, page 246

Chapter 13, “Understanding JSMCAPI Classes,” Email Class Constructor, page 272

Chapter 13, “Understanding JSMCAPI Classes,” Chat Class Constructor, page 251

Chapter 13, “Understanding JSMCAPI Classes,” A2AChat Class Constructor, page 233

Chapter 13, “Understanding JSMCAPI Classes,” GenericTask Class Constructor, page 309

_User

_User is a base class. The subclasses are User and Buddy. These classes define the properties that pertain to the user such as the addresses, languages or presence. _User is a virtual class and should not be instantiated.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” _User Class Constructor, page 230

MediaType

This class defines the media that an agent can handle.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” MediaType Class Constructor, page 352

Reason

The Reason class defines the message or error message that accompanies an event or request. Globalization of the messages is implemented. Furthermore extra data can be passed in this object for providing a detailed message.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Reason Class Constructor, page 357

Statistics

Statistics are provided by the routing server for agent, call, task, group, and user. The following classes describe the statistics for each component:

- AgentStatistics
- CallStatistics
- TaskStatistics
- GroupStatistics
- GroupStatistics1
- GroupStatistics2
- UserStatistics1
- UserStatistics2

 Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” AgentStatistics Class Constructor, page 239

Chapter 13, “Understanding JSMCAPI Classes,” CallStatistics Class Constructor, page 250

Chapter 13, “Understanding JSMCAPI Classes,” TaskStatistics Class Constructor, page 376

Chapter 13, “Understanding JSMCAPI Classes,” GroupStatistics Constructor, page 319

Chapter 13, “Understanding JSMCAPI Classes,” GroupStatistics1 Class Constructor, page 321

Chapter 13, “Understanding JSMCAPI Classes,” GroupStatistics2 Class Constructor, page 324

Chapter 13, “Understanding JSMCAPI Classes,” UserStatistics1 Class Constructor, page 396

Chapter 13, “Understanding JSMCAPI Classes,” UserStatistics2 Class Constructor, page 399

Data

When a task is introduced to the system, data pertaining to the task is defined in the following classes:

- CallData
- EmailData
- ChatData
- GenericData

Application specific data is provided for task via the AppData class. Similarly, user data is defined in UserData.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” CallData Class Constructor, page 248

Chapter 13, “Understanding JSMCAPI Classes,” EmailData Class Constructor, page 286

Chapter 13, “Understanding JSMCAPI Classes,” ChatData Class Constructor, page 271

Chapter 13, “Understanding JSMCAPI Classes,” GenericData Class Constructor, page 308

Globals

This class defines functions that are used universally.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” GLOBALS Class Fields, page 312

MCEvent

Events passed to the application handler are defined by MCEvent.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” MCEvent Class Constructor, page 350

Copyright © 1988-2007, Oracle. All rights reserved.

Caps

Capabilities define the ability or allowed actions for a component. The following classes define specific capabilities:

- ChatConnectionCaps
- EmailConnectionCaps
- GenericConnectionCaps
- ExtensionCaps
- LineCaps
- ChatConnectionCaps
- EmaiConnectionCaps
- GenericConnectionCaps
- UserCaps

See Also

Chapter 13, “Understanding JSMCAPI Classes,” ChatConnectionCaps Class Constructor, page 269

Chapter 13, “Understanding JSMCAPI Classes,” EmailConnectionCaps Class Constructor, page 285

Chapter 13, “Understanding JSMCAPI Classes,” GenericConnectionCaps Class Constructor, page 306

Chapter 13, “Understanding JSMCAPI Classes,” ExtensionCaps Class Constructor, page 294

Chapter 13, “Understanding JSMCAPI Classes,” LineCaps Class Constructor, page 346

ForwardMode

ForwardMode describes various forward modes that can be used while setting the forwarding mode for a Address.

See Also

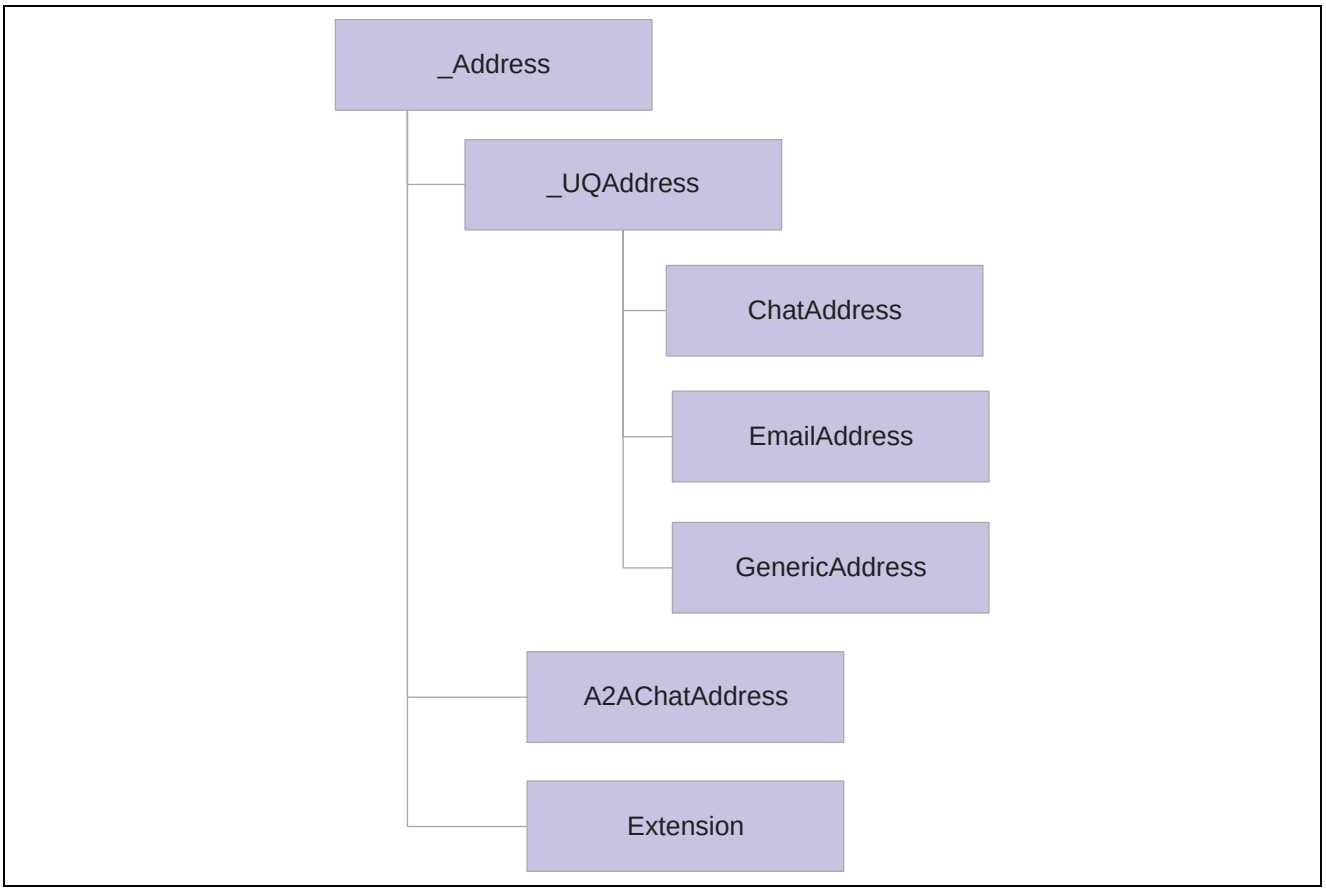
Chapter 13, “Understanding JSMCAPI Classes,” ForwardMode Class Constructor, page 296

_Address Class Hierarchy

The _Address class can be extended by other subclasses. The following flowchart shows the different subclasses, and how they interrelate.

Do not instantiate _Address or _UQAddress class objects. Instead, use the child classes.

 Copyright © 1988-2007, Oracle. All rights reserved.



_Address class hierarchy

See Also

Chapter 13, “Understanding JSMCAPI Classes,” _Address Class Constructor, page 225

Chapter 13, “Understanding JSMCAPI Classes,” _UQAddress Class Constructor, page 227

Chapter 13, “Understanding JSMCAPI Classes,” A2AChatAddress Class Constructor, page 237

Chapter 13, “Understanding JSMCAPI Classes,” Extension Class Constructor, page 287

Chapter 13, “Understanding JSMCAPI Classes,” ChatAddress Class Constructor, page 255

Chapter 13, “Understanding JSMCAPI Classes,” EmailAddress Class Constructor, page 275

Chapter 13, “Understanding JSMCAPI Classes,” GenericAddress Class Constructor, page 297

_Address Class Constructor

The following is the _Address class constructor.

Copyright © 1988-2007, Oracle. All rights reserved.

_Address

Syntax

```
_Address()
```

Description

Creates an _Address object that describes the address.

Parameters

None.

Returns

An _Address object.

_Address Class Fields

In this section we discuss the JSMCAPI _Address class fields. These fields are described in alphabetical order.

caps

Description

The capacities of the address.

Type: object.

id

Description

The ID of the address.

Type: string.

_Address Class Callback Event Method

The following is the callback event method used with a JSMCAPI _Address object.

onError

Syntax

```
onError(event)
```


Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires on the event of an address error.

_UQAddress Class Constructor

The following is the UQAddress class constructor.

_UQAddress

Syntax

```
_UQAddress()
```

Description

Creates a _UQAddress object, which is the base for the various addresses associated with the universal queue server.

Parameters

None.

Returns

A universal queue address object.

_UQAddress Class Fields

The _UQAddress class inherits the following fields from the _Address class.

- caps
- id

See Chapter 13, “Understanding JSMCAPI Classes,” _Address Class Fields, page 226.

The following are the _UQAddress class fields.

Tasks

Description

The associative array of the tasks on the queue managed by the address.

_UQAddress Methods

The following are the _UQAddress methods.

Copyright © 1988-2007, Oracle. All rights reserved.

acceptTask

Syntax

acceptTask(*task*, *reason*)

Description

Sends a request to the universal queue server to signal that the client has accepted the task.

Parameters

Parameter	Description
task	The task ID.
reason	The associated reason code.

Returns

None.

dequeueTask

Syntax

dequeueTask(*task*)

Description

Sends a request to the universal queue server to remove the task from its queue.

Parameters

Parameter	Description
task	The task ID.

Returns

None.

_UQAddress Class Callback Event Methods

The _UQAddress class inherits the onError callback event method from the _Address class.

See Chapter 13, “Understanding JSMCAPI Classes,” _Address Class Callback Event Method, page 226.

The following are the callback event methods used with a JSMCAPI universal queue address object. The callback event methods are described in alphabetical order.

 Copyright © 1988-2007, Oracle. All rights reserved.

onAccepted

Syntax

onAccepted(*event*)

Description

Fires when the task is accepted.

onAcceptingTask

Syntax

onAcceptingTask(*event*)

Description

Fires as the task is being accepted.

onDequeueingTask

Syntax

onDequeueingTask(*event*)

Description

Fires as the task is being dequeued.

onNotify

Syntax

onNotify(*event*)

Description

Fires on task notification.

onTaskAdded

Syntax

onTaskAdded(*event*)

Description

Fires when the task is added to the addressed queue.

onTaskRemoved

Syntax

onTaskRemoved(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fired when the task is removed

onUnassigned

Syntax

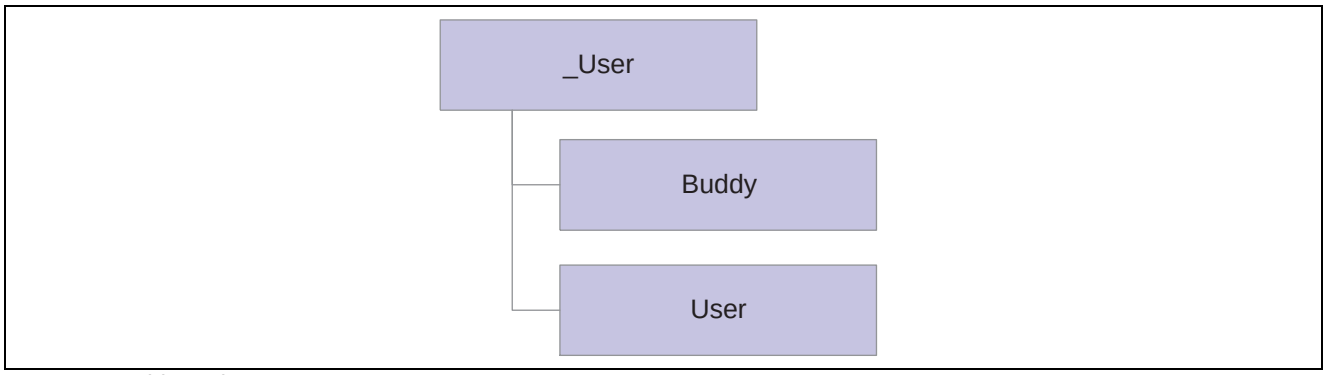
onUnassigned(*event*)

Description

Fired when the task is unassigned.

_User Class Hierarchy

The _User class can be extended by other subclasses. The following flowchart shows the different subclasses, and how they interrelate:



_User class hierarchy

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Buddy Class Constructor, page 245

Chapter 13, “Understanding JSMCAPI Classes,” User Class Constructor, page 377

_User Class Constructor

The following is the _User class constructor.

_User

Syntax

_User()

 Copyright © 1988-2007, Oracle. All rights reserved.

Description

Describes the base class of user/agent.

Parameters

None.

Returns

A _User object.

_User Class Fields

The following are the fields associated with the JSMCAPI _User class. These fields are discussed in alphabetical order.

agentID

Description

The agent's agent ID.

Type: string.

caps

Description

The agent capabilities on each group.

Type: associative array.

id

Description

The agent's PeopleSoft user ID.

Type: string.

name

Description

The agent's user name.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

presences

Description

The agent's presence on each group.

Type: associative array.

states

Description

Agent state on each group.

Type: associative array, including the constants beginning with ST_*.

These constants must be accessed from an instantiated object.

ST_LOGGEDIN

Description

The agent is logged in.

ST_LOGGEDOUT

Description

The agent is logged out.

ST_NOTREADY

Description

The agent is not ready.

ST_READY

Description

The agent is ready.

ST_UNKNOWN

Description

The agent's state is unknown.

ST_WORKNOTREADY

Description

The agent is in the work not ready state.

 Copyright © 1988-2007, Oracle. All rights reserved.

ST_WORKREADY

Description

The agent is in the work ready state.

statistics

Description

Agent statistics for CTI.

Type: AgentStatistics object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," AgentStatistics Class Constructor, page 239

statistics1

Description

Agent statistics for the universal queue server.

Type: UserStatistics1 object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," UserStatistics1 Class Constructor, page 396

statistics2

Description

Agent statistics for the universal queue server.

Type: UserStatistcs2 object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," UserStatistics2 Class Constructor, page 399

A2AChat Class Constructor

The following is the A2AChat class constructor.

A2AChat

Syntax

A2AChat(*event*,*address*,*chatType*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Creates an agent-to-agent chat object.

The chat is related to the A2AChatAddress. It does not extend Task as it is not generated or tracked by the universal queue server. Construction occurs inside the A2AChatAddress.

Parameters

Parameter	Description
event	Enter the event associated with the agent-to-agent chat.
address	Enter the address from A2AChatAddress.
chatType	Enter the chat type, for example consult or answer.

Returns

An agent-to-agent chat object.

A2AChat Class Fields

The following are the A2AChat fields.

address

Description

The address containing the agent-to-agent chat.

Type: A2AChatAddress object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," A2AChatAddress Class Constructor, page 237

agentID

Description

The associated agent's ID.

Type: string.

agentName

Description

The associated agent's name.

Type: string.

 Copyright © 1988-2007, Oracle. All rights reserved.

appData

Description

The application data that is provided to the client with the notification event.

Type: AppData object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," AppData Class Constructor, page 242

chatType

Description

The type of chat, either A2AChat.TYPE_CONSULT or A2AChat.TYPE_ANSWER.

Type: string, with one of the following constants:

Value	Description
TYPE_ANSWER	The A2AChat type answer. This type is generated when another agent wants this agent to answer an A2AChat.
TYPE_CONSULT	The A2AChat type consult. This type is generated when this agent wants to consult another agent.

Example

The A2AChat.TYPE_* can only be accessed as in the following example:

```
var value = A2AChat.TYPE_CONSULT;
```

See Also

Chapter 13, "Understanding JSMCAPI Classes," GLOBALS Class Fields, page 312

customerName

Description

The name of the customer in the ongoing chat.

Type: object.

id

Description

The ID of this chat.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

isConference

Description

A flag to check whether it is conference or not.

Type: object.

jr

Description

JournalRoute

Type: string.

question

Description

Question that is asked when customer initiates chat.

Type: object.

subject

Description

Subject of the chat. Customer will provide this subject when he initiates chat.

Type: object.

type

Description

The task type, chat.

Type: string.

uniqueueId

Description

The unique ID.

Type: string.

A2AChat Class Method

The following is the A2AChat class method.

 Copyright © 1988-2007, Oracle. All rights reserved.

getURL

Syntax

getURL(*defaultURL*)

Description

Returns the URL for the given task.

Parameters

Parameter	Description
defaultURL	If not null, this value will override the generated base URL.

Returns

The URL associated with this task.

A2AChatAddress Class Constructor

The following is the A2AChatAddress constructor.

A2AChatAddress

Syntax

A2AChatAddress()

Description

Agent-to-agent chat address. Handles the creation of agent-to-agent chats for incoming and outgoing chat communication.

Parameters

None.

Returns

An A2AChatAddress object.

A2AChatAddress Class Fields

The A2AChatAddress class inherits the following fields from the _Address class:

- id
- caps

Copyright © 1988-2007, Oracle. All rights reserved.

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Fields, page 226.

The following are the fields associated with the A2AChatAddress class.

id

Description

The address ID.

Type: string.

tasks

Description

The array of A2AChats associated with this address.

Type: associative array.

See Also

Chapter 13, "Understanding JSMCAPI Classes," GLOBALS Class Fields, page 312

A2AChatAddress Class Method

The following is the A2AChatAddress class method.

initiateChat

Syntax

initiateChat(*agentId*)

Description

Initializes the A2AChatAddress by creating an A2AChat task.

Parameters

Parameter	Description
agentId	The ID of the agent with whom you want to chat.

Returns

None.

 Copyright © 1988-2007, Oracle. All rights reserved.

A2AChatAddress Class Callback Event Methods

The A2AChatAddress class inherits the onError callback event method from the _Address class.

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Callback Event Method, page 226.

The following are the A2AChatAddress class callback methods:

onChatEnded

Syntax

onChatEnded(*event*)

Description

This event gets called when an agent to agent chat is closed.

onInitiatingChat

Syntax

onInitiatingChat(*event*)

Description

This event gets called when chat is getting initiated.

onNotify

Syntax

onNotify(*event*)

Description

This event gets called when there is a new A2Achat notification.

AgentStatistics Class Constructor

The following is the AgentStatistics class constructor.

AgentStatistics

Syntax

AgentStatistics()

Description

The agent statistics information.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

None.

Returns

An AgentStatistics object.

AgentStatistics Class Fields

The following are the AgentStatistics class fields.

averageCallDuration

Description

The average call duration, in seconds, for an agent.

Type: number

averageHoldDuration

Description

The average hold duration, in seconds, for an agent.

Type: number.

callsHandled

Description

The total number of calls handled by an agent.

Type: number.

data

Description

An associative array of key value pairs that includes all agent statistics.

Type: associative array.

percentIdleTime

Description

The percentage of time for which the agent is idle.

Type: number

 Copyright © 1988-2007, Oracle. All rights reserved.

percentTimeAvailable

Description

The percentage of time for which the agent is in ready state.

Type: number

percentTimeInCurrentState

Description

The percentage of time in current state.

Type: number

percentTimeUnavailable

Description

The percentage of time for which the agent is in not ready state.

Type: number

timeCurrentLogin

Description

The time since login for the agent.

Type: number.

timeWorking

Description

The time spent working on tasks.

Type: number.

totalTaskAcceptedLogin

Description

The number of tasks accepted since login time

Type: number

totalTaskDoneLogin

Description

Number of tasks done by the agent since login time i.e. current login.

Type: number.

Copyright © 1988-2007, Oracle. All rights reserved.

totalTaskUnassignedLogin

Description

Number of tasks unassigned since login time

Type: number.

unavailableDuration

Description

The total time for which an agent is unavailable.

Type: string.

waitDuration

Description

The total time an agent has to wait for a call.

Type: string.

AppData Class Constructor

The following is the AppData class constructor.

AppData

Syntax

```
AppData()
```

Description

The AppData object describes the key-value pairs of the data received on the agent-to-agent chat event.

Parameters

None.

Returns

Type: AppData object.

AppData Class Fields

The following are the AppData class fields.

Copyright © 1988-2007, Oracle. All rights reserved.

data

Description

An associative array of the data in name value pairs.

Type: associative array.

groupId

Description

The group ID passed in by the notify event.

Type: string.

jr

Description

The JournalRoute.

Type: string.

question

Description

The initial question of the task.

Type: string.

strData

Description

The raw application data as a string.

Type: string.

subject

Description

The subject of the task.

Type: string.

uniqueId

Description

The unique ID.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

url

Description

The URL passed in by the AppData.

Type: string with the following constant:

Value	Description
URL	The associated URL.

userId

Description

The agent's user ID.

Type: string.

username

Description

The agent's user name.

Type: string.

wizUrl

Description

The wizard URL that is used for popping up new task window.

Type: string.

AppData Class Method

The following is the AppData class method.

addKeyValue

Syntax

addKeyValue(*key, value*)

Description

Add the key value pair to the AppData object.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
key	The key.
value	The value.

Returns

None.

Buddy Class Constructor

The Buddy class extends the _User class.

See Chapter 13, "Understanding JSMCAPI Classes," _User Class Constructor, page 230.

The following is the Buddy class constructor.

Buddy

Syntax

```
Buddy()
```

Description

Describes the buddy. An agent can register buddies and is notified of state changes.

Parameters

None.

Returns

A Buddy object.

Buddy Class Fields

The Buddy class inherits the following fields from the _User class:

- agentID
- caps
- id
- name
- presences
- states

Copyright © 1988-2007, Oracle. All rights reserved.

- statistics
- statistics1
- statistics2

See Chapter 13, “Understanding JSMCAPI Classes,” _User Class Constructor, page 230.

Buddy Class Callback Event Methods

The following are the Buddy class callback event methods.

onStat1

Syntax

onStat1(*event*)

Description

Fires when statistics1 is received.

onStat2

Syntax

onStat2(*event*)

Description

Fires when statistics2 is received.

onState

Syntax

onState(*event*)

Description

Fires when the state event is received.

Call Class Constructor

The Call class extends the Task class.

See Chapter 13, “Understanding JSMCAPI Classes,” Task Class Constructor, page 373.

The following is the Call class constructor.

 Copyright © 1988-2007, Oracle. All rights reserved.

Call

Syntax

Call(*strCall*)

Description

The Call object describes the call task information associated with the line.

Parameters

Parameter	Description
strCall	Enter the call to use with this object.

Returns

A Call object.

Call Class Fields

The Call class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority
- type
- urlAbs
- urlRel

See Chapter 13, "Understanding JSMCAPI Classes," Task Class Fields, page 374.

The following are additional Call class fields.

line

Description

The line object that is associated with this call.

Type: line object.

Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Line Class Constructor, page 325

statistics

Description

Call statistics object associated with the call.

Type: CallStatistics object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” CallStatistics Class Constructor, page 250

CallData Class Constructor

The following is the CallData class constructor.

CallData

Syntax

```
CallData()
```

Description

The CallData object describes the key-value pairs of the call data with the call object.

Parameters

None.

Returns

A CallData object.

CallData Class Fields

The following are the CallData class fields:

ani

Description

The ANI caller id.

Type: string.

 Copyright © 1988-2007, Oracle. All rights reserved.

callId

Description

The call ID.

Type: string.

callType

Description

The call type.

Type: string.

data

Description

The key value pairs of the call data.

Type: string.

dnis

Description

The DNIS callee ID.

Type: string.

CallData Class Method

The following is the CallData class method.

addKeyValue

Syntax

```
addKeyValue(key), value
```

Description

Add key value to the CallData object.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
key	The key.
value	The value.

Returns

None.

CallStatistics Class Constructor

The following is the CallStatistics class constructor.

CallStatistics

Syntax

```
CallStatistics()
```

Description

The call statistics information.

Parameters

None.

Returns

A CallStatistics object.

CallStatistics Class Fields

The following are the CallStatistics class fields.

data

Description

An associative array of key-value pairs that includes all statistics.

Type: associative array.

Copyright © 1988-2007, Oracle. All rights reserved.

holdTime

Description

Duration that the call is on hold.

Type: string.

queueTime

Description

Duration that this call has been in the queue.

Type: string.

talkTime

Description

Duration that the call is established.

Type: string.

Chat Class Constructor

The Chat class extends the Task class.

See Chapter 13, "Understanding JSMCAPI Classes," Task Class Constructor, page 373.

The following is the Chat class constructor.

Chat

Syntax

Chat(*event, address*)

Description

The Chat object describes the task information for tasks associated with the ChatAddress.

Parameters

Parameter	Description
event	The onNotify event.
address	The chat address.

Returns

A Chat object.

Copyright © 1988-2007, Oracle. All rights reserved.

Chat Class Fields

The Chat class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority
- type

  The task type for chat that is Task.TYPE_CHAT.
- urlAbs
- urlRel

See Chapter 13, “Understanding JSMCAPI Classes,” Task Class Fields, page 374.

The following are the additional Chat class fields.

address

Description

The address containing the agent-to-agent chat.

Type: A2AChatAddress object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” A2AChatAddress Class Constructor, page 237

agentId

Description

The agent’s ID.

Type: string.

appData

Description

The application data that is provided to the client with the notification event.

Type: AppData object.

 Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 13, "Understanding JSMCAPI Classes," AppData Class Constructor, page 242

chatconnection

Description

The chat connection object associated with this call.

Type: line object

chatType

Description

Type of chat, for example consulting chat or answering chat

Type: string

customerName

Description

The customer username.

Type: string.

groupId

Description

The group ID.

Type: string.

question

Description

The question for this chat.

Type: string.

subject

Description

The subject of the chat.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

statistics

Description

The chat statistics object that associated with this chat.

Type: string.

userData

Description

The user data associated with the chat.

Type: object.

Chat Class Method

The Chat class extends the Task class.

See Chapter 13, "Understanding JSMCAPI Classes," Task Class Constructor, page 373.

The following is the Chat class method.

gettpUrl

Syntax

gettpUrl(*defaultUrl*)

Description

Returns the URL for the given task for a third-party routing server.

Parameters

Parameter	Description
defaultUrl	If not null, this value will override the generated base URL.

Returns

Returns the URL associated with this task.

getUrl

Syntax

getUrl(*defaultUrl*)

Description

Returns the URL for the given task.

 Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
defaultUrl	The default URL for this chat. If not null this value will override the generated base URL.

Returns

Returns the URL for the chat.

ChatAddress Class Constructor

The ChatAddress class extends the _UQAddress class.

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Class Constructor, page 227.

The following is the ChatAddress class constructor.

ChatAddress

Syntax

```
ChatAddress()
```

Description

Handles the creation of customer chat tasks.

Parameters

None.

Returns

A ChatAddress object.

ChatAddress Class Fields

The ChatAddress class inherits the following fields from the _Address class:

- id

  The address's ID is equal to Task.TYPE_CHAT.

- caps

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Fields, page 226.

The ChatAddress class inherits the following field from the _UQAddress class:

tasks

Copyright © 1988-2007, Oracle. All rights reserved.

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Class Fields, page 227.

The following are ChatAddress class fields.

chatconnections

Description

List of ChatConnection objects that are associated with chat address.

Type: chatconnection object.

ChatAddress Class Methods

The ChatAddress class inherits the following methods from the _UQAddress class:

- acceptTask
- dequeueTask

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Methods, page 227.

The following are ChatAddress class methods.

chat

Syntax

```
chat(agentId,buddyId,reason  )
```

Description

Sends a request to initiate a new chat.

Parameters

Parameter	Description
agentId	The source agent Id.
buddyId	The Id of the buddy.
reason	The reason code for chat request.

Returns

None.

getChatconnectionByConnectionId

Syntax

```
getChatconnectionByConnectionId(connectionId)
```

 Copyright © 1988-2007, Oracle. All rights reserved.

Description

Get the chatconnection object with the chatconnection Id.

Parameters

Parameter	Description
connectionId	The connection id associated with a chat.

Returns

Line object, null if there is no connection with that chatconnection id.

getChatconnectionindexByConnectionId

Syntax

```
getChatconnectionindexByConnectionId
(connectionId)
```

Description

Get the chatconnection index with the chatconnection Id.

Parameters

Parameter	Description
connectionId	The connection id associated with a chat.

Returns

index, −1 if there is no connection with that chatconnection id.

getFreeChatconnection

Syntax

```
getFreeChatconnection()
```

Description

Get the free chatconnection object. A connection is free if there is no activity on this line.

Returns

Returns chatconnection object, null if there is no free line.

getFreeChatconnectionIndex

Syntax

```
getFreeChatconnectionIndex()
```

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Get the free chatconnection index. A connection is free if there is no activity on this line.

Returns

Returns chatconnection index, null if there is no free line.

ChatAddress Callback Event Methods

The ChatAddress class inherits the onError callback event method from the _Address class.

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Callback Event Method, page 226.

The ChatAddress class inherits the following callback event methods from the _UQAddress class:

- onAccepted
- onAcceptingTask
- onDequeueingTask
- onNotify
- onTaskAdded
- onTaskRemoved
- onUnassigned

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Class Fields, page 227.

The following are ChatAddress callback event methods:

onCapabilitiesChanged

Syntax

onCapabilitiesChanged*event*

Description

Fires when ChatAddress capabilities change

ChatConnection Class Constructor

The following is the chatconnection class constructor.

ChatConnection

Syntax

ChatConnection()

 Copyright © 1988-2007, Oracle. All rights reserved.

Description

Chat connection object.

Parameters

None

Returns

Returns a chatconnection object.

ChatConnection Class Fields

The following are chatconnection class fields:

caps

Description

The capabilities of the ChatConnection.

chat

Description

Chat task associated with the connection.

Type: string

connectionId

Description

The connection Id.

Type: string

id

Description

The constant representing ChatConnection Id.

Type: string

state

Description

The connection state.

Copyright © 1988-2007, Oracle. All rights reserved.

Type: string

Value	Description
ST_DIALING	Dialing state.
ST_DROPPED	Dropped state.
ST_IDLE	Idle state.
ST_INCOMING	Incoming state.
ST_TALKING	Talking state.

ChatConnection Class Methods

The following are chatconnection class methods:

answer

Syntax

answer(*agentId*,*reason*)

Description

Answer a chat request.

Parameters

Parameter	Description
agent Id	The Id of the agent answering the call.
reason	The reason code for the answer request.

Returns

None

conference

Syntax

conference(*agentId*,*reason*)

Description

Conference a chat with another agent.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
agent Id	The Id of the agent invited to the conference.
reason	The reason code for the conference request.

Returns

None.

forward

Syntax

forward(*fromagentId*,*toagentId*, *qid*, *userdata*, *chatdata*, *reason*)

Description

Forward chat to another agent or another queue.

Parameters

Parameter	Description
fromagentid	The agent id from whom task is forwarded.
toagentid	The target agentid to whom task is forwarded.
qid	The target queue to forward the task.
userdata	The user data.
chatdata	The chat data.
Reason	The reason code to forward the task.

Returns

None.

gethistory

Syntax

gethistory(*agentId*,*noofLines*, *reason*)

Description

Request chat history by specifying number of lines of chat history needed.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
agent Id	The source agent id.
noofLines	Number of lines of history required.
reason	The reason code for the history request.

Returns

None.

getUrl

Syntax

getUrl(*defaultURL*)

Description

Get the URL for screen popup.

Parameters

Parameter	Description
defaultUrl	The default popup URL.

Returns

Returns a URL string.

message

Syntax

message(*agentId*,*message*, *reason*)

Description

Send a chat message while chatting.

Parameters

Parameter	Description
agent Id	The source agent id.
message	The message to send to other party.
reason	The reason code for the message request.

Copyright © 1988-2007, Oracle. All rights reserved.

Returns

None.

pushURL

Syntax

pushURL(*URL*)

Description

Request to push a URL to the customer.

Parameters

Parameter	Description
URL	URL to push.

Returns

None.

reject

Syntax

reject(*agentId*,*reason*)

Description

Reject a chat.

Parameters

Parameter	Description
agent Id	The source agent id.
reason	The reason code for reject.

Returns

None.

release

Syntax

release(*agentId*,*reason*)

Description

Release a chat.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
agent Id	The source agent id.
reason	The reason code for release request.

Returns

None.

typing

Syntax

typing(*agentId*,*reason*)

Description

Indicates typing in a chat conversation by one party.

Parameters

Parameter	Description
agent Id	The source agent id.
reason	The reason code for typing.

Returns

None.

ChatConnection Class Callback Event Methods

The following are ChatConnection class callback event methods.

onAccepted

Syntax

onAccepted(*event*)

Description

Fires when chat is accepted by a agent.

onAnswering

Syntax

onAnswering(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when answering the chat.

onCapabilitiesChanged

Syntax

onCapabilitiesChanged(*event*)

Description

Fires when chat capabilities change.

onChatdataChanged

Syntax

onChatdataChanged(*event*)

Description

Fires when there is a change in chat data.

onConferencing

Syntax

onConferencing(*event*)

Description

Fires when conferencing the chat.

onDialing

Syntax

onDialing(*event*)

Description

Fires when chat is in the process of connecting.

onDropped

Syntax

onDropped(*event*)

Description

Fires when chat is dropped.

Copyright © 1988-2007, Oracle. All rights reserved.

onError

Syntax

onError(*event*)

Description

Fires when a there is a chat connection error.

onForwarded

Syntax

onForwarded(*event*)

Description

Fires when chat is forwarded.

onForwardError

Syntax

onForwardError(*event*)

Description

Fires when there is error in forwarding the task.

onForwarding

Syntax

onForwarding(*event*)

Description

Fires when forwarding the chat.

onHistory

Syntax

onHistory(*event*)

Description

Fires when agent receives chat history.

onIncomingChat

Syntax

onIncomingChat(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when ChatConnection state is changed to INCOMING.

onMessage

Syntax

onMessage(*event*)

Description

Fires when there is incoming message.

onPartyAdded

Syntax

onPartyAdded(*event*)

Description

Fires when a new chat party is added.

onPartyChanged

Syntax

onPartyChanged(*event*)

Description

Fires when the chat party changes.

onPartyRemoved

Syntax

onPartyRemoved(*event*)

Description

Fires when a chat party is removed.

onProperties

Syntax

onProperties(*event*)

Description

Fires when a agent receives chat properties.

Copyright © 1988-2007, Oracle. All rights reserved.

onPushURL

Syntax

onPushURL(*event*)

Description

Fires when a routing system push URL.

onRejected

Syntax

onRejected(*event*)

Description

Fires when a chat is rejected.

onReleased

Syntax

onReleased(*event*)

Description

Fires when a chat is released.

onReleasing

Syntax

onReleasing(*event*)

Description

Fires when conferencing the chat.

onRevoked

Syntax

onRevoked(*event*)

Description

Fires when a chat is revoked.

onTalking

Syntax

onTalking(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when an agent/customer is in the state of talking.

onTyping

Syntax

onTyping(*event*)

Description

Fires when typing in a chat conversation.

onUserdataChanged

Syntax

onUserdataChanged(*event*)

Description

Fires when there is a change in user data.

ChatConnectionCaps Class Constructor

The following is the chatconnectioncaps class constructor.

ChatConnectionCaps

Syntax

ChatConnectionCaps(*strCaps*)

Description

Describes the ChatConnection capabilities.

Parameters

Parameter	Description
strCaps	Chat connection capability.

Returns

Returns a chatconnection object.

Copyright © 1988-2007, Oracle. All rights reserved.

ChatConnectionCaps Class Fields

The following are the ChatConnectionCaps class fields.

canAnswer

Description

Answer capability.

Type: boolean

canConference

Description

Conference capability.

Type: boolean

canConferenceSingle

Description

Conference single capability.

Type: boolean

canForward

Description

Forward capability.

Type: boolean

canIndicateTyping

Description

Indicate Typing capability.

Type: boolean

canPushURL

Description

PushURL capability.

Type: boolean

Copyright © 1988-2007, Oracle. All rights reserved.

canReject

Description

Reject capability.

Type: boolean

canSendMessage

Description

SendMessage capability.

Type: boolean

ChatData Class Constructor

The following is ChatData class constructor.

ChatData

Syntax

```
ChatData()
```

Description

The ChatData object describes the key-value pairs of the chat data with the chat object.

ChatData Class Fields

The following is ChatData class field.

data

Description

Key value pairs that includes data.

Type: object.

ChatData Class Methods

The following is ChatData class method.

Copyright © 1988-2007, Oracle. All rights reserved.

addKeyValue

Syntax

addKeyValue(*key*, *value*)

Description

Add key value to the chatdata object.

Parameters

Parameter	Description
key	The variable name.
value	The value of the variable.

Email Class Constructor

The Email class extends the Task class.

See Chapter 13, "Understanding JSMCAPI Classes," Task Class Constructor, page 373.

The following is the email class constructor.

Email

Syntax

Email(*event*)

Description

The Email object describes the task information for tasks associated with the EmailAddress.

Parameters

Parameter	Description
event	The notification event.

Returns

Returns an Email object.

Email Class Fields

The Chat class inherits the following fields from Task class:

- caseid

 Copyright © 1988-2007, Oracle. All rights reserved.

- cost
- customerid
- group
- id
- onStat
- priority
- type

  The task type is equal to Task.TYPE_EMAIL.
- urlAbs
- urlRel

See Chapter 13, “Understanding JSMCAPI Classes,” Task Class Fields, page 374.

The following are the additional Email class fields.

address

Description

The address containing the A2AChat.

agentId

Description

The agent’s ID.

appData

Description

The application data that is provided to the client with the notification event.

Type: AppData object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Task Class Fields, page 374

customerName

Description

The customer username.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

emailconnection

Description

The emailconnection object that associated with this email.

Type: line object.

emailId

Description

The unique email id generated from an Enqueue. It identifies the email stored in the database ie. it serves as a key to the data stored in the database.

Type: string.

groupId

Description

The group id.

Type: string.

question

Description

The question for this email.

Type: string.

statistics

Description

The statistics object that is associated with this email.

Type: object.

subject

Description

The subject of this email.

Type: string.

userData

Description

The user data that is associated with this email.

Copyright © 1988-2007, Oracle. All rights reserved.

Type: object.

Email Class Method

The following are the Email class methods:

gettpUrl

Syntax

gettpUrl(*defaultUrl*)

Description

Returns the URL for the given task for third-party routing.

Parameters

Parameter	Description
defaultUrl	If not null, this value will override the generated base URL.

Returns

The URL associated with this task.

getUrl

Syntax

getURL(*defaultUrl*)

Description

Returns the URL for the given task.

Parameters

Parameter	Description
defaultUrl	If not null, this value will override the generated base URL.

Returns

The URL associated with this task.

EmailAddress Class Constructor

The EmailAddress class extends the _UQAddress class.

Copyright © 1988-2007, Oracle. All rights reserved.

See Chapter 13, “Understanding JSMCAPI Classes,” _UQAddress Class Constructor, page 227.

The following is the EmailAddress class constructor.

EmailAddress

Syntax

```
EmailAddress()
```

Description

Handles the creation of email tasks.

Parameters

None.

Returns

An EmailAddress object.

EmailAddress Class Fields

The EmailAddress class inherits the following fields from the _Address class:

- id

  The address’s ID is equal to Task.TYPE_EMAIL.
- caps

See Chapter 13, “Understanding JSMCAPI Classes,” _Address Class Fields, page 226.

The EmailAddress class inherits the tasks field from the _UQAddress class.

See Chapter 13, “Understanding JSMCAPI Classes,” _UQAddress Class Fields, page 227.

The following are EmailAddress class fields.

agent

Description

The agent id of this email address.

emailconnections

Description

Describes the EmailConnection object.

 Copyright © 1988-2007, Oracle. All rights reserved.

EmailAddress Class Methods

The EmailAddress class inherits the following fields from the _UQAddress class:

- acceptTask
- dequeueTask

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Methods, page 227.

getEmailconnectionByConnectionId

Syntax

getEmailconnectionByConnectionId(*connectionId*)

Description

Get the emailconnection object with the emailconnection id.

Parameters

Parameter	Description
connectionId	The connection id associated with a email.

Returns

Returns line object, null if there is no connection with that emailconnection id.

getEmailconnectionindexByConnectionId

Syntax

getEmailconnectionindexByConnectionId(*connectionId*)

Description

Get the emailconnection index with the emailconnection id.

Parameters

Parameter	Description
connectionId	The connection id associated with a email.

Returns

Returns index, -1 if there is no connection with that emailconnection id.

getFreeEmailconnection

Syntax

getFreeEmailconnection()

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Get the free email connection object. A connection is free if there is no activity on this line.

Returns

Returns emailconnection object, null if there is no free line.

EmailAddress Callback Event Methods

The EmailAddress class inherits the onError callback event method from the _Address class.

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Callback Event Method, page 226.

The EmailAddress class inherits the following callback event methods from the _UQAddress class:

- onAccepted
- onAcceptingTask
- onDequeueingTask
- onNotify
- onTaskAdded
- onTaskRemoved
- onUnassigned

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Class Callback Event Methods, page 228.

EmailConnection Class Constructor

The following is EmailConnection class constructor.

EmailConnection

Syntax

```
EmailConnection()
```

Description

Handles email connection.

EmailConnection Class Fields

The following are EmailConnection class fields.

 Copyright © 1988-2007, Oracle. All rights reserved.

caps

Description

The capabilities of the EmailConnection.

connectionId

Description

The email connection Id.

Type: string

email

Description

Email task id associated with this email connection.

Type: string with following constants:

id

Description

The id of the EmailConnection.

Type: string

state

Description

The connection state.

Type: string

Value	Description
ST_DIALING	Dialing state.
ST_DROPPED	Dropped state.
ST_IDLE	Idle state.
ST_INCOMING	Incoming state.
ST_TALKING	Talking state.

Copyright © 1988-2007, Oracle. All rights reserved.

EmailConnection Class Methods

The following are the EmailConnection class methods:

abandon

Syntax

abandon(*reason*)

Description

Abandon email task without completing the task .

Parameters

Parameter	Description
Reason	The reason code to abandon.

Returns

None.

answer

Syntax

answer(*reason*)

Description

Answer/Accept the email assignment.

Parameters

Parameter	Description
Reason	The reason code to answer.

Returns

None.

complete

Syntax

complete(*reason*)

Description

Send task completion notification.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
Reason	The reason code to complete.

Returns

None.

forward

Syntax

forward(*fromagentId*,*toagentId*, *qid*, *userdata*, *emaildata*, *reason*)

Description

Forward email to another agent or to another queue.

Parameters

Parameter	Description
fromagentid	The agent id from whom task is forwarded.
toagentid	The target agent id to whom task is forwarded.
qid	The target queue to forward the task.
userdata	The user data.
emaildata	The email data.
Reason	The reason code to forward the task.

Returns

None.

reject

Syntax

reject(*reason*)

Description

Reject email assignment.

Parameters

Parameter	Description
Reason	The reason code to reject.

Copyright © 1988-2007, Oracle. All rights reserved.

Returns

None.

withdraw_RES

Syntax

withdraw_RES
(*reason*)

Description

Send response for the request of (Withdraw of Email) by routing server. This is the response from the agent to the routing system.

Parameters

Parameter	Description
Reason	The reason code to withdraw.

Returns

None.

EmailConnection Class Callback Event Methods

The following are EmailConnection class callback event methods:

onAnswering

Syntax

onAnswering(*event*)

Description

Fires when email task is getting answered.

onCapabilitiesChanged

Syntax

onCapabilitiesChanged(*event*)

Description

Fires when there is a change in connection capabilities.

Copyright © 1988-2007, Oracle. All rights reserved.

onCompleted

Syntax

onCompleted(*event*)

Description

Fires when EmailConnection state is changed to COMPLETED.

onDropped

Syntax

onDropped(*event*)

Description

Fires when EmailConnection state is changed to DROPPED.

onEmaildataChanged

Syntax

onEmaildataChanged(*event*)

Description

Fires when there is a change in email data.

onError

Syntax

onError(*event*)

Description

Fires when a there is a email connection error.

onForwarded

Syntax

onForwarded(*event*)

Description

Fires when email is forwarded to another queue or to another agent.

onForwardError

Syntax

onForwardError(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when there is error in forwarding the task.

onForwarding

Syntax

onForwarding(*event*)

Description

Fires when forwarding the email.

onIncoming

Syntax

onIncoming(*event*)

Description

Fires when EmailConnection state is changed to INCOMING.

onProcessing

Syntax

onProcessing(*event*)

Description

Fires when EmailConnection state is changed to PROCESSING.

onRejected

Syntax

onRejected(*event*)

Description

Fires when email task is rejected by agent and the routing system acknowledges the same.

onRevoked

Syntax

onRevoked(*event*)

Description

Fires when email is revoked by routing system before agent accepts that.

Copyright © 1988-2007, Oracle. All rights reserved.

onUserdataChanged

Syntax

onUserdataChanged(*event*)

Description

Fires when there is a change in user data.

onWithdraw_REQ

Syntax

onWithdraw_REQ(*event*)

Description

Fires when there is a request for email withdraw from routing server. This is a request from routing server to the agent.

EmailConnectionCaps Class Constructor

The following is EmailConnectionCaps class constructor:

EmailConnectionCaps

Syntax

EmailConnectionCaps(*strCaps*)

Description

Describes the EmailConnection capabilities.

Parameters

Parameter	Description
strCaps	The capability of the connection

Returns

Returns EmailConnection object.

EmailConnectionCaps Class Fields

The following are the EmailConnectionCaps class fields.

Copyright © 1988-2007, Oracle. All rights reserved.

canAnswer

Description

Answer/Accept an email capability.

Type: boolean

canComplete

Description

Capability to complete an assigned email.

Type: boolean

canForward

Description

Capability to forward an email.

Type: boolean

canReject

Description

Capability to reject an email.

Type: boolean

EmailData Class Constructor

The following is EmailConnectionCaps class constructor.

EmailData

Syntax

```
EmailData()
```

Description

The EmailData object describes the key-value pairs of the email data with the Email object.

Returns

Returns EmailData object.

Copyright © 1988-2007, Oracle. All rights reserved.

EmailData Class Fields

The following is EmailData class field.

data

Description

Key value pairs that includes email data.

Type: object.

EmailData Class Methods

The following is EmailData class method.

addKeyValue

Syntax

addKeyValue(*key, value*)

Description

Add key value to the emaildata object.

Parameters

Parameter	Description
key	The variable name.
value	The value of the variable.

Extension Class Constructor

The Extension class extends the _Address class.

See Chapter 13, “Understanding JSMCAPI Classes,” _Address Class Constructor, page 225.

The following is the Extension class constructor.

Extension

Syntax

Extension(*numOfLines*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

The Extension object describes the CTI address.

Parameters

Parameter	Description
numOfLines	Enter the number of lines for this extension.

Returns

An Extension object.

Extension Class Fields

The Extension class inherits the following fields from the _Address class:

- caps
- id

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Fields, page 226.

The following are the Extension class fields.

forwardMode

Description

The forward mode of the extension.

Type: forwardMode object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," ForwardMode Class Constructor, page 296

isDnd

Description

Flag for Do Not Disturb.

Type: boolean.

lines

Description

List of line objects associated with this extension.

Type: list.

 Copyright © 1988-2007, Oracle. All rights reserved.

numOfLines

Description

The number of lines for this extension.

Type: number.

Extension Class Methods

The Extension class inherits the onError method from the Address class.

The following are the Extension class methods.

cancelDnd

Syntax

cancelDnd(*reason*)

Description

Cancel the DND (do not disturb).

Parameters

Parameter	Description
reason	The reason for DND cancellation.

Returns

None.

cancelForwardSet

Syntax

cancelForwardSet(*reason*)

Description

Cancel the forward.

Parameters

Parameter	Description
reason	The reason for cancelling the forward.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

forwardSet

Syntax

forwardSet(*number, mode, reason*)

Description

Forward the call to another number/extension.

Parameters

Parameter	Description
mode	The forward mode.
number	The number to which all calls will be forwarded.
reason	The reason to forward.

Returns

None.

getDialingLine

Syntax

getDialingLine()

Description

Get the dialing line object.

Parameters

None

Returns

Returns a line object. Null if there is no dialing line.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Line Class Constructor, page 325

getFreeLine

Syntax

getFreeLine()

Description

Get the free line object. A line is free if there is no activity on this line.

 Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

None

Returns

Returns a line object. It returns null if there is no free line.

getLineByConnectionId

Syntax

getLineByConnectionId(*connectionID*)

Description

Get the line object with the call.

Parameters

Parameter	Description
connectionId	The connection ID associated with a call.

Returns

Returns a Line object. Returns null if there is no line with that call.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Line Class Constructor, page 325

getOffHookLine

Syntax

getOffHookLine()

Description

Get the offhook line object.

Parameters

None.

Returns

Returns a Line object. Returns null if there is no offhook line.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Line Class Constructor, page 325

Copyright © 1988-2007, Oracle. All rights reserved.

setDnd

Syntax

setDnd(*reason*)

Description

Fires when setting DND (do not disturb).

Parameters

Parameter	Description
reason	The reason to set the DND.

Returns

None.

Extension Class Callback Event Methods

The following are the Extension class callback event methods.

onCancelingDnd

Syntax

onCancelingDnd(*event*)

Description

Fires when canceling DND (do not disturb).

onCancelingForward

Syntax

onCancelingForward(*event*)

Description

Fires when canceling forward.

onDnd

Syntax

onDnd(*event*)

Description

Fires when DND (Do Not Disturb) is requested and processed.

Copyright © 1988-2007, Oracle. All rights reserved.

onDndCanceled

Syntax

onDndCanceled(*event*)

Description

Fires when DND is cancelled.

Parameters

Parameter	Description
event	The event object.

Returns

None.

onForwardCanceled

Syntax

onForwardCanceled(*event*)

Description

Fires when forward is canceled.

onForwarded

Syntax

onForwarded(*event*)

Description

Fires when call is forwarded.

onForwarding

Syntax

onForwarding(*event*)

Description

Fires when forwarding the call.

onSettingDnd

Syntax

onSettingDnd(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when setting DND (do not disturb).

ExtensionCaps Class Constructor

The following is the Extension class constructor.

ExtensionCaps

Syntax

ExtensionCaps(*strCaps*)

Description

Describes the extension's capabilities.

Parameters

Parameter	Description
strCaps	A string comprising the extension's capabilities.

Returns

An ExtensionCaps object.

ExtensionCaps Class Fields

The following are the Extension class fields.

canCancelDnd

Description

This extension can cancel DND (do not disturb).

Type: boolean.

canDial

Description

This extension can dial out.

Type: boolean.

Copyright © 1988-2007, Oracle. All rights reserved.

canFwdBusy

Description

This extension can forward calls if busy.

Type: boolean.

canFwdBusyNoAnswer

Description

This extension can forward calls if busy/no answer.

Type: boolean.

canFwdCancelForward

Description

This extension can cancel forward.

Type: boolean.

canFwdDefault

Description

This extension can forward.

Type: boolean.

canFwdNoAnswer

Description

This extension can forward if no answer.

Type: boolean.

canFwdUnconditional

Description

This extension can forward unconditionally.

Type: boolean.

canRefreshState

Description

This extension can refresh state.

Type: boolean.

Copyright © 1988-2007, Oracle. All rights reserved.

canSetDnd

Description

This extension can set DND (do not disturb).

Type: boolean.

ForwardMode Class Constructor

The following is the ForwardMode class constructor.

ForwardMode

Syntax

```
ForwardMode()
```

Description

Describes various forward modes that can be used while setting the forwarding mode for an Address.

Parameters

None.

Returns

A ForwardMode object.

ForwardMode Class Field

The following is the ForwardMode class field.

mode

Description

The current forwarding mode.

Type: string of the following constants.

Value	Description
BUSY	The BUSY forwarding mode.
BUSYNOANSWER	The BUSYNOANSWER forwarding mode.
DEFAULT	The DEFAULT forwarding mode.

 Copyright © 1988-2007, Oracle. All rights reserved.

Value	Description
NOANSWER	The NOANSWER forwarding mode.
NONE	No forwarding mode.
UNCONDITIONAL	The UNCONDITIONAL forwarding mode.

GenericAddress Class Constructor

The GenericAddress class extends the _UQAddress class.

The following is the GenericAddress class constructor.

GenericAddress

Syntax

```
GenericAddress()
```

Description

Handles the creation of generic tasks.

Parameters

None.

Returns

A GenericAddress object.

GenericAddress Class Fields

The GenericAddress class inherits the following fields from the _Address class.

- caps
- id

  The address's ID is equal to Task.TYPE_GENERIC.

See Chapter 13, "Understanding JSMCAPI Classes," _Address Class Fields, page 226.

The GenericAddress class inherits the tasks field from the _UQAddress class.

See Chapter 13, "Understanding JSMCAPI Classes," _UQAddress Class Fields, page 227.

The following are GenericAddress class fields:

Copyright © 1988-2007, Oracle. All rights reserved.

agent

Description

The agent id of this generic address owner.

Type: object

genericconnections

Description

List of GenericConnections.

Type: genericconnectionobject

GenericAddress Class Methods

The GenericAddress class inherits the following methods from the _UQAddress class:

- acceptTask
- dequeueTask

See Chapter 13, “Understanding JSMCAPI Classes,” _UQAddress Methods, page 227.

The following are GenericAddress own class methods.

getFreeGenericconnection

Syntax

```
getFreeGenericconnectio()
```

Description

Get the free generic connection object. A connection is free if there is no activity on this line.

Parameters

None.

Returns

Returns genericconnection object, null if there is no free line.

getGenericconnectionByConnectionId

Syntax

```
getFreeGenericconnectio(connectionId)
```

Description

Get the genericconnection object with the genericconnection id.

 Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
connectionId	The connection id associated with a generic.

Returns

Returns line object, null if there is no connection with that genericconnection id.

getGenericconnectionindexByConnectionId

Syntax

getGenericconnectionindexByConnectionId(*connectionId*)

Description

Get the genericconnection index with the genericconnection id.

Parameters

Parameter	Description
connectionId	The connection id associated with a generic.

Returns

Returns index, -1 if there is no connection with that genericconnection id.

GenericAddress Class Callback Event Methods

The GenericAddress class inherits the onError callback event method from the _Address class.

See Chapter 13, “Understanding JSMCAPI Classes,” _Address Class Callback Event Method, page 226.

The GenericAddress class inherits the following callback event methods from the _UQAddress class:

- onAcceptingTask
- onDequeueingTask
- onTaskAdded
- onTaskRemoved
- onNotify
- onAccepted
- onUnassigned

See Chapter 13, “Understanding JSMCAPI Classes,” _UQAddress Class Callback Event Methods, page 228.

Copyright © 1988-2007, Oracle. All rights reserved.

GenericConnection Class Constructor

The following is the GenericConnection class constructor.

GenericConnection

Syntax

```
GenericConnection()
```

Description

Handles Generic connection.

Parameters

None.

Returns

A genericconnection object.

GenericConnection Class Fields

The following are GenericConnection class fields:

caps

Description

The capabilities of the GenericConnection.

connectionId

Description

The generic connection Id.

Type: string

generic

Description

The generic task id associated with this generic connection.

Type: string

Copyright © 1988-2007, Oracle. All rights reserved.

id

Description

The id of the GenericConnection.

Type: string

state

Description

The connection state.

Type: string with following constants:

Value	Description
ST_DIALING	Dialing state.
ST_DROPPED	Dropped state.
ST_IDLE	Idle state.
ST_INCOMING	Incoming state.
ST_TALKING	Talking state.

GenericConnection Class Methods

The following are the GenericConnection class methods:

abandon

Syntax

abandon(*reason*)

Description

Abandon generic task.

Parameters

Parameter	Description
Reason	The reason code to abandon.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

answer

Syntax

answer(*reason*)

Description

Answer/Accept the generic task assignment.

Parameters

Parameter	Description
Reason	The reason code to answer.

Returns

None.

complete

Syntax

complete(*reason*)

Description

Send task completion notification.

Parameters

Parameter	Description
Reason	The reason code to complete.

Returns

None.

forward

Syntax

forward(*fromagentId*,*toagentId*, *qid*, *userdata*, *genericdata*, *reason*)

Description

Forward generic task to another Agent or to another queue.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
fromagentid	The agent id from whom task is forwarded.
toagentid	The target agent id to whom task is forwarded.
qid	The target queue to forward the task.
userdata	The user data.
emaildata	The email data.
Reason	The reason code to forward the task.

Returns

None.

reject

Syntax

reject(*reason*)

Description

Reject generic task assignment.

Parameters

Parameter	Description
Reason	The reason code to reject.

Returns

None.

withdraw_RES

Syntax

withdraw_RES
(*reason*)

Description

Send response for the request of (Withdraw of Generic task) by routing server. This is the response from the agent to the routing system.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
Reason	The reason code to withdraw.

Returns

None.

GenericConnection Class Callback Event Methods

The following are ChatConnection class callback event methods:

onCapabilitiesChanged

Syntax

onCapabilitiesChanged(*event*)

Description

Fires when there is a change in GenericData.

onCompleted

Syntax

onCompleted(*event*)

Description

Fires when GenericConnection state is changed to COMPLETED.

onDropped

Syntax

onDropped(*event*)

Description

Fires when GenericConnection state is changed to DROPPED.

onError

Syntax

onError(*event*)

Description

Fires when a there is a generic connection error.

 Copyright © 1988-2007, Oracle. All rights reserved.

onForwarded

Syntax

onForwarded(*event*)

Description

Fires when generic task is forwarded.

onForwardError

Syntax

onForwardError(*event*)

Description

Fires when there is error in forwarding the generic task.

onForwarding

Syntax

onForwarding(*event*)

Description

Fires when forwarding the generic task.

onGenericdataChanged

Syntax

onGenericdataChanged(*event*)

Description

Fires when there is a change in GenericData.

onIncoming

Syntax

onIncoming(*event*)

Description

Fires when GenericConnection state is changed to INCOMING.

onProcessing

Syntax

onProcessing(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when GenericConnection state is changed to PROCESSING.

onRejected

Syntax

onRejected(*event*)

Description

Fires when generic task is rejected by agent and the routing system acknowledges the same.

onRevoked

Syntax

onRevoked(*event*)

Description

Fires when generic task is revoked by routing system before agent accepts that.

onUserdataChanged

Syntax

onUserdataChanged(*event*)

Description

Fires when there is a change in user data.

onWithdraw_REQ

Syntax

onWithdraw_REQ(*event*)

Description

Fires when there is a request for generic task withdraw from routing server. This is a request from routing server to the agent.

GenericConnectionCaps Class Constructor

The following is GenericConnectionCaps class constructor.

GenericConnectionCaps

Syntax

GenericConnectionCaps(*strCaps*)

 Copyright © 1988-2007, Oracle. All rights reserved.

Description

Describes the GenericConnection capabilities.

Parameters

Parameter	Description
strCaps	The capability of the connection.

Returns

Returns GenericConnection object.

GenericConnectionCaps Class Fields

The following are the GenericConnectionCaps class fields

canAnswer

Description

Answer/Accept a generic task capability.

Type: boolean

canComplete

Description

Complete an assigned generic task capability.

Type: boolean

canForward

Description

Forward a generic task capability.

Type: boolean

canReject

Description

Reject a generic task capability.

Type: boolean

Copyright © 1988-2007, Oracle. All rights reserved.

GenericData Class Constructor

The following is GenericConnectionCaps class constructor:

GenericData

Syntax

```
GenericData()
```

Description

The GenericData object describes the key-value pairs of the generic data with the generic object

Returns

Returns GenericData object.

GenericData Class Fields

The following is GenericData class field:

data

Description

Key value pairs that includes data.

Type: object.

GenericData Class Methods

The following is GenericData class method:

addKeyValue

Syntax

```
addKeyValue(key, value )
```

Description

Add key value to the genericdata object.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
key	The variable name.
value	The value of the variable.

GenericTask Class Constructor

The GenericTask class extends the Task class.

The following is the GenericTask class constructor

GenericTask

Syntax

GenericTask(*event*)

Description

The GenericTask object describes the generic task information for tasks associated with the GenericAddress object.

Parameters

Parameter	Description
event	The triggering event.

Returns

A GenericTask object.

GenericTask Class Fields

The GenericTask class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority

Copyright © 1988-2007, Oracle. All rights reserved.

- type

  The task type for generic task that is Task.TYPE_GENERIC.
- urlAbs
- urlRel

See Chapter 13, “Understanding JSMCAPI Classes,” Task Class Fields, page 374.

The following are the additional GenericTask class fields.

address

Description

The address containing the A2AChat.

agentId

Description

The agent id to whom this task is assigned.

Returns a string.

appData

Description

The application data that is provided to the client with the notification event.

Type: AppData object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” AppData Class Constructor, page 242

customerName

Description

The customer username.

Type: string.

genericconnection

Description

The generic connection object that associated with this Generic task.

Type: line

 Copyright © 1988-2007, Oracle. All rights reserved.

genericId

Description

The unique generic id generated from an Enqueue. It identifies the data in the database associated with the generic task.

Type: string.

groupId

Description

The group id for which task is initiated.

Type: string.

question

Description

The question of this generic task which is raised by customer.

Type: string.

statistics

Description

Task statistics of the generic task.

Type: object.

subject

Description

The subject of the generic task.

Type: string.

userdata

Description

User data object that is associated with this generic task.

Type: object.

GenericTask Class Method

The following is the GenericTask class method.

Copyright © 1988-2007, Oracle. All rights reserved.

gettpUrl

Syntax

gettpUrl(*defaultUrl*)

Description

Returns the URL for the given task for the third-party routing.

Parameters

Parameter	Description
defaultUrl	If not null, this value will override the generated base URL.

Returns

Returns the URL for the given task.

getUrl

Syntax

getUrl(*defaultUrl*)

Description

Returns the URL for the given task.

Parameters

Parameter	Description
defaultUrl	If not null, this value will override the generated base URL.

Returns

Returns the URL for the given task.

GLOBALS Class Fields

The following are global fields, which may be accessed without instantiating an object. They are accessed as shown.

A2AChat.PS_JR

Description

The Journal Routing constant.

Copyright © 1988-2007, Oracle. All rights reserved.

Example

A2AChat.PS_JR = "ps_jr";

A2AChat.TYPE_ANSWER

Description

The constant representing an A2AChat type of answer.

This type is generated when a different user wants this user to answer an A2AChat.

Example

A2AChat.TYPE_ANSWER = "answer";

A2AChat.TYPE_CONSULT

Description

The constant representing an A2AChat type of consult.

This type is generated when this user wants to consult a different user.

Example

A2AChat.TYPE_CONSULT = "consult";

Server.TYPE_CTI

Description

The constant representing a CTI server.

Example

Server.TYPE_CTI = "CTI";

Server.TYPE_UQ

Description

The UQ server type.

Example

Server.TYPE_UQ = "UQ";

Task.TYPE_A2ACHAT

Description

The constant representing an A2AChat.

Copyright © 1988-2007, Oracle. All rights reserved.

Example

Task.TYPE_A2ACHAT = "A2ACHAT";

Task.TYPE_CHAT

Description

The constant representing a chat task.

Example

Task.TYPE_CHAT = "CHAT";

Task.TYPE_CTI

Description

The constant representing a CTI task.

Example

Task.TYPE_CTI = "CTI";

Task.TYPE_EMAIL

Description

The constant representing an email task.

Example

Task.TYPE_EMAIL = "EMAIL";

Task.TYPE_GENERIC

Description

The constant representing a generic task.

Example

Task.TYPE_GENERIC = "GENERIC";

GLOBALS Class Methods

The following are the GLOBALS class methods.

Copyright © 1988-2007, Oracle. All rights reserved.

initJSMCAPI

Syntax

initJSMCAPI()

Description

The initialization function of the JSMCAPI. This is the first function should be called by application.

Parameters

None.

Returns

None. Initializes JSMCAPI.

isValid

Syntax

isValid(*obj*)

Description

Returns true if an object is not undefined and not null. *Obj* can be any JavaScript object or literal. This method is a convenience method to check that an object is both not null and not undefined.

Parameters

Parameter	Description
obj	*Obj* can be any JavaScript object or literal.

Returns

Returns a boolean.

Returns true if an object is not undefined and not null.

MCFBroadcast

Syntax

MCFBroadcast(*cluster*, *queue*, *task*, *state*, *presence*, *message*, *securitylevel*, *importancelevel*, *senderid*, *NameValuePairString*)

Description

Broadcast a message to any queue, cluster, or only agents.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
cluster	Cluster Id to which we need to send broadcast message.
queue	Denotes the queue Id in the cluster.
task	Denotes the task, such as email, chat, voice, or generic.
state	Denotes the state, such as LoggedIn and NotLoggedIn.
presence	Denotes whether the agent is ready or not ready.
Message	Enter the message to broadcast.
security level	Security level defined by application developers.
importance level	Importance level defined by application developers.
sender Id	The sender's user id
namevaluepairs	Any extra data formed as name-value pair string.

Returns

None

Group Class Constructor

The following is the Group class constructor.

Group

Syntax

```
Group()
```

Description

The Group object describes the group information.

Parameters

None.

Returns

A Group object.

Copyright © 1988-2007, Oracle. All rights reserved.

Group Class Fields

The following are the Group class fields.

id

Description

The group ID.

Type: string.

name

Description

The group name.

Type: string.

registered

Description

True if the Group is registered on the server.

Type: boolean.

statistics

Description

The group statistics for CTI.

Type: GroupStatistics object

See Also

Chapter 13, “Understanding JSMCAPI Classes,” GroupStatistics1 Class Constructor, page 321

statistics1

Description

The group statistics for the queue server.

Type: GroupStatistics1 object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” GroupStatistics1 Class Constructor, page 321

Copyright © 1988-2007, Oracle. All rights reserved.

statistics2

Description

The group statistics for the queue server.

Type: GroupStatistics2 object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” GroupStatistics2 Class Constructor, page 324

Group Class Callback Event Methods

The following are the Group class callback event methods.

onStat

Syntax

onStat(*event*)

Description

Fires when there is new statistics going to this group.

onStat1

Syntax

onStat1(*event*)

Description

Fires when statistics1 is received.

onStat2

Syntax

onStat2(*event*)

Description

Fires when statistics2 is received.

onTaskAdded

Syntax

onTaskAdded(*event*)

 Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when a task added to this group.

onTaskRemoved

Syntax

onTaskRemoved(*event*)

Description

Fires when a task is removed from this group.

GroupStatistics Constructor

The following is the GroupStatistics class constructor.

GroupStatistics

Syntax

GroupStatistics()

Description

The group statistics information.

Parameters

None.

Returns

A GroupStatistics object.

GroupStatistics Fields

The following are the GroupStatistics class fields.

data

Description

Key value pairs that include all statistics.

Copyright © 1988-2007, Oracle. All rights reserved.

listOfTasksInTheQueueByTaskType

Description

Denotes the task type, task state and the time for which the task is in the system.

maxTaskCompletionTime

Description

Longest wait time for a task in the queue.

newestTask

Description

Time elapsed for the most recent task.

newestTaskCompletionTime

Description

Difference between queue time and the time when task is done.

numberOfAbandoned

Description

Number of tasks that are abandoned.

numberOfLoggedIn

Description

Number of agents that are logged in the queue.

numberOfQueued

Description

Number of tasks that are queued.

numUnassignedTasks

Description

Number of unassigned tasks.

Copyright © 1988-2007, Oracle. All rights reserved.

queuedWaitTime

Description

The average wait time, in seconds, of a queued task.

queueUpTime

Description

Time since the queue is available on the system (startup/boot time).

relativeQueueLoad

Description

Relative queue load.

timeElapsedOldestTask

Description

Time elapsed for the oldest task (difference between current time and en-queue time).

GroupStatistics1 Class Constructor

The following is the GroupStatistics1 class constructor.

GroupStatistics1

Syntax

`GroupStatistics1``(`*`data`*`)`

Description

UQ Group statistics sent on group refresh 1.

Parameters

Parameter	Description
data	Statistical data from the UQ server.

Returns

A GroupStatistics1 object.

Copyright © 1988-2007, Oracle. All rights reserved.

GroupStatistics1 Class Fields

The following are the GroupStatistics1 class fields.

mostRecentTaskDone

Description

The most recently done task in the group.

Type: string.

mostRecentTaskDoneData

Description

The data for the most recent task done.

Type: string.

mostRecentTaskEnqueued

Description

The most recently enqueued task in the group.

Type: string.

mostRecentTaskEnqueuedData

Description

The data for the most recently enqueued task.

Type: string.

numAgentsAvailable

Description

Number of agents available in the group.

Type: string.

numAgentsLoggedIn

Description

Number of agents logged in on the group.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

numEscalation

Description

Number of escalated tasks in the group.

Type: string.

numOverflow

Description

Number of overflowed tasks in the group.

Type: string.

numTaskAccepted

Description

Number of tasks accepted in the group.

Type: string.

numTaskDone

Description

Number of tasks done in the group.

Type: string.

numTaskQueued

Description

Number of tasks queued in the group.

Type: string.

reasonFlag

Description

The reason flag for this GroupStatistics event.

Type: string.

taskTotalTimeInSystem

Description

The total time in the system.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

timeSinceStart

Description

The time since start.

Type: string.

GroupStatistics2 Class Constructor

The following is the GroupStatistics2 class constructor.

GroupStatistics2

Syntax

GroupStatistics2(*data*)

Description

UQ Group statistics sent on group refresh 2.

Parameters

Parameter	Description
data	UQ server statistical data.

Returns

A GroupStatistics2 object.

GroupStatistics2 Class Fields

The following are the GroupStatistics2 class fields.

averageTaskDuration

Description

The average task duration in the group.

Type: string.

averageWaitTime

Description

The average wait time in the group.

 Copyright © 1988-2007, Oracle. All rights reserved.

Type: string.

oldestTask

Description

The oldest task in the group.

Type: string.

recentTask

Description

The most recent task in the group.

Type: string.

timeElapsedOldestTask

Description

The time elapsed for the oldest task in the group.

Type: string.

timeElapsedRecentTask

Description

The time elapsed for the most recent task in the group.

Line Class Constructor

The following is the Line class constructor.

Line

Syntax

```
Line()
```

Description

The Line object describes the line of the extension. JSMCAPI only supports one extension with two lines and two extensions with one line in each.

Parameters

None.

Copyright © 1988-2007, Oracle. All rights reserved.

Returns

A Line object.

Line Class Fields

The following are the Line class fields.

call

Description

The Call object on the line.

Type: Call object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Call Class Constructor, page 246

caps

Description

The capabilities of the line.

Type: LineCaps object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," LineCaps Class Constructor, page 346

connectionid

Description

The call connection ID on the line.

Type: string.

id

Description

The line ID.

Type: string.

 Copyright © 1988-2007, Oracle. All rights reserved.

isMuted

Description

Flag to see whether extension is muted or not.

Type: string.

state

Description

The line state.

Type: string with the following constants:

Value	Description
ST_DIALING	The dialing state.
ST_DROPPED	The dropped state.
ST_HELD	The held state.
ST_IDLE	The idle state.
ST_OFFHOOK	The offhook state.
ST_RINGING	The ringing state.
ST_TALKING	The talking state.

Line Class Methods

The following are the Line class methods.

alternate

Syntax

alternate(*reason*)

Description

Alternate a call.

Parameters

Parameter	Description
reason	The reason to alternate a call.

Copyright © 1988-2007, Oracle. All rights reserved.

Returns

None.

answer

Syntax

answer(*reason*)

Description

Answer a call.

Parameters

Parameter	Description
reason	The reason to answer a call.

Returns

None.

attachUserData

Syntax

attachUserDate(*userdata, reason*)

Description

Attach user data to a call.

Parameters

Parameter	Description
userdata	The user data to attach to the call.
reason	The reason to attach the user data.

Returns

None.

clear

Syntax

clear(*reason*)

Description

Clear a conference call.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
reason	The reason to clear the conference.

Returns

None.

complete

Syntax

complete(*reason*)

Description

Complete the two-step transfer/conference.

Parameters

Parameter	Description
reason	The reason to complete the two-step transfer/conference.

Returns

None.

conference

Syntax

conference(*destination*, *reason*, *userdata*, *calldata*)

Description

Conference a call.

Parameters

Parameter	Description
destination	The destination being invited into the conference.
reason	The reason to conference.
userdata	The user data to be attached.
calldata	The call data to be modified.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

conferenceSingle

Syntax

conferenceSingle(*destination*, *reason*, *userdata*, *calldata*)

Description

Single-step conference a call.

Parameters

Parameter	Description
destination	The destination being invited into the conference.
reason	The reason to conference.
userdata	The user data to be attached.
calldata	The call data to be modified.

Returns

None.

dial

Syntax

dial(*number*, *reason*, *userdata*)

Description

Dial out.

Parameters

Parameter	Description
destination	The destination being invited into the conference.
reason	The reason to conference.
userdata	The user data to be attached.

Returns

None.

dropParty

Syntax

dropParty(*destination*, *reason*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Drop a party in conference.

Parameters

Parameter	Description
destination	The destination being dropped from the conference.
reason	The reason to drop the party.

Returns

None.

getAni

Syntax

```
getAni()
```

Description

Get the ANI.

Parameters

None.

Returns

Returns a string representing the ANI.

getDescr

Syntax

```
getDescr()
```

Description

Get the description attached to the call on the line.

Parameters

None.

Returns

Returns a string.

getDnis

Syntax

```
getDnis()
```

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Get the DNIS.

Parameters

None.

Returns

Returns a string.

getPadvalue

Syntax

getPadvalue(*key*)

Description

Get the PAD value.

Returns

A string representing PAD value.

getReferenceId

Syntax

getReferenceId()

Description

Get the reference ID attached to the call on the line.

Parameters

None.

Returns

Returns a string.

getUrl

Syntax

getUrl(*defaultUrl*)

Description

Get the URL for screen pop-up.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
defaultUrl	The default pop-up URL.

Returns

Returns a string.

grabCall

Syntax

grabCall(*destination, reason*)

Description

Grab a call from the queue.

Parameters

Parameter	Description
destination	The destination to be grabbed.
reason	The reason for the grab.

Returns

None.

hold

Syntax

hold(*reason*)

Description

Hold a call.

Parameters

Parameter	Description
reason	The reason to hold the call.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

join

Syntax

join(*reason, conferenceId*)

Description

Join an existing call/conference.

Parameters

Parameter	Description
reason	The reason to join the call.
conferenceId	Call id.

Returns

None.

mute

Syntax

mute(*reason*)

Description

Mute an extension.

Parameters

Parameter	Description
reason	The reason to mute the call.

Returns

None.

park

Syntax

park(*destination, reason*)

Description

Park a call.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
destination	The destination on which the call will be parked.
reason	The reason to park the call.

Returns

None.

reconnect

Syntax

reconnect(*reason*)

Description

Reconnect to the original party during two-step transfer/conference.

Parameters

Parameter	Description
reason	The reason to reconnect.

Returns

None.

reject

Syntax

reject(*reason*)

Description

Reject an incoming call.

Parameters

Parameter	Description
reason	The reason to reject.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

release

Syntax

release(*reason*)

Description

Release a call.

Parameters

Parameter	Description
reason	The reason to release the call.

Returns

None.

retrieve

Syntax

retrieve(*reason*)

Description

Retrieve a held call.

Parameters

Parameter	Description
reason	The reason to retrieve the call.

Returns

None.

sendDTMF

Syntax

Syntax

sendDTMF(*reason*,*stringDTMF*)

Description

Send DTMF tones to the switch.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
reason	The reason for requesting DTMF.
stringDTMF	DTMF tones string (0-9,*,#).

Returns

None.

setcallresult

Syntax

setcallresult(*result*)

Description

Set call result.

Parameters

Parameter	Description
result	Result of the call.

Returns

None.

setcallresultDNC

Syntax

setcallresultDNC(*number, reason*)

Description

Do not call.

Parameters

Parameter	Description
number	The number not to be disturbed.
reason	The reason not to disturb.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

setcallresultReschedule

Syntax

setcallresultReschedule(*hour, minutes, day, month,year*)

Description

Reschedule a call.

Parameters

Parameter	Description
hour	Hours
Minutes	Minutes
Day	Day
Month	Month
Year	Year

Returns

None.

transfer

Syntax

transfer(*destination, reason, userdata, calldata*)

Description

Transfer a call.

Parameters

Parameter	Description
destination	The destination to which the call will be transferred.
reason	The reason to transfer.
userdata	The user data to be attached.
calldata	The call data to be modified.

Returns

None.

transferMute

Syntax

transferMute(*destination, reason, userdata, calldata*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Transfer mute a call.

Parameters

Parameter	Description
destination	The destination to which the call will be transferred.
reason	The reason to transfer.
userdata	The user data to be attached.
calldata	The call data to be modified.

Returns

None.

unmute

Syntax

unmute(*reason*)

Description

Unmute a call.

Parameters

Parameter	Description
reason	The reason to unmute a call.

Returns

None.

updateCallData

Syntax

updateCallData(*calldata, reason*)

Description

Update call data to a call.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
calldata	The call data to update.
reason	The reason to update.

Returns

None.

Line Class Callback Event Methods

The following are the Line class callback event methods.

onAlternating

Syntax

onAlternating(*event*)

Description

Fires when alternating the call.

onAnswering

Syntax

onAnswering(*event*)

Description

Fires when answering the call.

onAttachingUD

Syntax

onAttachingUD(*event*)

Description

Fires when attaching user data.

onCallDataChanged

Syntax

onCallDataChanged(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when the call data on the call is changed.

onCapabilitiesChanged

Syntax

onCapabilitiesChanged(*event*)

Description

Fires when the capabilities changed.

onClearing

Syntax

onClearing(*event*)

Description

Fires when clearing the conference.

onCompleting

Syntax

onCompleting(*event*)

Description

Fires when completing the two-step transfer/conference.

onConferencing

Syntax

onConferencing(*event*)

Description

Fires when conferencing the call.

onDialing

Syntax

onDialing(*event*)

Description

Fires when there is an outgoing call.

Copyright © 1988-2007, Oracle. All rights reserved.

onDropped

Syntax

onDropped(*event*)

Description

Fires when the call is released.

onError

Syntax

onError(*event*)

Description

Fires when there is error.

onGrabbing

Syntax

onGrabbing(*event*)

Description

Fires when grabbing a call from a queue.

onHeld

Syntax

onHeld(*event*)

Description

Fires when the call is on hold.

onHolding

Syntax

onHolding(*event*)

Description

Fires when holding the call.

onJoining

Syntax

onJoining(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when joining a call or a conference.

onMuted

Syntax

onMuted(*event*)

Description

Fires when a call is muted.

onOffHook

Syntax

onOffHook(*event*)

Description

Fires when the line is off hook.

onOnHook

Syntax

onOnHook(*event*)

Description

Fires when the line is on hook.

onParking

Syntax

onParking(*event*)

Description

Fires when parking the call.

onPartyAdded

Syntax

onPartyAdded(*event*)

Description

Fires when a new call party coming.

Copyright © 1988-2007, Oracle. All rights reserved.

onPartyChanged

Syntax

onPartyChanged(*event*)

Description

Fires when the call party has changed.

onPartyRemoved

Syntax

onPartyRemoved(*event*)

Description

Fires when a call party is removed.

onReconnecting

Syntax

onReconnecting(*event*)

Description

Fires when reconnecting the call.

onRejected

Syntax

onRejected(*event*)

Description

Fires when call is rejected.

onRejecting

Syntax

onRejecting(*event*)

Description

Fires agent rejects an incoming call.

onReleasing

Syntax

onReleasing(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when releasing the call.

onRetrieving

Syntax

onRetrieving(*event*)

Description

Fires when retrieving the call.

onRinging

Syntax

onRinging(*event*)

Description

Fires when there is an incoming call.

onSetcallresult

Syntax

onSetcallresult(*event*)

Description

Fires when there is a request for setting call result.

onSetcallresultDNC

Syntax

onSetcallresultDNC(*event*)

Description

Fires when there is a request for do not call.

onSetcallresultReschedule

Syntax

onSetcallresultReschedule(*event*)

Description

Fires when there is a request to reschedule a call.

Copyright © 1988-2007, Oracle. All rights reserved.

onTalking

Syntax

onTalking(*event*)

Description

Fires when the call is established.

onTransfering

Syntax

onTransfering(*event*)

Description

Fires when transferring the call.

onUnmuted

Syntax

onUnmuted(*event*)

Description

Fires when call is unmuted.

onUpdatingCD

Syntax

onUpdatingCD(*event*)

Description

Fires when updating the call data.

onUserDataChanged

Syntax

onUserDataChanged(*event*)

Description

Fires when the user data attached on the call is changed.

LineCaps Class Constructor

The following is the LineCaps class constructor.

Copyright © 1988-2007, Oracle. All rights reserved.

LineCaps

Syntax

LineCaps(*strCaps*)

Description

Describes the line's capabilities.

Parameters

Parameter	Description
strCaps	The line's capabilities.

Returns

A LineCaps object.

LineCaps Class Fields

The following are the LineCaps class fields.

canAlternate

Description

The line has alternate capability.

Type: boolean.

canAnswer

Description

The line has answer capability.

Type: boolean.

canAttachUserDate

Description

The line has attach user data capability.

Type: boolean.

Copyright © 1988-2007, Oracle. All rights reserved.

canClear

Description

The line has clear capability.

Type: boolean.

canComplete

Description

The line has complete capability.

Type: boolean.

canConference

Description

The line has conference capability.

Type: boolean.

canConferenceSingle

Description

The line has conference single capability.

Type: boolean.

canDropParty

Description

The line has drop party capability.

Type: boolean.

canHold

Description

The line has hold capability.

Type: boolean.

canMute

Description

The line has mute capability.

Type: boolean.

Copyright © 1988-2007, Oracle. All rights reserved.

canPark

Description

The line has park capability.

Type: boolean.

canReconnect

Description

The line has reconnect capability.

Type: boolean.

canReject

Description

The line has reject capability.

Type: boolean.

canRelease

Description

The line has release capability.

Type: boolean.

canRetrieve

Description

The line has retrieve capability.

Type: boolean.

canSendDTMF

Description

The line has DTMF capability.

Type: boolean.

canSetcallresult

Description

The line has set call result capability.

Type: boolean.

Copyright © 1988-2007, Oracle. All rights reserved.

canSetcallresultDNC

Description

The line has do not call capability.

Type: boolean.

canSetcallresultReschedule

Description

The line has reschedule capability.

Type: boolean.

canTransfer

Description

The line has transfer capability.

Type: boolean.

canTransferMute

Description

The line has transfer mute capability.

Type: boolean.

canUnmute

Description

The line has unmute capability.

Type: boolean.

canUpdateCallData

Description

The line has update call data capability.

Type: boolean.

MCEvent Class Constructor

The following is the MCEvent class constructor.

Copyright © 1988-2007, Oracle. All rights reserved.

MCEvent

Syntax

```
MCEvent()
```

Description

The MCEvent will be passed to the application event handler.

Parameters

None.

Returns

An MCEvent object.

MCEvent Class Fields

The following are the MCEvent class fields.

extension

Description

The extension associated with the event.

Type: Extension object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Extension Class Constructor, page 287

group

Description

The group associated with the event.

Type: Group object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Group Class Constructor, page 316

reason

Description

The reason associated with the event.

Type: Reason object.

Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Reason Class Constructor, page 357

user

Description

The user associated with the event.

Type: User object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” User Class Constructor, page 377

MediaType Class Constructor

The following is the MediaType class constructor.

MediaType

Syntax

```
MediaType()
```

Description

Distinguishes between email and CTI media types.

Parameters

None.

Returns

A MediaType object.

MediaType Class Field

The following is the MediaType field.

type

Description

The media type.

Type: string with the following constants:

 Copyright © 1988-2007, Oracle. All rights reserved.

Value	Description
MT_CHAT	The chat media type.
MT_EMAIL	The email media type.
MT_GENERIC	The generic media type.
MT_VOICE	The voice media type (CTI).

PSMC Class Constructor

The following is the PSMC class constructor.

PSMC

Syntax

```
PSMC()
```

Description

The global object that the application can access.

Parameters

None.

Returns

A PSMC object.

PSMC Class Fields

The following are the PSMC class fields.

renserver

Description

The PSMC instance of the RenServer object.

Type: RenServer object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” RenServer Class Constructor, page 359

Copyright © 1988-2007, Oracle. All rights reserved.

servers

Description

An associative array the different server objects, indexed by server ID.

Type: associative array.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Server Class Constructor, page 360

sessions

Description

An associative array that holds the different session objects, indexed by server ID.

Type: associative array.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Session Class Constructor, page 363

PSMC Class Methods

The following are the PSMC class methods.

closeSession

Syntax

closeSession(*serverId*)

Description

Closes the specified session and server object and deletes them.

Parameters

Parameter	Description
serverId	The ID of the REN server cluster for a UQ server, or the server ID for a CTI server.

Returns

None.

getCallById

Syntax

getCallById(*id*)

 Copyright © 1988-2007, Oracle. All rights reserved.

Description

Retrieve the call object by the call ID.

Parameters

Parameter	Description
id	The call ID.

Returns

Type: Call object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," Call Class Constructor, page 246

getChatById

Syntax

getChatById(*id*)

Description

Retrieve the chat object by the task id.

Parameters

Parameter	Description
id	The task ID.

Returns

Type: chat object.

getEmailById

Syntax

getEmailById(*id*)

Description

Retrieve the email object by the task id.

Parameters

Parameter	Description
id	The task ID.

Copyright © 1988-2007, Oracle. All rights reserved.

Returns

Type: email object.

getGenericTaskById

Syntax

getGenericTaskById(*id*)

Description

Retrieve the Generic object by the task id.

Parameters

Parameter	Description
id	The task ID.

Returns

Type: generic object.

getLineById

Syntax

getLineById(*id*)

Description

Retrieve the line object by the line ID.

Parameters

Parameter	Description
id	The line ID.

Returns

Type: Line object.

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Line Class Constructor, page 325

openSession

Syntax

openSession(*id*)

Description

Creates the session and server objects for the PSMC.

 Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
id	The ID of the REN server cluster for a UQ server, or the server ID for a CTI server.

Returns

None.

start

Syntax

```
start()
```

Description

Start the JSMCAPI.

Parameters

None.

Returns

None.

stop

Syntax

```
stop()
```

Description

Stop the JSMCAPI.

Parameters

None.

Returns

None.

Reason Class Constructor

The following is the Reason class constructor.

Copyright © 1988-2007, Oracle. All rights reserved.

Reason

Syntax

reason(*code*, *desc*, *reasondata1*, *reasondata2*, *reasondata3*)

Description

The Reason object carries the reason code and reason description.

Parameters

Parameter	Description
code	The reason code.
desc	The reason description.
reasondata1	The reason data1.
reasondata2	The reason data2.
reasondata3	The reason data3

Returns

Type: Reason object.

Reason Class Fields

The following are the Reason class fields.

code

Description

The reason code.

Type: string.

desc

Description

The reason description.

Type: string.

reasonData1

Description

Place holder for extra data.

Copyright © 1988-2007, Oracle. All rights reserved.

Type: string.

reasonData2

Description

Place holder for extra data.

Type: string.

reasonData3

Description

Place holder for extra data.

Type: string.

RenServer Class Constructor

The following is the RenServer class constructor.

RenServer

Syntax

```
RenServer()
```

Description

The RenServer object describes the REN server cluster information.

Parameters

None.

Returns

A RenServer object.

RenServer Class Fields

The following are the RenServer class fields.

isRunning

Description

Flag containing the state of the REN server connection.

Copyright © 1988-2007, Oracle. All rights reserved.

Type: boolean.

url

Description

The REN server URL.

Type: string.

RenServer Class Callback Event Methods

The following are the RenServer class callback event methods.

onDown

Syntax

onDown(*event*)

Description

Fires when the REN server is down.

onUp

Syntax

onUp(*event*)

Description

Fires when the REN server is up.

Server Class Constructor

The following is the Server class constructor.

Server

Syntax

Server(*serverId*)

Description

The Server object describes the server information.

 Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
serverId	The ID of the REN server cluster for a UQ server, or the server ID for a CTI server.

Returns

A Server object.

Server Class Fields

The following are the Server class fields.

id

Description

The server ID.

Type: string.

info

Description

The server information string.

Type: string.

state

Description

The server state.

Type: string with the following constants.

Value	Description
ST_INSERVICE	The server is in service.
ST_OUTOFSERVICE	The server is out of service.

type

Description

The type of server, either CTI or UQ.

Type: string with the following constants.

Copyright © 1988-2007, Oracle. All rights reserved.

Value	Description
TYPE_CTI	The server is a CTI server.
TYPE_MCS	The server is a third-party MultiChannel server.
TYPE_UQ	The server is a queue server.

Example

The TYPE_* fields are global fields, which may be accessed without instantiating an object. They are accessed as shown in this example:

```
Server.TYPE_UQ = "UQ";
```

See Also

Chapter 13, “Understanding JSMCAPI Classes,” GLOBALS Class Fields, page 312

Server Class Callback Event Methods

The following are the Server class callback event methods.

onBroadcast

Syntax

onBroadcast(*event*)

Description

Fires when there is broadcast message.

onHbLost

Syntax

onHbLost(*event*)

Description

Fires when the server’s heartbeat is lost.

onHbRecovered

Syntax

onHbRecovered(*event*)

Description

Fires when the server’s heartbeat is recovered.

 Copyright © 1988-2007, Oracle. All rights reserved.

onInService

Syntax

onInService(*event*)

Description

Fires when the server changes state to in service.

onOutOfService

Syntax

onOutOfService(*event*)

Description

Fires when the server changes state to out of service.

onRestart

Syntax

onRestart(*event*)

Description

Fires when server restarts.

Session Class Constructor

The following is the Session class constructor.

Session

Syntax

Session(*serverId*)

Description

The Session object describes the session with the server.

Parameters

Parameter	Description
serverId	The ID of the REN server cluster for a UQ server, or the server ID for a CTI server.

Returns

Returns a Session object.

Copyright © 1988-2007, Oracle. All rights reserved.

Session Class Fields

The following are the Session class fields.

addresses

Description

The associative array of addresses that the session register.

Type: associative array.

buddies

Description

The buddy hash table in this session.

Type: associative array.

groups

Description

The group hash table in this session.

Type: associative array.

id

Description

The session ID generated by the server.

Type: string.

serverId

Description

The serverId will be unique for every session. It is used to lookup the protocol objects.

Type: string.

state

Description

The session state.

Type: string with the following constants.

Copyright © 1988-2007, Oracle. All rights reserved.

Value	Description
ST_ACTIVE	The session is in the active state.
ST_CLOSED	The session is in the closed state.
ST_CLOSING	The session is in the closing state.
ST_IDLE	The session is in the idle state.

user

Description

The current user for the session.

Type: User object.

See Also

Chapter 13, "Understanding JSMCAPI Classes," User Class Constructor, page 377

Session Class Methods

The following are the Session class methods.

broadcastSubscribe

Syntax

broadcastSubscribe(*cluster*, *queue*, *task*, *state*, *presence*, *method*)

Description

Subscribe to broadcast messages.

Returns

Object.

broadcastUnsubscribe

Syntax

broadcastunsubscribe(*type*)

Description

Unsubscribe to broadcast messages.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

close

Syntax

```
close()
```

Description

Close the session with the server.

Parameters

None.

Returns

None.

open

Syntax

```
open()
```

Description

Open the session with the server.

Parameters

None.

Returns

None.

registerAddress

Syntax

```
registerAddress(address)
```

Description

Register the address to the server so that the server will report all events that occur on this address.

Parameters

Parameter	Description
address	The address to be registered.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

registerBuddy

Syntax

registerBuddy(*buddy*)

Description

Register the buddy to the JSMCAPI so that the JSMCAPI will report all state events that occur on this buddy.

Parameters

Parameter	Description
buddy	The buddy to be registered.

Returns

None.

registerGroup

Syntax

registerGroup(*group*)

Description

Register the group to the server so that the server will report all events that occur on this group.

Parameters

Parameter	Description
group	The group to be registered.

Returns

None.

registerUser

Syntax

registerUser(*user*)

Description

Register the user to the server so that the server will report all those events that happen on this user.

Parameters

Parameter	Description
user	The user to be registered.

Copyright © 1988-2007, Oracle. All rights reserved.

Returns

None.

setAutoRecovery

Syntax

setAutoRecovery(*autorecover*)

Description

Sets the auto recovery feature of the queue server connection to off.

Parameters

Parameter	Description
autoRecover	Recover.

statPublish

Syntax

statPublish(*userStat, groupStat*)

Description

Sets the statistics publishing levels for the queue server.

Parameters

Parameter	Description
userStat	Either 0, 1, or 2 for UserStatistics publishing. Enter 0 for no statistics publishing, 1 for Statistics1 publishing, and 2 for Statistics2 and Statistic1 publishing.
groupStat	Either 0, 1, or 2 for GroupStatistics publishing. Enter 0 for no statistics publishing, 1 for Statistics1 publishing, and 2 for Statistics2 and Statistic1 publishing.

Returns

None.

unregisterAddress

Syntax

unRegisterAddress(*address*)

Description

Unregister the address from the server so that the server will not report any events about this address.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
address	The address to be unregistered.

Returns

None.

unregisterBuddy

Syntax

unregisterBuddy(*buddy*)

Description

Unregister the buddy from the JSMCAPI so that the JSMCAPI will not report any events about the buddy state change.

Parameters

Parameter	Description
buddy	The buddy to be unregistered.

Returns

None.

unregisterGroup

Syntax

unregisterGroup(*group*)

Description

Unregister the group from the server so that the server will not report any events about this group.

Parameters

Parameter	Description
group	The group to be unregistered.

Returns

None.

unregisterUser

Syntax

unregisterUser(*user*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Unregister the user from the server so that the server will not report any events to this user.

Parameters

Parameter	Description
user	The user to be unregistered.

Returns

None.

Session Class Callback Event Methods

The following are the Session class callback event methods.

onAddressRegistered

Syntax

onAddressRegistered(*event*)

Description

Fires when the address is registered by the session.

onAddressUnregistered

Syntax

onAddressUnregistered(*event*)

Description

Fires when the address is unregistered by the session.

onBuddyRegistered

Syntax

onBuddyRegistered(*event*)

Description

Fires when the buddy is registered by the session.

onBuddyUnregistered

Syntax

onBuddyUnregistered(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when the buddy is unregistered by the session.

onClosed

Syntax

onClosed(*event*)

Description

Fires when the session is closed.

onError

Syntax

onError(*event*)

Description

Fires when there is a session error.

onGroupRegistered

Syntax

onGroupRegistered(*event*)

Description

Fires when a group is registered by the session.

onGroupUnregistered

Syntax

onGroupUnregistered(*event*)

Description

Fires when a group is unregistered by the session.

onInfo

Syntax

onInfo(*event*)

Description

Fires when there is a session information event, such as the user is already logged in.

Copyright © 1988-2007, Oracle. All rights reserved.

onOpened

Syntax

onOpened(*event*)

Description

Fires when the session is opened.

onUserRegistered

Syntax

onUserRegistered(*event*)

Description

Fires when the user is registered by the session.

onUserUnregistered

Syntax

onUserUnregistered(*event*)

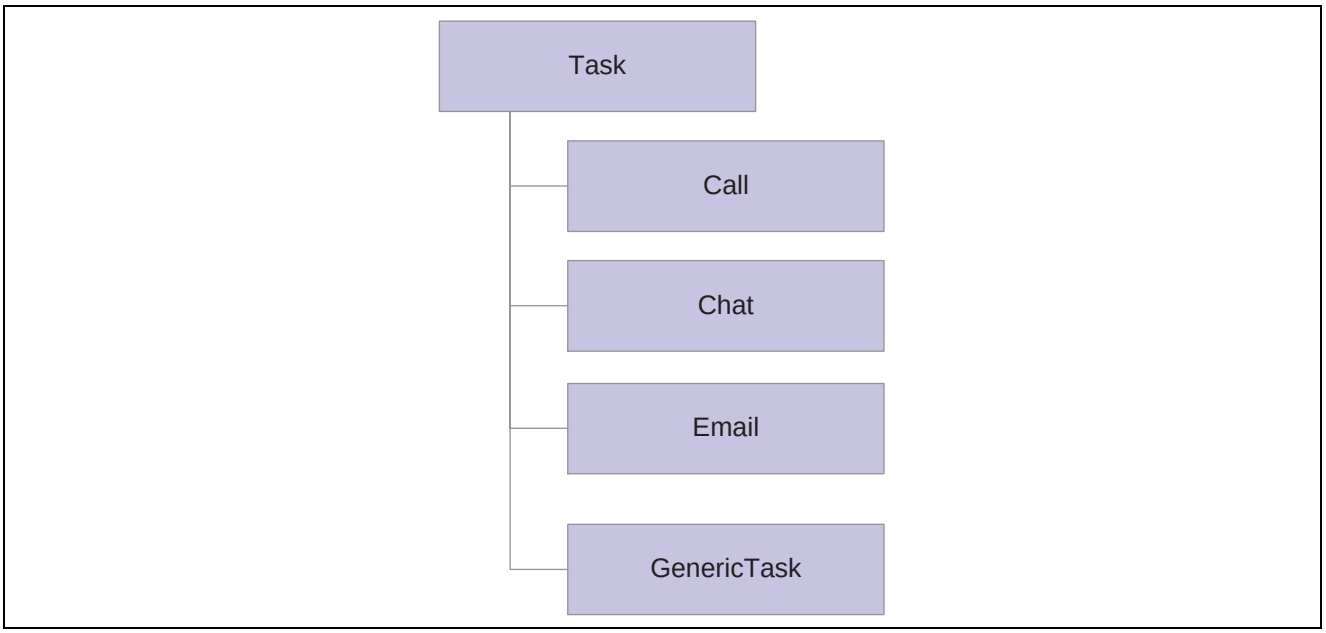
Description

Fires when the user gets unregistered by the session.

Task Class Hierarchy

The Task class can be extended by other subclasses. The following flowchart shows the different subclasses and how they interrelate.

Copyright © 1988-2007, Oracle. All rights reserved.



Task class hierarchy

See Also

Chapter 13, “Understanding JSMCAPI Classes,” Call Class Constructor, page 246

Chapter 13, “Understanding JSMCAPI Classes,” Chat Class Constructor, page 251

Chapter 13, “Understanding JSMCAPI Classes,” Email Class Constructor, page 272

Chapter 13, “Understanding JSMCAPI Classes,” GenericAddress Class Constructor, page 297

Task Class Constructor

The following is the Task class constructor.

Task

Syntax

```
Task()
```

Description

Task is an abstract base class.

Parameters

None.

Returns

Type: Task object.

Copyright © 1988-2007, Oracle. All rights reserved.

Task Class Fields

The following are the Task class fields.

caseid

Description

The associated case ID.

Type: string.

cost

Description

The associated cost.

Type: string.

customerid

Description

The associated customer ID.

Type: string.

group

Description

The group for which task is assigned.

Type: string.

id

Description

The task ID.

Type: string.

onStat

Description

Method that will be triggered when there is statistics published regarding this task.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

priority

Description

The priority of the task.

Type: string.

type

Description

The type of the task.

Type: string, one of the following:

Value	Description
TYPE_A2ACHAT	The A2AChat type.
TYPE_CHAT	The chat type.
TYPE_CTI	The CTI type.
TYPE_EMAIL	The email type.
TYPE_GENERIC	The generic type.

Example

The TYPE_* fields are global fields, which may be accessed without instantiating an object. They are accessed as shown in this example:

```
Task.TYPE_GENERIC = "GENERIC";
```

See Also

Chapter 13, “Understanding JSMCAPI Classes,” GLOBALS Class Fields, page 312

urlAbs

Description

The absolute URL that is useful in constructing URL for new window.

Type: urlAbs object.

urlRel

Description

The relative URL that is useful in constructing URL for new window.

Type: urlRel object.

Copyright © 1988-2007, Oracle. All rights reserved.

TaskStatistics Class Constructor

The following is the TaskStatistics class constructor.

TaskStatistics

Syntax

```
TaskStatistics()
```

Description

Describes the task statistics information.

Returns

Task statistics object.

TaskStatistics Class Fields

The following are the TaskStatistics class fields.

data

Description

Key value pairs that includes all statistics.

Type: collection.

holdTime

Description

Time duration that the task is on hold.

Type: number

queueTime

Description

Time duration in the queue for this task.

Type: number

Copyright © 1988-2007, Oracle. All rights reserved.

talkTime

Description

Time duration that the task is established.

Type: number

User Class Constructor

The User class extends the _User class.

See Chapter 13, “Understanding JSMCAPI Classes,” _User Class Constructor, page 230.

The following is the User class constructor.

User

Syntax

```
User(id, name, agentId, agentPassword)
```

Description

Describes the user/agent.

Parameters

Parameter	Description
id	The user ID.
name	The user name.
agentId	The agent ID.
agentPassword	The agent’s password.

Returns

A User object.

User Class Fields

The User class inherits the following fields from the _User class:

- agentId
- caps
- id
- name

Copyright © 1988-2007, Oracle. All rights reserved.

- presences
- ST_LOGGEDIN
- ST_LOGGEDOUT
- ST_NOTREADY
- ST_READY
- ST_UNKNOWN
- ST_WORKNOTREADY
- ST_WORKREADY
- states
- statistics
- statistics1
- statistics2

The following are the User class fields.

addresses

Description

The addresses associated with this agent.

Type: associative array.

agentPassword

Description

The password for this agent.

Type: string.

language

Description

The language for this agent.

Type: string.

registeredAddresses

Description

The registered addresses for this user.

Type: associative array.

Copyright © 1988-2007, Oracle. All rights reserved.

statesUq

Description

States of the user for the queue server indexed by group number

Type: object.

User Class Methods

The following are the User class methods.

ctiBusyUq

Syntax

ctiBusyUq(*busy*, *subtractcost*, *task*)

Description

Flip ctiBusy flag and/or reduce cost of a CTI task from an agent's workload. If the task is the same as the agent's assigned task, then the task's cost will be used. In case they do not match the task type is checked and the default cost of the task is used to re-calculate the workload.

Parameters

Parameter	Description
busy	The ctibusy flag that can have a value of 0 or 1.
subtractcost	Flag to reduce cost of the cti task from the agent's workload. It can be set to 0 or 1.
task	The task object.

disableMedia

Syntax

disableMedia(*group*, *mediaType*, *extension*, *reason*)

Description

Disable a media channel for an agent on a group/queue.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
group	The group (queue) on which to disable the specified media type.
mediaType	The media type to disable.
extension	The extension on which to disable the specified media type.
reason	The reason for disabling this media type.

Returns

None.

enableMedia

Syntax

enableMedia(*group*, *mediaType*, *extension*, *reason*)

Description

Enable a media channel for an agent on a group/queue.

Parameters

Parameter	Description
group	The group (queue) on which to enable the specified media type.
mediaType	The media type to enable.
extension	The extension on which to enable the specified media type.
reason	The reason for enabling this media type.

Returns

None.

isMediaEnabled

Syntax

isMediaEnabled(*mediaType*, *group*)

Description

Determines if the specified media type is enabled.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
mediaType	Enter the media type.
group	Enter the group.

Returns

Returns true if the media type is enabled.

login

Syntax

login(*group*, *extension*, *reason*)

Description

Log the user in to a queue.

Parameters

Parameter	Description
group	The group to which the user will log in.
extension	The extension.
reason	The reason for the login.

Returns

None.

loginUq

Syntax

loginUq(*group*)

Description

Log a user in to a UQ server queue.

Parameters

Parameter	Description
group	The group (queue) to which the user will login.

Returns

None.

Copyright © 1988-2007, Oracle. All rights reserved.

logout

Syntax

logout(*group, extension, reason*)

Description

Log the user out of the specified group (queue).

Parameters

Parameter	Description
group	The group from which the user will log out.
extension	The extension.
reason	The reason for the log out.

Returns

None.

logoutUq

Syntax

logoutUq(*group*)

Description

Log the user out from the UQ server queue.

Parameters

Parameter	Description
group	The group (queue) from which to log the user out.

Returns

None.

register

Syntax

register(*extension, reason*)

Description

Register an extension for the user.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
extension	The extension to be registered.
reason	The reason to register the extension.

Returns

None.

setNotReady

Syntax

setNotReady(*group*, *presence*, *extension*, *reason*)

Description

Set the user state as not ready.

Parameters

Parameter	Description
group	The group on which the user is changing state.
presence	The presence in the new state.
extension	The extension with which the user is changing state.
reason	The reason to set not ready.

Returns

None.

setPresence

Syntax

setPresence(*group*, *presence*, *extension*, *reason*)

Description

Set the presence for the current state.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
group	The group on which the user is changing presence.
presence	The presence for the state.
extension	The extension with which the user is changing presence.
reason	The reason to set the presence.

Returns

None.

setPresenceUq

Syntax

setPresenceUq(*group*, *presenceText*, *reason*)

Description

Set the Presence for UQ servers.

Parameters

Parameter	Description
group	The group (queue) on which the user is changing his or her presence.
presenceText	The new presence.
reason	The reason to set the new presence.

Returns

None.

setReady

Syntax

setReady(*group*, *presence*, *extension*, *reason*)

Description

Set the user state as ready.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
group	The group on which the user is changing state.
presence	The presence for the state.
extension	The extension with which the user is changing state.
reason	The reason to set the state.

Returns

None.

setWorkNotReady

Syntax

setWorkNotReady(*group*, *presence*, *extension*, *reason*)

Description

Set the user state as work not ready.

Parameters

Parameter	Description
group	The group on which the user is changing state.
presence	The presence for the state.
extension	The extension with which the user is changing state.
reason	The reason to set work not ready.

Returns

None.

setWorkReady

Syntax

setWorkReady(*group*, *presence*, *extension*, *reason*)

Description

Set the user state as work ready.

Copyright © 1988-2007, Oracle. All rights reserved.

Parameters

Parameter	Description
group	The group on which the user is changing state.
presence	The presence for the state.
extension	The extension with which the user is changing state.
reason	The reason to set work ready.

Returns

None.

unregister

Syntax

unregister(*extension, reason*)

Description

Unregister an extension for the user.

Parameters

Parameter	Description
extension	The extension to be unregistered.
reason	The reason to unregister the extension.

Returns

None.

User Class Callback Event Methods

The following are the User class callback event methods.

onCapabilitiesChanged

Syntax

onCapabilitiesChanged(*event*)

Description

Fires when capabilities changed.

Copyright © 1988-2007, Oracle. All rights reserved.

onCtiBusy

Syntax

onCtiBusy(*event*)

Description

Fires when the UQ server is notified that the CTI is busy.

onCTIBUSYUq

Syntax

onCTIBUSYUq(*event*)

Description

Fires when the UQ server is notified that the CTI is busy.

onCtiClear

Syntax

onCtiClear(*event*)

Description

Fires when the UQ server is notified that the CTI is clear.

onDropped

Syntax

onDropped(*event*)

Description

Fires when a user is dropped.

onError

Syntax

onError(*event*)

Description

Fires when an error occurs.

onInfo

Syntax

onInfo(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when an information event occurs. Currently only used on UQ server.

onLoggedIn

Syntax

onLoggedIn(*event*)

Description

Fires when the user is logged in.

onLoggedOut

Syntax

onLoggedOut(*event*)

Description

Fires when the user is logged out.

onLoggingIn

Syntax

onLoggingIn(*event*)

Description

Fires when the user is logging in.

onLoggingOut

Syntax

onLoggingOut(*event*)

Description

Fires when the user is logging out.

onMediaDisabled

Syntax

onMediaDisabled(*event*)

Description

Fires when a media type is disabled.

Copyright © 1988-2007, Oracle. All rights reserved.

onMediaEnabled

Syntax

onMediaEnabled(*event*)

Description

Fires when a media type is enabled.

onNotReady

Syntax

onNotReady(*event*)

Description

Fires when a user state changes to not ready.

onPresenceChanged

Syntax

onPresenceChanged(*event*)

Description

Fires when the user's presence is changed.

onReady

Syntax

onReady(*event*)

Description

Fires when the user's state changes to ready.

onRegistered

Syntax

onRegistered(*event*)

Description

Fires when the address is registered.

onRegistering

Syntax

onRegistering(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when registering the address.

onSettingNotReady

Syntax

onSettingNotReady(*event*)

Description

Fires when setting the user's state to not ready.

onSettingPresence

Syntax

(*event*)
onSettingPresence

Description

Fires when setting presence for the current state.

onSettingReady

Syntax

onSettingReady(*event*)

Description

Fires when setting the user's state to ready.

onSettingWorkNotReady

Syntax

onSettingWorkNotReady(*event*)

Description

Fires when setting the user's state to work not ready.

onSettingWorkReady

Syntax

onSettingWorkReady(*event*)

Description

Fires when setting the user's state to work ready.

Copyright © 1988-2007, Oracle. All rights reserved.

onStat

Syntax

onStat(*event*)

Description

Fires when statistics are received.

onStat1

Syntax

onStat1(*event*)

Description

Fires when statistics1 is received.

onStat2

Syntax

onStat2(*event*)

Description

Fires when statistics2 received.

onUnknown

Syntax

onUnknown(*event*)

Description

Fires when the user's state changes to unknown.

onUnregistered

Syntax

onUnregistered(*event*)

Description

Fires when the address gets unregistered.

onUnregistering

Syntax

onUnregistering(*event*)

Copyright © 1988-2007, Oracle. All rights reserved.

Description

Fires when unregistering the address.

onWorkNotReady

Syntax

onWorkNotReady(*event*)

Description

Fires when the user's state changes to work not ready.

onWorkReady

Syntax

onWorkReady(*event*)

Description

Fires when the user's state changes to work ready.

UserCaps Class Constructor

The following is the UserCaps class constructor.

UserCaps

Syntax

UserCaps(*strCaps*)

Description

The UserCaps object describes the user capabilities.

Parameters

Parameter	Description
strCaps	The user capabilities.

Returns

A UserCaps object.

Copyright © 1988-2007, Oracle. All rights reserved.

UserCaps Class Fields

The following are the UserCaps class fields

canHandleChat

Description

Chat task handling capability of the user.

Type: boolean.

canHandleEmail

Description

Email task handling capability of the user.

Type: boolean.

canHandleGeneric

Description

Generic task handling capability of the user.

Type: boolean.

canHandleVoice

Description

Voice task handling capability of the user.

Type: boolean.

canLogin

Description

Login capability of the user.

Type: boolean.

canLogout

Description

Logout capability of the user.

Type: boolean.

Copyright © 1988-2007, Oracle. All rights reserved.

canRefreshState

Description

The refreshState capability of the user.

Type: boolean.

canSetNotReady

Description

The setNotReady capability of the user.

Type: boolean.

canSetPresence

Description

The setPresence capability of the user.

Type: boolean.

canSetReady

Description

The setReady capability of the user.

Type: boolean.

canSetWorkNotReady

Description

The setWorkNotReady capability of the user.

Type: boolean.

canSetWorkReady

Description

The setWorkReady capability of the user.

Type: boolean.

UserData Class Constructor

The following is the UserData class constructor.

Copyright © 1988-2007, Oracle. All rights reserved.

UserData

Syntax

```
UserData()
```

Description

The UserData object describes the key-value pairs of attached user data.

Parameters

None.

Returns

A UserData object.

UserData Class Fields

The following are the UserData class fields constants.

UserData Class Field Constants

Description

UserData class uses the following constants.

Value	Description
COMPONENT	The component.
DESCR	The description
ICSCRIPTPROGRAMNAME	The IC script program name.
ICTYPE	The IC type.
MARKET	The market.
MENU	The menu.
REFERENCEID	The reference ID
TARGET	The target.
URLABS	The absolute URL.
URLREL	The relative URL.

Copyright © 1988-2007, Oracle. All rights reserved.

UserData Class Method

The following is the UserData class method.

addKeyValue

Syntax

addKeyValue(*key*, *value*)

Description

Add the key-value pair to the UserData object.

Parameters

Parameter	Description
key	The key.
value	The value.

Returns

None.

UserStatistics1 Class Constructor

The following is the UseStatistics1 class constructor.

UserStatistics1

Syntax

UserStatistics1()

Description

UQ User statistics sent on user refresh 1.

Parameters

None.

Returns

A UserStatistics1 object.

Copyright © 1988-2007, Oracle. All rights reserved.

UserStatistics1 Class Fields

The following are the UserStatistics1 class fields.

availableCost

Description

The amount of unused work.

ctiBusy

Description

Flag indicating that a user is engaged on CTI task.

currentQueue

Description

Current queue/group the User last logged into.

Type: string.

mostRecentTaskData

Description

Most recent task's data.

Type: string.

mostRecentTaskId

Description

Most recent task's ID.

Type: string.

numTaskAccepted

Description

The number of tasks accepted by the user.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

numTasksDone

Description

The number of tasks done by the user.

numTasksUnassigned

Description

The number of tasks unassigned by the user.

Type: string.

presenceText

Description

The presence text for the user.

Type: string.

reasonFlag

Description

The reason flag for this event.

Type: string.

state

Description

Current state of the user.

Type: string.

timeInCurrentState

Description

Time in current state for the user.

Type: string.

timeSinceLoggedIn

Description

The time since the user logged in.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

UserStatistics2 Class Constructor

The following is the UserStatistics2 class constructor.

UserStatistics2

Syntax

```
UserStatistics2()
```

Description

UQ User statistics sent on user refresh 2 event.

Parameters

None.

Returns

A UserStatistics2 object.

UserStatistics2 Class Fields

The following are the UserStatistics2 class fields.

currentQueue

Description

The current queue or group for the user.

Type: string.

timeIdle

Description

Idle time for the user.

Type: string.

timeInCurrentState

Description

The time in the current state for the user.

Type: string.

Copyright © 1988-2007, Oracle. All rights reserved.

timeNotReady

Description

The time during which the user was in the not ready state.

Type: string.

timeSinceLogin

Description

The time since the user logged in.

Type: string.

totalTimeAvailable

Description

The total time during which the user has been in the available state.

Type: string.

totalTimeUnavailable

Description

The total time during which the user has been in the unavailable state.

Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 14

Configuring the Email Channel

This chapter provides an overview of the email channel and discusses how to:

- Configure PeopleSoft Integration Broker for the email channel.
- Demonstrate the email channel.

Understanding the Email Channel

Because the email channel uses a PeopleSoft Integration Broker gateway, the email channel requires additional configuration outside of PeopleSoft MultiChannel Framework configuration pages.

PeopleSoft MultiChannel Framework (MCF) provides a framework for:

- Fetching emails from a mail server.
- Storing and retrieving email parts in a database.
- Managing attachments.
- Queueing of emails by the queue server.

Application developers use the PeopleCode application package PT_MCF_MAIL to develop PeopleSoft Application Engine programs and PeopleSoft Pure Internet Architecture components for email processing.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, “Mail Classes”.

PeopleSoft MultiChannel Framework email channel features include:

- The GETMAILTARGET target connector.
- A PeopleCode application package, PT_MCF_MAIL, to retrieve, store, and delete email.
- Support for both Post Office Protocol 3 (POP3) and Internet Message Access Protocol 4 (IMAP4) email protocols.
- Support for email attachments, including:
  - An attachments repository running on the same web server as PeopleSoft Integration Broker.
  - URL access to email attachments.
  - A relative addressing scheme to enable flexibility of repository location.
  - User- and role-based security to control access to attachments.
- The option to access email headers, attachments, and body from the mail server.
- Support for multiple message size thresholds to control distribution of email between the database, the attachment repository, and the mail server.

Copyright © 1988-2007, Oracle. All rights reserved.

- Support for UTF8 Unicode as supported by the Sun Java Runtime Environment (JRE) 1.3.1 or later.

  See http://java.sun.com.
- Time zone conversions to support global service-level agreements.
- Sample email application pages demonstrating the functionality of the PT_MCF_MAIL application package.

Handling Email

This section describes some factors to consider when handling email using PeopleSoft MultiChannel Framework.

POP3 Versus IMAP4

Most mail servers support simultaneous client connections through both POP3 and IMAP4. Using IMAP4 for your MCF email connection provides significant benefit over using POP3. IMAP4 allows for the use of email folders, which allows malformed emails to be set aside for separate processing. Quarantining malformed emails allows system administrators to remove or correct invalid emails without interrupting normal processing.

Note. Using IMAP4 for your MCF email connection does not preclude the use of POP3 for any other client connections to your mail server.

Time Zone Offsets

Set the values of the email sent time zone offset (in minutes) and the receive time zone offset (in minutes) whenever possible. The default value is 800, which indicates time zone information is not available. When available, the values range from +720 to −720.

Connector Determination of Email Size

The connector sometimes cannot determine the size of the message. In such cases the size is set to 0 and an error message is written to the gateway error log.

Malformed Emails

Emails with syntax that does not comply to the latest standard definitions may not be processed successfully by MCF. MCF will return as much as possible of non-compliant emails, however, at times, invalid parts will be ignored. For example, email addresses with invalid characters will not be displayed, such as two @ symbols. Also, if the header information is not valid, such as a malformed content-type, the associated email part may not be displayed.

A relatively common example of a malformed email is one in which there is 8-bit encoded information in a header, such as the To, Cc or Subject field, or in an attachment file name. According to the standard definitions, this information must be encoded using a 7-bit encoding, as described in RFC 2045 and others. This encoding is usually done by the email sender.

Note. PeopleSoft's email reader only supports email messages that follow the SMTP specification. Many email clients and mail servers have their own proprietary formats, please be aware that PeopleSoft MCF will only interpret these emails according to the RFC specification. For example, some mail servers and clients allow 8-bit encoded data in email headers. PeopleSoft's email reader does not allow this, however, as it does not comply with the specification.

 Copyright © 1988-2007, Oracle. All rights reserved.

Configuring PeopleSoft Integration Broker for the Email Channel

This section discusses how to:

- Configure the gateway.
- Configure GETMAILTARGET properties.

Configure nodes and gateways for the email channel in PeopleSoft Integration Broker.

Pages Used to Configure Integration Broker for Email

Page Name	Object Name	Navigation	Usage
Node Info	IB_NODE	PeopleTools, Integration Broker, Node Definitions, Node Info	Define the MCF_GetMail node.
Connectors	IB_NODECONN	PeopleTools, Integration Broker, Node Definitions, Connectors	Define the MCF_GetMail connector.
Gateways	IB_GATEWAY	PeopleTools, Integration Broker, Gateways, Gateway ID	Set up the PeopleSoft Integration Broker gateway for the MCF_GetMail connector node.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker, "Managing Integration Gateways"

Configuring the Gateway

Access the Gateways page.

1. Create a local gateway.
2. Enter the gateway URL in the form http://<gatewayservername>/PSIGW /PeopleSoftListeningConnector, where <gatewayservername> is the fully qualified host name for the gateway machine.

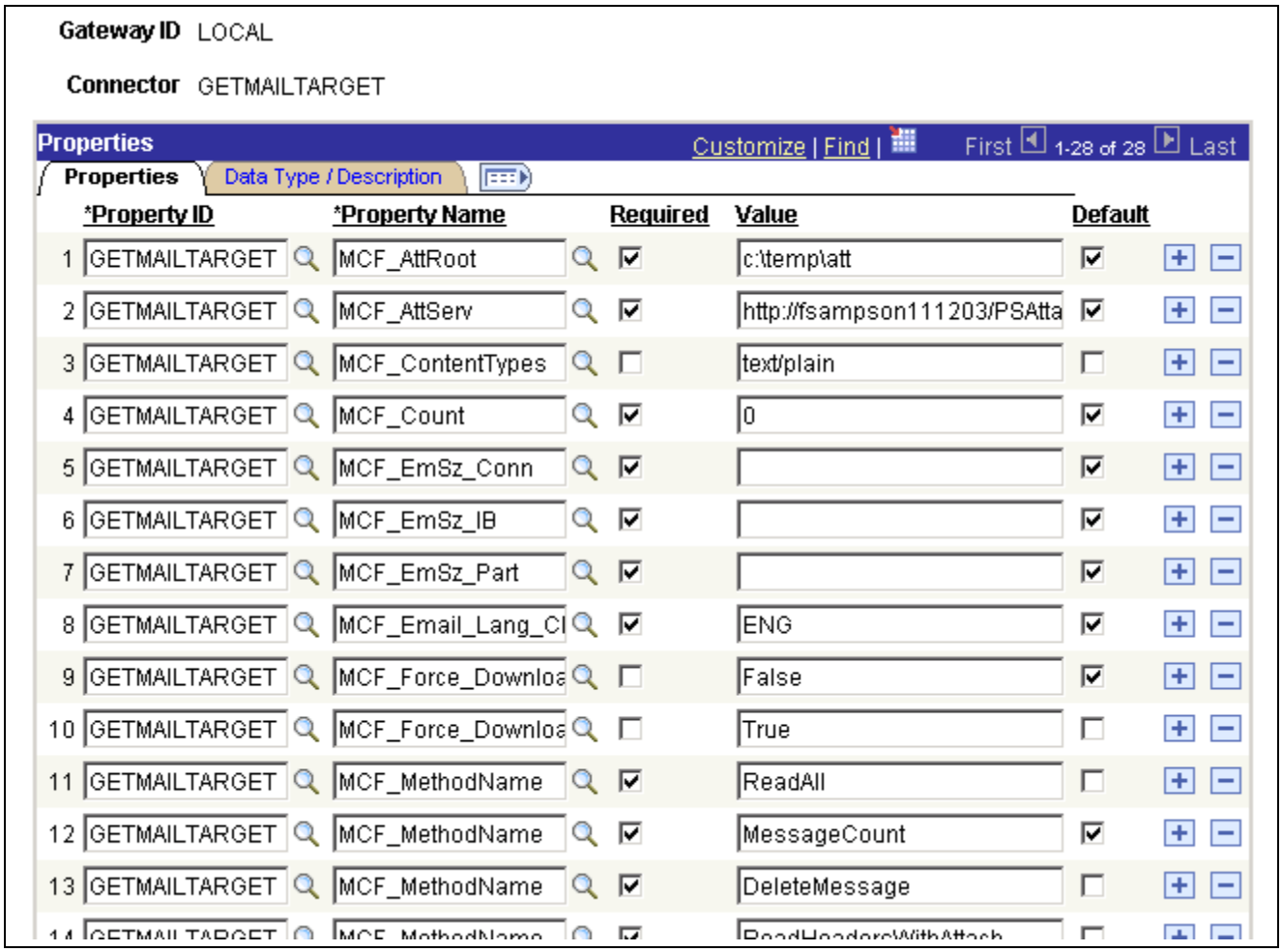
   For example, http://host1.peoplesoft.com/PSIGW/PeopleSoftListeningConnector.
3. Save the page.
4. Click Load Gateway Connectors.

   The connector properties appear.
5. Save the page again.
6. Configure the GETMAILTARGET connector properties.

Configuring GETMAILTARGET Properties

Access the Connector Properties page.

Copyright © 1988-2007, Oracle. All rights reserved.



Connector Properties page

Configure GETMAILTARGET connector properties on the MCF_GETMAIL node. You can also set threshold values for email routed to the attachment repository.

The following table lists node properties:

Copyright © 1988-2007, Oracle. All rights reserved.

Property Name	Default Value	Description
MCF_AttRoot	*c:\temp\att*	Defines the location of the attachment repository on the gateway. For example, if the attachment root is c:\temp\att, then attachments for emails addressed to user name *support* are downloaded to c:\temp\att\<mail server host name>\<mailbox>. **Note.** The directory on the remote machine where the attachments are stored should be shared. The user permissions for should be stored on the node (local node) that will retrieve the attachments. **Note.** The user name or mailbox portion of the attachment root path may include encoded characters. All email attachments, and email parts not of the content types specified for the MCF_ContentTypes property, are always written to the attachment repository. A relative URL is returned to the application that allows the application to access the attachment. The attachment retrieval depends on the file system of the operating system. For Windows NTFS/FAT 16 file system, the folder length or maximum characters of the file extension is 256/512. For Unix and Solaris UFS file system, the length is 256. Attachment security is managed at a user and role level using the PT_MCF_EMAIL application package. Users can configure attachment folder paths using metastrings to enable multiple folders based on several variables: • *%OPRID%:* PeopleSoft user ID. • *%DBNAME%:* database name. • *%CURRDATE%:* current date. • *%CURRHOUR%:* current hour.

Copyright © 1988-2007, Oracle. All rights reserved.

Property Name	Default Value	Description
MCF_AttServ	*http://<machine_name>.<domain_name>/PSAttachServlet/ps/*	Defines the location of the MCF attachment repository servlet, located on the gateway web server. Attachment relative URLs are appended to this address to create a fully qualified URL to reference an attachment in the repository. **Note.** When you are setting a remote gateway to access email attachments stored on the remote machine, the database that the gateway/psattachservlet accesses should have the permissions to view the attachments. **Note.** If your web server is installed with a PeopleSoft Pure Internet Architecture site name other than *ps,* include the site name in the specified path.
MCF_ContentTypes	*text/plain*	Specifies email content types that will be stored in the database. Content types not specified will be stored as attachments. Enter content types separated by commas. For example, you can Enter *text/html* and *text/xml* to enable storage in the database of HTML email and email containing XML. The content type *text/plain* is always stored in the database, regardless of values specified for MCF_ContentTypes. Do not include binary encoded content types in the content types list unless the application can handle base64 encoded text.
MCF_Count	*0*	Defines the default number of emails retrieved from the mail server unless otherwise specified in the SyncRequest. The PeopleSoft MultiChannel Framework email application package always specifies the number of emails to be retrieved and does not reference this property.

Copyright © 1988-2007, Oracle. All rights reserved.

Property Name	Default Value	Description
MCF_EmSz_Conn	*4000000*	Defines the connector threshold, in bytes. This parameter ensures that emails retrieved from a POP3 mail server do not exceed the available memory on the gateway server. Email content retrieved from IMAP4 servers can be streamed to a file on the attachment repository, but content retrieved from POP3 servers must first be read into memory before being written to a file. Therefore, the MCF_EmSz_Conn threshold can be set very high for IMAP4 mail servers. PeopleSoft MultiChannel Framework does not process an email if its size is greater than this value. The status returned to the application is *1*. The triggering email is neither downloaded nor deleted from the mail server. Only the email header is returned to the application. See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Mail Classes".
MCF_EmSz_IB	*500000*	Defines the PeopleSoft Integration Broker threshold, in bytes. This is the maximum size message that the GETMAILTARGET connector can send through PeopleSoft Integration Broker. The purpose of this threshold is to ensure that retrieved emails do not exceed the available memory on the gateway server and that the server running the application that requested the emails. A PeopleSoft Integration Broker message can contain one or more emails, depending on the number of emails the application retrieves per message. As soon as this threshold is reached, the remainder of the triggering email is written to the repository, and no more emails are retrieved. Emails fetched prior to the triggering email are returned normally. The triggering email is returned with a return status of *2,* indicating that some of its parts were written to the repository. See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Mail Classes".
MCF_EmSz_Part	*30000*	Defines the text part threshold, in bytes. If any text part of an email exceeds this value , the part is routed to the attachment repository. This ensures that excessively large objects of text are not written to the database. For databases with limitations on the maximum size of rows or long text field lengths, this value must be lower than that limit to avoid SQL errors when saving emails.

Copyright © 1988-2007, Oracle. All rights reserved.

Property Name	Default Value	Description
MCF_Email_Lang_CD	*ENG*	Defines the expected language of emails downloaded by this node. This code is included in the email header returned to the application. Configure a node for each language you expect to handle, because Simple Mail Transfer Protocol (SMTP) mail servers and clients do not guarantee correct identification of the email's language when creating an email.
MCF_MethodName	*MessageCount*	Defines the methods associated with the application class package. The defined method is the default method used unless otherwise specified in the SyncRequest. The email application package always specifies the method to be used and does not reference this property. See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Mail Classes".
MCF_Password	None	Defines the default password for the mailbox unless otherwise specified in the SyncRequest. See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Mail Classes".
MCF_Port	*143*	Defines the mail server port to be used. By default, POP3 servers use port 110 and IMAP4 servers use port 143. Confirm the port number with your system administrator and set it accordingly here.
MCF_Protocol	*IMAP4*	Defines the protocol used by the connector to access emails on the mail server. Supported values are *POP3* or *IMAP4*.

 Copyright © 1988-2007, Oracle. All rights reserved.

Property Name	Default Value	Description
MCF_Quarantine	*Quarantine*	Defines a quarantine folder for email meeting the following criteria: • Connector size overflow. • Unsupported encoding. • Unknown Java exception. • JavaMail content error. • No attachment repository. • Mail parse exception. The quarantine folder is created for each user account and must be managed by each email user, not by system administrators. **Note.** Folders are supported only on IMAP4 mail servers.
MCF_Server	None	Defines the fully qualified host name of the mail server. This is the default fully qualified host name of the mail server, unless otherwise specified in the SyncRequest. For example, bigserver.peoplesoft.com. See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "Mail Classes".
MCF_User	None	Defines the user name for the email account (mailbox) being accessed. This is the default user name used unless otherwise specified in the SyncRequest. For example, if emails are addressed to support@peoplesoft.com, the user name is *support*.
MCF_Force_Download_Attachments	*False*	Enables downloading attachments that might otherwise be interpreted as text. If set to True all attachments, including text/plain, irrespective of their size, are downloaded to the attachment repository. This property enables reading of non-ASCII attachments.

Copyright © 1988-2007, Oracle. All rights reserved.

Note. The POP3 and IMAP4 email protocols calculate message size differently. POP3 does not include header size, but IMAP4 does. As a result, message sizes affecting thresholds behave differently depending on the protocol used.

Multipurpose Internet Mail Extensions (MIME), is an Internet standard for representing multipart and multimedia data in email so that they can be exchanged between different email systems. The parts may be nested. PeopleSoft embeds the Sun JavaMail API to implement support for the MIME standard. However, all parts of an email are represented in PeopleSoft as level 1 rowsets, regardless of whatever hierarchy existed in the original email. Email clients determine the MIME format of the emails sent from them, so identical email content sent from different email clients may have a different MIME structures.

Demonstrating the Email Channel

To demonstrate the email channel, use the Email Sample Pages (MCFEM_DEMOERMS_CMP) component.

This section discusses how to:

- Use the GetMail - Server page.
- Use the MailStore - DB page.

The PeopleSoft system provides email sample pages to demonstrate the functionality of the PT_MCF_MAIL application package.

Note. The email sample pages are not intended for any purpose other than demonstration and troubleshooting. They should not be used in production.

The email sample pages can be also used to test the configuration of your integration gateway and MCF_GETMAIL node.

Note. Using the email sample pages requires access to a mail server.

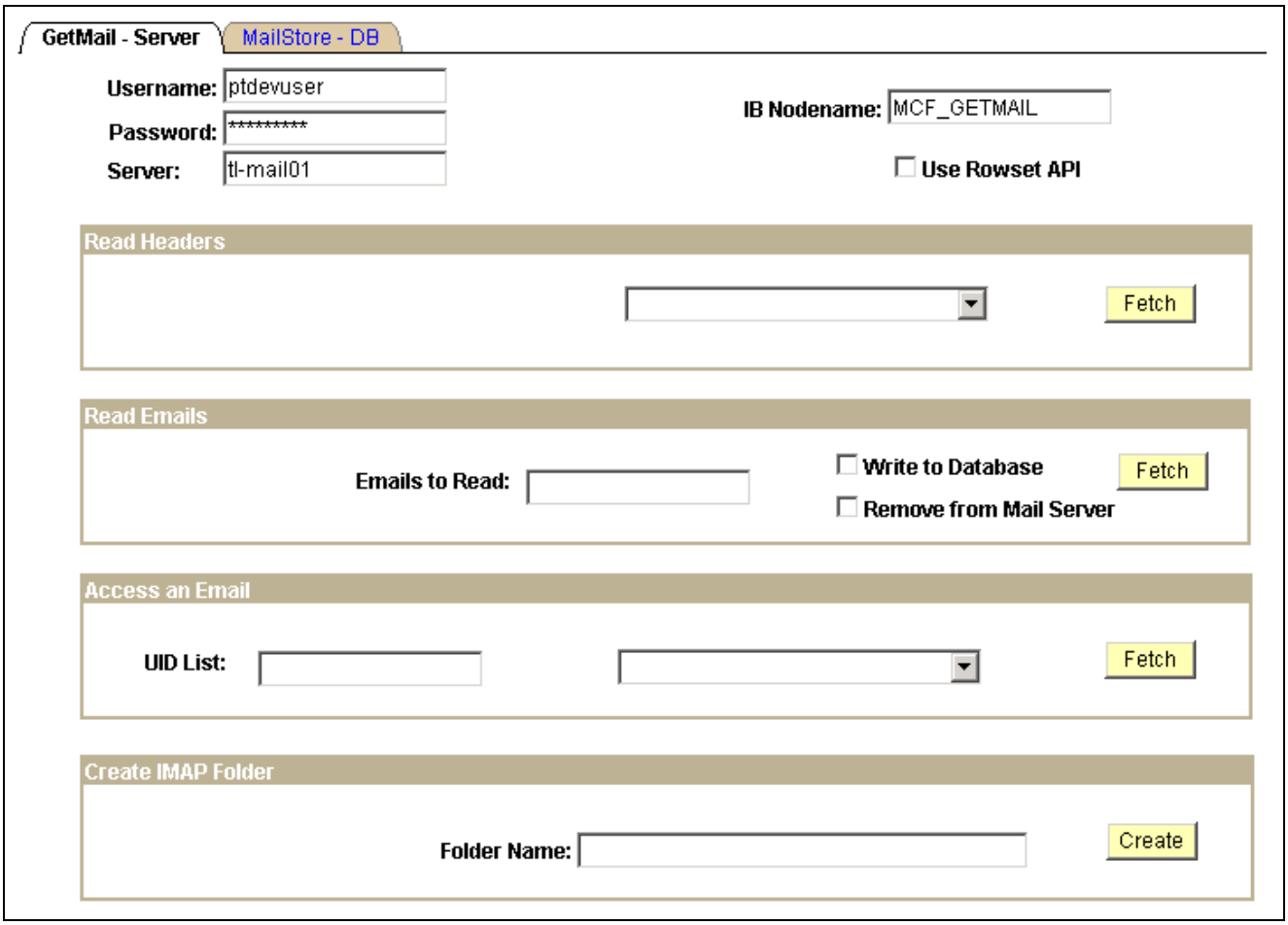
Pages Used to Demonstrate the Email Channel

Page Name	Object Name	Navigation	Usage
GetMail - Server	MCFGETMAIL_PG	PeopleTools, MultiChannel Framework, Email, Sample Pages, GetMail - Server	Demonstrate GetMail functionality.
MailStore - DB	MCFMAILSTORE_PG	PeopleTools, MultiChannel Framework, Email, Samples Pages, MailStore - DB	Demonstrate MailStore functionality.

Using the GetMail - Server Page

Access the GetMail - Server page.

 Copyright © 1988-2007, Oracle. All rights reserved.



GetMail - Server page

This page demonstrates reading or deleting emails from the mail server. It also demonstrates the storage and retrieval of emails to and from the database tables. Output of the response to each of the requests is written to a file at %PS_SERVDIR%/files/mcfdata.out on the application server. Check this file after each test for the output data.

To demonstrate email functionality, send an email to the user name and server that you enter on the GetMail - Server page.

Username and **Password**	Enter a valid username and password for an account on the mail server.
Server	Enter the mail server.
IB Nodename (integration broker node name)	Enter the PeopleSoft Integration Broker node used to access email from the mail server. The default is MCF_GETMAIL
Use Rowset API	Select to use the rowset-based GetMail API. Deselect to use the new Mail Classes API.

Read Headers

Select a method to be called from the drop-down list box, then click Fetch. Select from:

Copyright © 1988-2007, Oracle. All rights reserved.

Message Count: Writes the number of emails in the specified mailbox to the output file mcfdata.out.

ReadHeadersWithAttach: Writes headers and attachment information for all emails in the specified mailbox to the output file mcfdata.out.

Read Emails

Enter the number of emails to read, select other parameters, then click Fetch to read emails.

Emails to Read Enter the number of emails to retrieve from the specified mailbox and write to the output file mcfdata.out.

Write to Database Select to have the retrieved emails written to the database as well as to the output file mcfdata.out.

Remove from Mail Server Select to delete emails from the mail server after they have been retrieved .

Access an Email

Enter an email UID list, select a method from the drop-down list box, then click Fetch to access the selected emails.

UID List (unique identifier list) Enter the unique ID of an email to be retrieved or deleted. To delete, you can enter a comma-delimited list of unique email IDs (UID) to be deleted.

Each email has a unique number (the UID) associated with it that is permanently guaranteed not to refer to any other email in the mailbox. To find the UID of an email, run the *ReadHeadersWith Attach* option first, which returns the UID for each email. If an invalid UID is specified, one empty row is returned.

Read Email w/ attachment Retrieves the specified email from the specified mailbox and writes it to the output file mcfdata.out.

Remove Message Deletes the specified email from the specified mailbox.

Create IMAP Folder

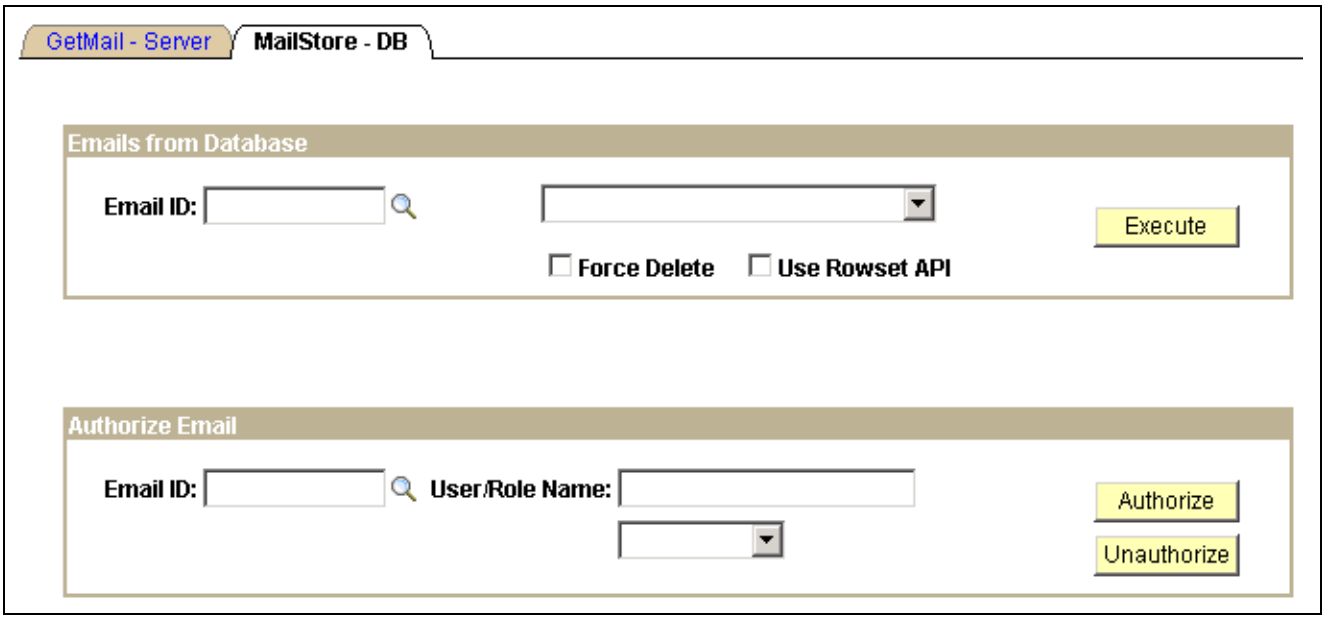
Create an email folder; only valid for IMAP4 mail servers.

Folder Name Enter the name of the folder to create.

Using the MailStore - DB Page

Access the MailStore - DB page.

Copyright © 1988-2007, Oracle. All rights reserved.



MailStore - DB page

View an example of how to access emails from the database (how to retrieve and delete email) and how to authorize email attachments for viewing or deletion using the PT_MCF_MAIL application package provided with PeopleSoft MultiChannel Framework.

Email ID Enter the PeopleSoft email ID (not to be confused with the UID) with which the email was saved to the database.

Emails from Database

Enter an email ID, select an action from the drop-down list box, and then click Execute to perform the selected action.

Select from:

Delete Email from Database Deletes the specified email from the database and any corresponding attachments from the repository.

Retrieve Email Retrieves the specified email from the database and writes it to the output file mcfdata.out.

Force Delete Select after entering *Delete Email from Database* to force the deletion even if there is an error deleting any associated attachments from the repository.

Use Rowset API Select to use the rowset-based GetMail API.

Deselect to use the new Mail Classes API.

Authorize Email

Enter an email ID, a user role or name, and click Authorize or Unauthorize to perform the specified action.

Email ID Enter the PeopleSoft email ID (not to be confused with the UID) with which the email was saved to the database.

User/Role Name Enter a PeopleSoft user ID or role to be authorized or unauthorized to view the attachment associated with the selected email.

Copyright © 1988-2007, Oracle. All rights reserved.

After entering an ID or role, select *User* or *Role,* as appropriate, from the drop-down list box.

See Also

Chapter 11, “Using PeopleSoft MCF Broadcast and Working with Sample Pages,” Using the Email Sample Page, page 153

 Copyright © 1988-2007, Oracle. All rights reserved.

CHAPTER 15

Configuring Instant Messaging

This chapter provides an overview of instant messaging and discusses how to:

- Configure instant messaging in PeopleSoft MultiChannel Framework.
- Use the instant messaging sample pages.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference, "MCFIMInfo Class"

Understanding Instant Messaging

PeopleSoft MCF extends the HTML chat functionality delivered in PeopleTools 8.42 to popular instant messaging networks. Users can initiate an outbound chat with a customer using instant messaging networks such as AOL Instant Messenger, IBM/Lotus Sametime, or Yahoo! Messenger. This enables an external user to participate in a business transaction through instant messaging.

PeopleSoft MultiChannel Framework enables instant messaging through public networks (AOL Instant Messenger and Yahoo! Messenger) and through an enterprise Lotus Sametime Connect network.

You can enable a user to initiate an instant messaging session from any page using either a button or PeopleCode.

The PeopleSoft system does not supply the client for any supported instant messaging software. Each user must install the appropriate client software. However, presence detection for AOL Instant Messenger and Yahoo! Messenger users functions even without installed client software.

Note. The instant messaging presence detection integration in PeopleTools (the "IM Integration") is subject to the respective IM vendor's network availability. In addition, the IM Integration is based on protocols made available by the respective IM Network vendors. Such protocols are subject to change at any time by the vendor. Therefore, PeopleSoft is not responsible for the availability or performance of any of the IM Networks or for any change in an IM vendor's protocol which may render the IM Integration inoperable.

Configuring Instant Messaging Servers

The supported public instant messaging networks, Yahoo! Messenger and AOL Instant Messenger, require installation of client software, but no server configuration. Installation and configuration of the client software is the responsibility of your administrators and customers, and is beyond the scope of this document.

Copyright © 1988-2007, Oracle. All rights reserved.

Lotus Sametime Connect is an instant messaging network running on an enterprise Sametime/Domino server, which must be configured to support PeopleSoft MultiChannel Framework instant messaging. To administer instant messaging with Sametime Connect, install Sametime Links and configure the server to enable anonymous users to connect to the server and query for the presence of other known users. Refer to your Lotus/Domino server documentation for more information.

Configuring Instant Messaging

To configure instant messaging, use the Instant Messaging Configuration (MCF_IM_CFG_CMP) component.

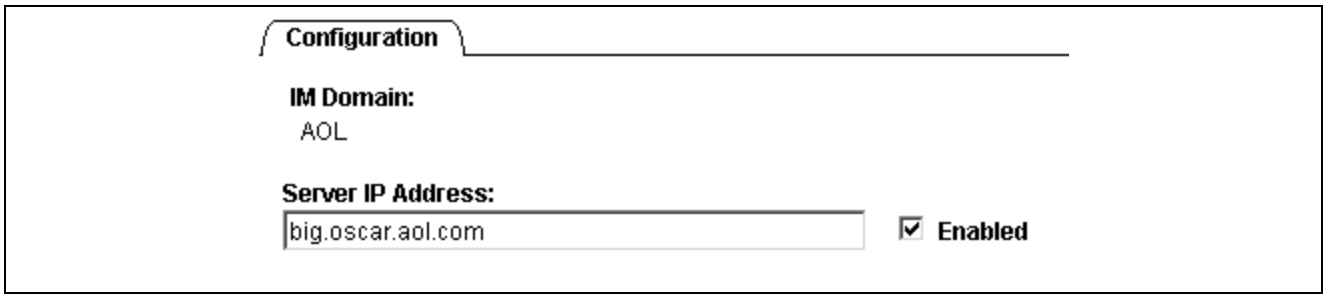
This section discusses how to configure instant messaging.

Page Used to Configure Instant Messaging

Page Name	Object Name	Navigation	Usage
Configuration	MCF_IM_CFG_PG	PeopleTools, MultiChannel Framework, Instant Messaging, Configuration	Enter or edit server IP address for instant messaging domains.

Configuring Instant Messaging

Configure instant messaging by enabling domains.



Configuration page

IM Domain The instant messaging service name.

Server IP Address The host name and port (if applicable) of the instant messaging service.

Warning! Do not alter the server address for Yahoo! or AOL.

The server addresses for Yahoo and AOL are preconfigured. However, enter the address for your Sametime Connect server.

Enabled Select Enabled to enable the configured instant messaging service.

If the enabled flag is unselected the network is still available, but the buttons will always show presence as offline.

 Copyright © 1988-2007, Oracle. All rights reserved.

Using the Instant Messaging Sample Pages

To demonstrate instant messaging, use the Instant Messaging Sample Pages (MCF_IM_DEMO_CMP) component.

This section discusses how to:

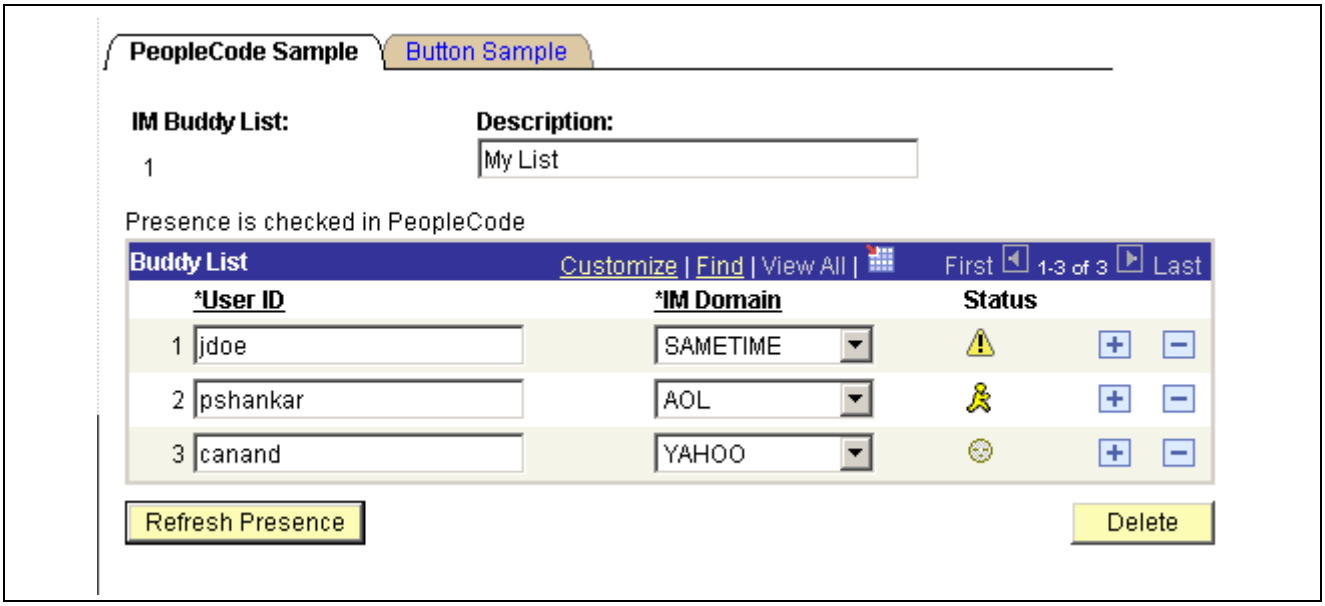
- Use the PeopleCode Sample page.
- Use the Button Sample page.

Pages Used to Test Instant Messaging

Page Name	Object Name	Navigation	Usage
PeopleCode Sample	MCF_IM_DEMO_PG	PeopleTools, MultiChannel Framework, Instant Messaging, Sample Pages, PeopleCode Sample	Set up buddy lists and view presence for configured instant messaging clients using PeopleCode.
Button Sample	MCF_IM_DEMO1_PG	PeopleTools, MultiChannel Framework, Instant Messaging, Sample Pages, Button Sample	View presence for configured instant messaging clients using the instant messaging button.

Using the PeopleCode Sample Page

Access the PeopleCode Sample page.



PeopleCode Sample page

The PeopleCode Sample page demonstrates the use of instant messaging PeopleCode. The sample pages are intended for demonstration purposes, and should not be used in production.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode API Reference*, "MCFIMInfo Class".

Copyright © 1988-2007, Oracle. All rights reserved.

IM Buddy List Displays an identifier for the list.

Description Enter a description for the list.

User ID Enter the user Id.

The user ID must be that of a valid user of the instant messaging service specified in the IM Domain field.

IM Domain Select the instant messaging service associated with the specified user ID from the drop-down list box.

Status An icon displays the presence status of the specified user. The icons vary with the instant messaging service.

Click the icon to initiate an instant messaging session with the selected user. The user must be online.

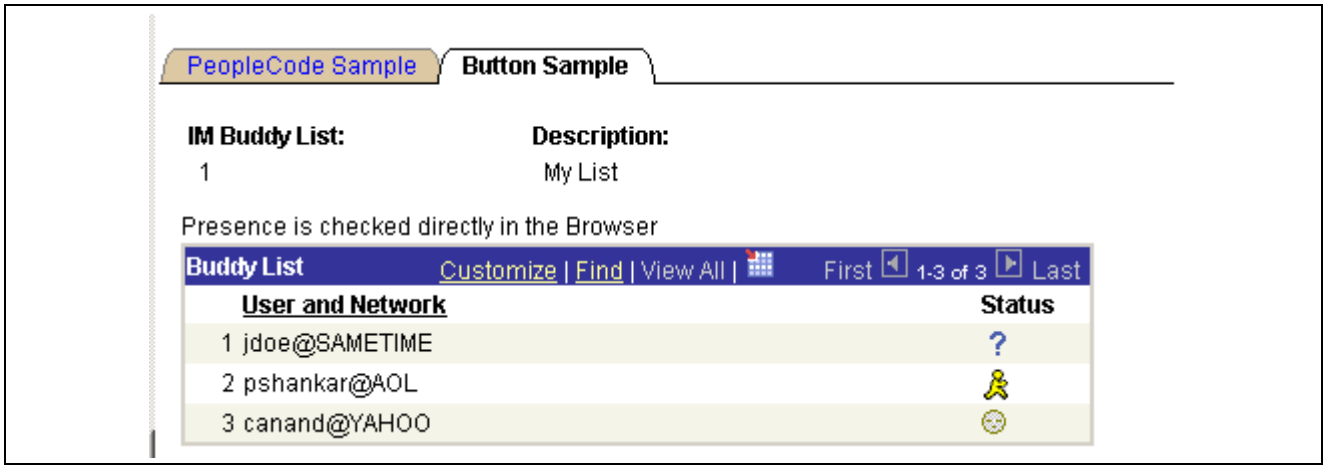
Note. Yahoo! and AOL do not update presence automatically. Use the Refresh Presence button to check the current status of Yahoo! and AOL users.

Refresh Presence Click the Refresh Presence button to check the status of AOL and Yahoo users.

Note. PeopleCode does not support presence detection in Sametime, so the Sametime user status does not display on the PeopleCode Sample page. Use the instant messaging button if you intend to check presence of or connect to Sametime users.

Using the Button Sample Page

Access the Button Sample page.



Button Sample page

The Button Sample page demonstrates the use of the instant messaging button implemented in PeopleSoft Application Designer. The sample pages are intended for demonstration purposes, and should not be used in production.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Application Designer*, "Using Page Controls," Specifying Destination Types.

IM Buddy List Displays an identifier for this list.

Copyright © 1988-2007, Oracle. All rights reserved.

Description Enter a description for this list.

User and Network Displays user ID and instant messaging service ID.

Status Displays presence status of the specified user.

Click the icon to initiate an instant messaging session with the selected user. You can send an instant message even if the user is not online.

Note. For SAMETIME icon to be visible, do not select the Sun JVM in the browser, else the icon does not display in button sample page.

Note. If the AOL icon does not display correctly, be sure that "Use HTTP 1.1" is selected in Microsoft Internet Explorer advanced internet options. If you must select "Use HTTP 1.1," be sure to close the current browser session and open another before continuing.

Copyright © 1988-2007, Oracle. All rights reserved.

 Copyright © 1988-2007, Oracle. All rights reserved.

APPENDIX A

JSMCAPI Quick Reference

This appendix discusses the constructors, fields, methods, callback event methods, and returns of the JSMCAPI classes.

JSMCAPI Classes

This section lists the JSMCAPI classes in alphabetical order, along with the constructor, fields, methods, and callback event methods, and returns associated with each class.

_Address

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
_Address()	_Address object

Fields

Field	Type
caps	Object
id	String

Callback Event Methods

Callback Event Methods	Returns
onError(*event*)	None

_UQAddress

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The _UQAddress class extends the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
_UQAddress()	_UQAddress object

Fields

Field	Type
tasks	Associative array

Methods

Method	Returns
acceptTask(*task*, *reason*)	None
dequeueTask(*task*)	Object

Callback Event Methods

The _UQAddress class inherits the onError callback event method from the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Callback Event Methods	Returns
onAccepted(*event*)	None
onAcceptingTask(*event*)	None
onDequeueingTask(*event*)	None
onNotify(*event*)	None
onTaskAdded(*event*)	None
onTaskRemoved(*event*)	None
onUnassigned(*event*)	None

_User

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
_User()	_User object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Field	Type
agentId	String
caps	Associative array
id	String
name	String
presences	Associative array
states	Associative array of the constants beginning with ST_*.
ST_LOGGEDIN	String
ST_LOGGEDOUT	String
ST_NOTREADY	String
ST_READY	String
ST_UNKNOWN	String
ST_WORKNOTREADY	String
ST_WORKREADY	String
statistics	Object
statistics1	Object
statistics2	Object

A2AChat

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The A2AChat class extends the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Constructor

Constructor	Returns
A2AChat(*event*, *address*, *chatType*)	A2AChat object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Field	Type
address	Object
agentId	String
agentName	String
appData	Object
chatType	String
customerName	Object
id	String
isConference	Object
jr	String of the PS_JR constant.
type	String of the following constants: • TYPE_ANSWER • TYPE_CONSULT
uniqueueId	String

Method

Method	Returns
getUrl(*defaultUrl*)	String

A2AChatAddress

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
A2AChatAddress()	A2AChatAddress object

Fields

The A2AChatAddress class inherits the following fields from the _Address class:

- id
- caps

Copyright © 1988-2007, Oracle. All rights reserved.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Field	Type
id	String
tasks	Associative array

Methods

Method	Returns
initiateChat(*agentId*)	None

Callback Event Methods

The A2AChatAddress class inherits the onError callback event method from the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Callback Event Methods	Returns
onChatEnded(*event*)	None
onInitiatingChat(*event*)	None
onNotify(*event*)	None

AgentStatistics

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
AgentStatistics()	AgentStatistics object

Fields

Field	Type
averageCallDuration	Number
averageHoldDuration	Number
callsHandled	Number
data	Associative array
percentIdleTime	Number
percentTimeAvailable	Number

Copyright © 1988-2007, Oracle. All rights reserved.

Field	Type
percentTimeInCurrentState	Number
percentTimeUnavailable	Number
taskAcceptedCurrentLogin	Number
timeCurrentLogin	Number
timeWorking	Number
totalTaskAcceptedLogin	Number
totalTaskDoneLogin	Number
totalTaskDoneLogin	Number
unavailableDuration	Number
waitDuration	Number

AppData

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
AppData()	AppData object

Fields

Field	Type
data	Associative array
groupId	String
jr	String
question	String
strData	String
subject	String
uniqueId	String
url	String of the constant URL

Copyright © 1988-2007, Oracle. All rights reserved.

Field	Type
userId	String
username	String
wizUrl	String

Method

Method	Returns
addKeyValue(*key*, *value*)	None

Buddy

This section lists the constructor, methods, and callback event methods associated with this class.

The Buddy class extends the _User class.

See Appendix A, "JSMCAPI Quick Reference," _User, page 422.

Constructor

Constructor	Returns
Buddy()	Buddy object

Fields

The Buddy class inherits the following fields from the _User class:

- agentID
- caps
- id
- name
- presences
- states
- statistics
- statistics1
- statistics2

See Appendix A, "JSMCAPI Quick Reference," _User, page 422.

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods

Callback Event Methods	Returns
onStat1(*event*)	None
onStat2(*event*)	None
onState(*event*)	None

Call

This section lists the constructor, fields, and returns (if applicable) associated with this class.

The Call class extends the Task class.

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Constructor

Constructor	Returns
Call(*strCall*)	Call object

Fields

The Call class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority
- type
- urlAbs
- urlRel

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Field	Type
line	Object
statistics	Object
userData	String

Copyright © 1988-2007, Oracle. All rights reserved.

CallData

This section lists the constructor, fields, methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
CallData()	CallData object

Fields

Field	Type
ani	String
callId	String
callType	String
data	String
dnis	String

Method

Method	Returns
addKeyValue(*key* , *value*)	None

CallStatistics

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
CallStatistics()	CallStatistics object

Fields

Field	Type
data	Associative array
holdTime	String
queueTime	String
talkTime	String

Copyright © 1988-2007, Oracle. All rights reserved.

Chat

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

The Chat class extends the Task class.

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Constructor

Constructor	Returns
Chat(*event*, *address*)	Chat object

Fields

The Chat class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority
- type
- urlAbs
- urlRel

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Field	Type
address	Object
agentId	String
appData	Object
chatconnection	Object
chatType	String
customerName	String
groupId	String
question	String
statistics	Object

Copyright © 1988-2007, Oracle. All rights reserved.

Field	Type
subject	String
userData	Object

Method

Method	Returns
gettpUrl(*defaultUrl*)	String
getUrl(*defaultUrl*)	String

ChatAddress

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The ChatAddress class extends the _UQAddress class.

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Constructor

Constructor	Returns
ChatAddress()	ChatAddress object

Fields

The ChatAddress class inherits the following fields from the _Address class:

- id
- caps

Fields	Returns
chatconnections	Object
tasks	Object

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

The ChatAddress class inherits the tasks field from the _UQAddress class.

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Methods

The ChatAddress class inherits the following methods from the _UQAddress class:

- acceptTask
- dequeueTask

Copyright © 1988-2007, Oracle. All rights reserved.

Methods	Returns
chat (*agentId*, *buddyId*, *reason*)	None
getChatconnectionByConnectionId (*connectionId*)	Object
getChatconnectionindexByConnectionId (*connectionId*)	Index
getFreeChatconnection	Object
getFreeChatconnectionIndex	Object

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Callback Event Methods

The ChatAddress class inherits the onError callback event method from the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

The ChatAddress class inherits the following callback event methods from the _UQAddress class:

- onAccepted
- onAcceptingTask
- onDequeueingTask
- onNotify
- onTaskAdded
- onTaskRemoved
- onUnassigned

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Callback Event Methods	Returns
onCapabilitiesChanged (*event*)	None

ChatConnection

This section lists the constructor, fields, method, callback event methods and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
ChatConnection(*event*)	ChatConnection object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Fields	Returns
caps	Object
chat	String
connectionId	String
id	String
state	String

Method

Methods	Returns
answer (*agentid*, *reason*)	None
conference (*agentid*, *reason*)	None
forward (*fromagentid*, *toagentid*, *qid*, *userdata*, *chatdata*, *reason*)	None
gethistory (*agentid*, *noofLines*, *reason*)	None
getUrl (*deafultURL*)	String
message (*agentid*, *message*, *reason*)	None
pushURL(*URL*)	None
reject(*agentid*, *reason*)	None
release(*agentid*, *reason*)	None
typing (*agentid*, *reason*)	None

Callback Event Methods

Callback Event Methods	Rreturns
onAccepted (*event*)	None
onAnswering (*event*)	None
onCapabilitiesChanged (*event*)	None
onChatdataChanged (*event*)	None
onConferencing (*event*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods	Rreturns
onDialing (*event*)	None
onDropped (*event*)	None
onError (*event*)	None
onForwarded (*event*)	None
onForwardError (*event*)	None
onForwarding (*event*)	None
onHistory (*event*)	None
onIncomingChat (*event*)	None
onMessage (*event*)	None
onPartyAdded (*event*)	None
onPartyChanged (*event*)	None
onPartyRemoved (*event*)	None
onProperties (*event*)	None
onPushURL (*event*)	None
onRejected (*event*)	None
onReleased (*event*)	None
onReleasing (*event*)	None
onRevoked (*event*)	None
onTalking (*event*)	None
onTyping (*event*)	None
onUserdataChanged (*event*)	None

ChatConnectionCaps

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
ChatConnectionCaps(*event*)	ChatConnectionCaps object

Fields

Fields	Returns
canAnswer	Boolean
canConference	Boolean
canConferenceSingle	Boolean
canForward	Boolean
canIndicateTyping	Boolean
canPushURL	Boolean
canReject	Boolean
canSendMessage	Boolean

ChatData

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
ChatData(*event*)	ChatData object

Fields

Fields	Returns
data	Object

Method

Methods	Returns
addKeyValue(*key*, *value*)	None

Connection

This section lists the constructor methods, and returns (if applicable) associated with this class.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
Connection (*event*)	ConnectionEvent object

Methods

Methods	Returns
eventUserDataChanged(*event*)	None

ConnectionListener

This section lists the constructor methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
ConnectionListener (*event*)	ConnectionListener object

Methods

Methods	Returns
requestAttachUserdata(*object*, *request*)	None

ConnectionRequest

This section lists the constructor method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
ConnectionRequest (*event*)	ConnectionRequest object

Methods

Methods	Returns
getUserData( )	UserData

Email

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

The Email class extends the Task class.

See Appendix A, “JSMCAPI Quick Reference,” Task, page 463.

 Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
Email(*event*)	Email object

Fields

The Email class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority
- type
- urlAbs
- urlRel

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Field	Type
address	Object
agentId	Object
appData	Object
customerName	String
emailconnection	Object
emailId	String
groupId	String
question	String
statistics	Object
subject	String
userData	Object

Copyright © 1988-2007, Oracle. All rights reserved.

Method

Method	Returns
gettpUrl(*defaultUrl*)	Object
getUrl(*defaultUrl*)	String

EmailAddress

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The EmailAddress class extends the _UQAddress class.

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Constructor

Constructor	Returns
EmailAddress()	EmailAddress object

Fields

The EmailAddress class inherits the following fields from the _Address class:

- id
- caps

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

The EmailAddress class inherits the tasks field from the _UQAddress class.

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Fields	Returns
agent	Object
emailconnections	Object

Methods

The EmailAddress class inherits the following methods from the _UQAddress class:

- acceptTask
- dequeueTask

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

 Copyright © 1988-2007, Oracle. All rights reserved.

Methods	Returns
getEmailconnectionByConnectionId (*connectionId*)	Object
getEmailconnectionindexByConnectionId (*connectionId*)	Index
getFreeEmailconnection	Object

Callback Event Methods

The EmailAddress class inherits the onError callback event method from the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

The EmailAddress class inherits the following callback event methods from the _UQAddress class:

- onAccepted
- onAcceptingTask
- onDequeueingTask
- onNotify
- onTaskAdded
- onTaskRemoved
- onUnassigned

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

EmailConnection

This section lists the constructor, fields, method, callback event methods and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
EmailConnection(*event*)	EmailConnection object

Fields

Fields	Returns
caps	Object
connectionId	String
email	String
id	String
state	String

Copyright © 1988-2007, Oracle. All rights reserved.

Method

Methods	Returns
abandon (*reason*)	None
answer (*reason*)	None
complete (*reason*)	None
forward (*fromagentid*, *toagentid*, *qid*, *userdata*, *Emaildata*, *reason*)	None
reject(*agentid*, *reason*)	None
withdraw_RES (*reason*)	None

Callback Event Methods

Callback Event Methods	Rreturns
onAnswering (*event*)	None
onCapabilitiesChanged (*event*)	None
onCompleted (*event*)	None
onDropped (*event*)	None
onEmaildataChanged (*event*)	None
onError (*event*)	None
onForwarded (*event*)	None
onForwardError (*event*)	None
onForwarding (*event*)	None
onIncoming (*event*)	None
onProcessing (*event*)	None
onRejected (*event*)	None
onRevoked (*event*)	None
onUserdataChanged (*event*)	None
onWithdraw_REQ (*event*)	None

EmailConnectionCaps

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
EmailConnectionCaps(*event*)	EmailConnectionCaps object

Fields

Fields	Returns
canAnswer	Boolean
canComplete	Boolean
canForward	Boolean
canReject	Boolean

EmailData

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
EmailData(*event*)	EmailData object

Fields

Fields	Returns
data	Object

Method

Methods	Returns
addKeyValue(*key*, *value*)	None

Extension

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The Extension class extends the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
Extension(*numOfLines*)	Extension object

Fields

The Extension class inherits the following fields from the _Address class:

- id
- caps

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

Field	Type
forwardMode	Object
isDnd	Boolean
lines	Associative array
numOfLines	Number

Methods

Method	Returns
cancelDnd(*reason*)	None
cancelForwardSet(*reason*)	None
forwardSet(*number*, *mode*, *reason*)	None
getDialingLine()	Object
getFreeLine()	Object
getLineByConnectionId(*connectionId*)	Object
getOffHookLine()	Object
setDnd(*reason*)	None

Callback Event Methods

The Extension class inherits the onError callback event method from the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

 Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods	Returns
onCancelingDnd(*event*)	None
onCancelingForward(*event*)	None
onDnd(*event*)	None
onDndCanceled(*event*)	None
onForwardCanceled(*event*)	None
onForwarded(*event*)	None
onForwarding(*event*)	None
onSettingDnd(*event*)	None

ExtensionCaps

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
ExtensionCaps(*strCaps*)	ExtensionCaps object

Fields

Field	Type
canCancelDnd	Boolean
canDial	Boolean
canFwdBusy	Boolean
canFwdBusyNoAnswer	Boolean
canFwdCancelForward	Boolean
canFwdDefault	Boolean
canFwdNoAnswer	Boolean
canFwdUnconditional	Boolean
canRefreshState	Boolean
canSetDnd	Boolean

Copyright © 1988-2007, Oracle. All rights reserved.

ForwardMode

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
ForwardMode()	ForwardMode object

Fields

Field	Type
mode	String of the following constants: • BUSY • BUSYNOANSWER • DEFAULT • NOANSWER • NONE • UNCONDITIONAL

GenericAddress

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The GenericAddress class extends the _UQAddress class.

See Appendix A, “JSMCAPI Quick Reference,” _UQAddress, page 421.

Constructor

Constructor	Returns
GenericAddress()	GenericAddress object

Fields

The GenericAddress class inherits the following fields from the _Address class:

- id
- caps

See Appendix A, “JSMCAPI Quick Reference,” _Address, page 421.

The GenericAddress class inherits the tasks field from the _UQAddress class.

See Appendix A, “JSMCAPI Quick Reference,” _UQAddress, page 421.

Copyright © 1988-2007, Oracle. All rights reserved.

Fields	Returns
agent	Object
genericconnections	Object

Methods

The GenericAddress class inherits the following methods from the _UQAddress class:

- acceptTask
- dequeueTask

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

Methods	Returns
getGenericconnectionByConnectionId (*connectionId*)	Object
getGenericconnectionindexByConnectionId (*connectionId*)	Index
getFreeGenericconnection	Object

Callback Event Methods

The GenericAddress class inherits the onError callback event method from the _Address class.

See Appendix A, "JSMCAPI Quick Reference," _Address, page 421.

The GenericAddress class inherits the following callback event methods from the _UQAddress class:

- onAccepted
- onAcceptingTask
- onDequeueingTask
- onNotify
- onTaskAdded
- onTaskRemoved
- onUnassigned

See Appendix A, "JSMCAPI Quick Reference," _UQAddress, page 421.

GenericConnection

This section lists the constructor, fields, method, callback event methods and returns (if applicable) associated with this class.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
GenericConnection(*event*)	GenericConnection object

Fields

Fields	Returns
caps	Object
connectionId	String
generic	String
id	String
state	String

Method

Methods	Returns
abandon (*reason*)	None
answer (*reason*)	None
complete (*reason*)	None
forward (*fromagentid*, *toagentid*, *qid*, *userdata*, *Genericdata*, *reason*)	None
reject(*agentid*, *reason*)	None
withdraw_RES (*reason*)	None

Callback Event Methods

Callback Event Methods	Rreturns
onCapabilitiesChanged (*event*)	None
onCompleted (*event*)	None
onDropped (*event*)	None
onError (*event*)	None
onForwarded (*event*)	None
onForwardError (*event*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods	Rreturns
onForwarding (*event*)	None
onGenericdataChanged (*event*)	None
onIncoming (*event*)	None
onProcessing (*event*)	None
onRejected (*event*)	None
onRevoked (*event*)	None
onUserdataChanged (*event*)	None
onWithdraw_REQ (*event*)	

GenericConnectionCaps

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
GenericConnectionCaps(*event*)	GenericConnectionCaps object

Fields

Fields	Returns
canAnswer	Boolean
canComplete	Boolean
canForward	Boolean
canReject	Boolean

GenericData

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
GenericData(*event*)	GenericData object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Fields	Returns
data	Object

Method

Methods	Returns
addKeyValue(*key*, *value*)	None

GenericTask

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The GenericTask class extends the Task class.

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Constructor

Constructor	Returns
GenericTask(*event*)	GenericTask object

Field

The GenericTask class inherits the following fields from the Task class:

- caseid
- cost
- customerid
- group
- id
- onStat
- priority
- type
- urlAbs
- urlRel

See Appendix A, "JSMCAPI Quick Reference," Task, page 463.

Field	Type
address	String
agentId	String

Copyright © 1988-2007, Oracle. All rights reserved.

Field	Type
appData	Object
customerName	String
genericconnection	Object
genericId	String
groupId	String
question	String
subject	String
userData	Object

Method

Method	Returns
gettpUrl(*defaultUrl*)	Object
getUrl(*defaultUrl*)	String

GLOBALS

This section lists the fields and methods associated with this class.

Fields

Field	Field Type
A2AChat.PS_JR	Constant
A2AChat.TYPE_ANSWER	Constant
A2AChat.TYPE_CONSULT	Constant
Server.TYPE.CTI	Constant
Server.TYPE_UQ	Constant
Task.TYPE_A2ACHAT	Constant
Task.TYPE_CHAT	Constant
Task.TYPE_CTI	Constant
Task.TYPE_EMAIL	Constant
Task.TYPE_GENERIC	Constant

Copyright © 1988-2007, Oracle. All rights reserved.

Methods

Method	Returns
initJSMCAPI()	None
isValid(*obj*)	Boolean
MCFBroadcast(*cluster*, *cluster*, *cluster*, *cluster*, *cluster*, *cluster*, *cluster*, *cluster*, *cluster*, *cluster*)	None

Group

This section lists the constructor, fields, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
Group()	Group object

Fields

Field	Type
id	String
name	String
registered	Boolean
statistics	Object
statistics1	Object
statistics2	Object

Callback Event Methods

Callback Event Methods	Returns
onStat(*event*)	None
onStat1(*event*)	None
onStat2(*event*)	None
onTaskAdded(*event*)	None
onTaskRemoved(*event*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

GroupStatistics

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
GroupStatistics()	GroupStatistics object

Fields

Field	Type
data	Associative array
listOfTasksInTheQueueByTaskType	Number
maxTaskCompletionTime	Number
newestTask	Number
newestTaskCompletionTime	Number
numberOfAbandoned	String
numberOfLoggedIn	String
numberOfQueued	String
numUnassignedTasks	Number
queuedWaitTime	String
queueUpTime	Number
relativeQueueLoad	String
timeElapsedOldestTask	

GroupStatistics1

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
GroupStatistics1(*data*)	GroupStatistics1 object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Field	Type
mostRecentTaskDone	String
mostRecentTaskDoneData	String
mostRecentTaskEnqueued	String
mostRecentTaskEnqueuedData	String
numAgentsAvailable	String
numAgentsLoggedIn	String
numEscalation	String
numOverflow	String
numTaskAccepted	String
numTaskDone	String
numTaskQueued	String
reasonFlag	String
taskTotalTimeInSystem	String
timeSinceStart	String

GroupStatistics2

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
GroupStatistics2(*data*)	GroupStatistics2 object

Fields

Field	Type
averageTaskDuration	String
averageWaitTime	String
oldestTask	String
recentTask	String

Copyright © 1988-2007, Oracle. All rights reserved.

Field	Type
timeElapsedOldestTask	String
timeElapsedRecentTask	String

Line

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
Line()	Line object

Fields

Field	Type
call	Object
caps	Object
connectionId	String
id	String
isMuted	
state	String of the following constants: • ST_DIALING • ST_DROPPED • ST_HELD • ST_IDLE • ST_OFFHOOK • ST_RINGING • ST_TALKING

Methods

Method	Returns
alternate(*reason*)	None
answer(*reason*)	None
attachUserData(*userdata*, *reason*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Method	Returns
clear(*reason*)	None
complete(*reason*)	None
conference(*destination*, *reason*, *userdata*, *calldata*)	None
conferenceSingle(*destination*, *reason*, *userdata*, *calldata*)	None
dial(*number*, *reason*, *userdata*)	None
dropParty(*destination*, *reason*)	None
getAni()	String
getDescr()	String
getDnis()	String
getReferenceId()	String
getUrl(*defaultUrl*)	String
grabCall(*destination*, *reason*)	String
hold(*reason*)	String
join (*reason*, *conferenceId*)	None
mute (*reason*)	None
park(*destination*, *reason*)	None
reconnect(*reason*)	None
reject(*reason*)	None
release(*reason*)	None
retrieve(*reason*)	None
sendDTMF(*reason*, stringDTMF)	None
setcallresult(*result*)	None
setcallresultDNC(*reason*, *number*)	None
setcallresultReschedule(*hours*, *minutes*,*day*, *month*,*year*)	None
transfer(*destination*, *reason*, *userdata*, *calldata*)	None
transferMute(*destination*, *reason*, *userdata*, *calldata*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Method	Returns
unmute(*reason*)	None
updateCallData(*calldata*, *reason*)	None

Callback Event Methods

Callback Event Methods	Returns
onAlternating(*event*)	None
onAnswering(*event*)	None
onAttachingUD(*event*)	None
onCallDataChanged(*event*)	None
onCapabilitiesChanged(*event*)	None
onClearing(*event*)	None
onCompleting(*event*)	None
onConferencing(*event*)	None
onDialing(*event*)	None
onDropped(*event*)	None
onError(*event*)	None
onGrabbing(*event*)	None
onHeld(*event*)	None
onHolding(*event*)	None
onJoining(*event*)	None
onMuted(*event*)	None
onOffHook(*event*)	None
onOnHook(*event*)	None
onParking(*event*)	None
onPartyAdded(*event*)	None
onPartyChanged(*event*)	None
onPartyRemoved(*event*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods	Returns
onReconnecting(*event*)	None
onRejected(*event*)	None
onRejecting(*event*)	None
onReleasing(*event*)	None
onRetrieving(*event*)	None
onRinging(*event*)	None
onSetcallresult(*event*)	
onSetcallresultDNC(*event*)	
onSetcallresultReschedule(*event*)	
onTalking(*event*)	None
onTransfering(*event*)	None
onUnmuted(*event*)	None
onUpdatingCD(*event*)	None
onUserDataChanged(*event*)	None

LineCaps

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
LineCaps(*strCaps*)	LineCaps object

Fields

Field	Type
canAlternate	Boolean
canAnswer	Boolean
canAttachUserData	Boolean
canClear	Boolean
canComplete	Boolean

Copyright © 1988-2007, Oracle. All rights reserved.

Field	Type
canConference	Boolean
canConferenceSingle	Boolean
canDropParty	Boolean
canHold	Boolean
canMute	Boolean
canPark	Boolean
canReconnect	Boolean
canReject	Boolean
canRelease	Boolean
canRetrieve	Boolean
canSendDTMF	Boolean
canSetcallresult	Boolean
canSetcallresultDNC	Boolean
canSetcallresultReschedule	Boolean
canTransfer	Boolean
canTransferMute	Boolean
canUnmute	Boolean
canUpdateCallData	Boolean

MCEvent

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
MCEvent()	MCEvent object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Field	Type
extension	Object
group	Object
reason	Object
user	Object

MediaType

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
MediaType()	MediaType object

Field

Field	Type
type	String of the following constants: • MT_EMAIL • MT_VOICE

PSMC

This section lists the constructor, fields, methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
PSMC()	PSMC object

Fields

Field	Type
renServer	Object
servers	Associative array
sessions	Associative array

Copyright © 1988-2007, Oracle. All rights reserved.

Methods

Method	Returns
closeSession(*serverId*)	None
getCallById(*id*)	Object
getChatById(*id*)	Object
getEmailById(*id*)	Object
getGenericTaskById(*id*)	Object
getLineById(*id*)	Object
openSession(*serverId*)	None
start()	None
stop()	None

Reason

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
Reason(*code*, *desc*)	Reason object

Fields

Field	Type
code	String
description	String
reasonData1	String
reasonData2	String
reasonData3	String

RenServer

This section lists the constructor, fields, and callback event methods, and returns (if applicable) associated with this class.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
RenServer()	RenServer object

Fields

Field	Type
isRunning	Boolean
url	String

Callback Event Methods

Callback Event Methods	Returns
onDown(*event*)	None
onUp(*event*)	None

Server

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
Server(*serverID*)	Server object

Fields

Field	Type
id	String
info	String
state	String of the following constants: • ST_INSERVICE • ST_OUTOFSERVICE
type	String of the following constants: • TYPE_CTI • TYPE_UQ

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods

Callback Event Methods	Returns
onBroadcast(*event*)	None
onHbLost(*event*)	None
onHbRecovered(*event*)	None
onInService(*event*)	None
onOutOfService(*event*)	None
onRestart(*event*)	None

Session

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
Session(*serverID*)	Session object

Fields

Field	Type
addresses	Associative array
buddies	Associative array
groups	Associative array
id	String
serverID	String
state	String of the following constants: • ST_ACTIVE • ST_CLOSED • ST_CLOSING • ST_IDLE
user	User object

Copyright © 1988-2007, Oracle. All rights reserved.

Methods

Method	Returns
broadcastSubscribe(*cluster*, *queue*, *task*, *state*, *presence*, *method*)	None
broadcastUnsubscribe(*type*)	None
close()	None
open()	None
registerAddress(*address*)	None
registerBuddy(*buddy*)	None
registerGroup(*group*)	None
registerUser(*user*)	None
setAutoRecovery(*autoRecover*)	None
statPublish(*userStat*, *userStat*)	None
unregisterAddress(*address*)	None
unregisterBuddy(*buddy*)	None
unregisterGroup(*group*)	None
unregisterUser(*user*)	None

Callback Event Methods

Callback Event Methods	Returns
onAddressRegistered(*event*)	None
onAddressUnregistered(*event*)	None
onBuddyRegistered(*event*)	None
onBuddyUnregistered(*event*)	None
onClosed(*event*)	None
onError(*event*)	None
onGroupRegistered(*event*)	None
onGroupUnregistered(*event*)	None
onInfo(*event*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods	Returns
onOpened(*event*)	None
onUserRegistered(*event*)	None
onUserUnregistered(*event*)	None

Task

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
Task()	Task object

Fields

Field	Type
caseid	String
cost	String
customerid	String
group	String
id	String
onStat	String
priority	String
type	String; one of the following: • TYPE_A2ACHAT • TYPE_CHAT • TYPE_CTI • TYPE_EMAIL • TYPE_GENERIC
urlAbs	Object
urlRel	Object

Copyright © 1988-2007, Oracle. All rights reserved.

Methods

Method	Returns
setUserData(*UserData data*)	None
getUserData( )	UserData

TaskStatistics

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
TaskStatistics()	TaskStatistics object

Fields

Field	Type
data	Associative array
holdTime	String
queueTime	String
talkTime	String

User

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

The User class extends the _User class.

See Appendix A, "JSMCAPI Quick Reference," _User, page 422.

Constructor

Constructor	Returns
User(*id*, *name*, *agentId*, *agentPassword*)	User object

Fields

The User class inherits the following fields from the _User class:

- agentId
- caps
- id

Copyright © 1988-2007, Oracle. All rights reserved.

- name
- presences
- states
- statistics
- statistics1
- statistics2

See Appendix A, "JSMCAPI Quick Reference," _User, page 422.

Field	Type
addresses	Associative array
agentPassword	String
language	String
registeredAddresses	Associative array
statesUq	Object

Methods

Method	Returns
disableMedia(*group*, *mediaType*, *extension*, *reason*)	None
enableMedia(*group*, *mediaType*, *extension*, *reason*)	None
isMediaEnabled(*mediaType group*)	None
login(*group*, *extension*, *reason*)	None
loginUq(*group*)	None
logout(*group*, *extension*, *reason*)	None
logoutUq(*group*)	None
register(*extension*, *reason*)	None
setNotReady(*group*, *presence*, *extension*, *reason*)	None
setPresence(*group*, *presence*, *extension*, *reason*)	None
setPresenceUq(*group*, *presenceText*, *reason*)	None
setReady(*group*, *presence*, *extension*, *reason*)	None
setWorkNotReady(*group*, *presence*, *extension*, *reason*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Method	Returns
setWorkReady(*group*, *presence*, *extension*, *reason*)	None
unRegister(*extension*, *reason*)	None

Callback Event Methods

Callback Event Methods	Returns
onCapabilitiesChanged(*event*)	None
onCtiBusy(*event*)	None
onCtiClear(*event*)	None
onDropped(*event*)	None
onError(*event*)	None
onInfo(*event*)	None
onLoggedIn(*event*)	None
onLoggedOut(*event*)	None
onLoggingIn(*event*)	None
onLoggingOut(*event*)	None
onMediaDisabled(*event*)	None
onMediaEnabled(*event*)	None
onNotReady(*event*)	None
onPresenceChanged(*event*)	None
onReady(*event*)	None
onRegistered(*event*)	None
onRegistering(*event*)	None
onSettingNotReady(*event*)	None
onSettingPresence(*event*)	None
onSettingReady(*event*)	None
onSettingWorkNotReady(*event*)	None
onSettingWorkReady(*event*)	None

Copyright © 1988-2007, Oracle. All rights reserved.

Callback Event Methods	Returns
onStat(*event*)	None
onStat1(*event*)	None
onStat2(*event*)	None
onUnkown(*event*)	None
onUnregistered(*event*)	None
onUnregistering(*event*)	None
onWorkNotReady(*event*)	None
onWorkReady(*event*)	None

UserCaps

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
UserCaps()	UserCaps object

Fields

Fields	Returns
canHandleChat	Boolean
canHandleEmail	Boolean
canHandleGeneric	Boolean
canHandleVoice	Boolean
canLogin	Boolean
canLogout	Boolean
canRefreshState	Boolean
canSetNotReady	Boolean
canSetPresence	Boolean
canSetReady	Boolean

Copyright © 1988-2007, Oracle. All rights reserved.

Fields	Returns
canSetWorkNotReady	Boolean
canSetWorkReady	Boolean

UserData

This section lists the constructor, fields, method, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
UserData()	UserData object

Constants

The UserData class uses the following constants:

- COMPONENT
- DESCR
- ISCRIPTPROGRAMNAME
- ICTYPE
- MARKET
- MENU
- REFERENCEID
- TARGET
- URLABS
- URLREL

Method

Method	Returns
add(*key*, *value*)	None
addKeyValue(*key*, *value*)	None
getValue(String key)	String
remove(*key*)	None
getKeys( )	String array

UserStatistics1

This section lists the constructor, fields, and returns (if applicable) associated with this class.

Copyright © 1988-2007, Oracle. All rights reserved.

Constructor

Constructor	Returns
UserStatistics1()	UserStatistics1 object

Fields

Field	Type
availableCost	String
ctiBusy	String
currentQueue	String
mostRecentTaskData	String
mostRecentTaskId	String
numTaskAccepted	String
numTasksDone	String
numTasksUnassigned	String
presenceText	String
reasonFlag	String
state	String
timeInCurrentState	String
timeSinceLoggedIn	String

UserStatistics2

This section lists the constructor, fields, methods, and callback event methods, and returns (if applicable) associated with this class.

Constructor

Constructor	Returns
UserStatistics2()	UserStatistics2 object

Copyright © 1988-2007, Oracle. All rights reserved.

Fields

Field	Type
currentQueue	String
timeIdle	String
timeInCurrentState	String
timeNotReady	String
timeSinceLogin	String
totalTimeAvailable	String
totalTimeUnavailable	String

Copyright © 1988-2007, Oracle. All rights reserved.

APPENDIX B

Installing Digital Certificates for REN SSL

Digital certificates are required to provide client and server authentication for REN SSL. A digital certificate is an electronic means of establishing your credentials for web or business transactions that are issued by a certification authority (CA). The CA is a trusted third party who signs and issues the certificates for users after verifying their authentication using secure means.

This appendix describes a sample way of installing digital certificates and configure REN SSL. PeopleSoft customers may have their own means of obtaining and installing digital certificates for REN SSL.

Installing Digital Certificates

This section outlines the basic steps to install digital certificates. Before installing digital certificates, you must create the application server domain.

Note. The application server domain must have write permissions. All certificates are stored under <PS_HOME>/appserv/<domain name>. The cacerts file has write permissions under <PS_HOME>/JRE/lib/security.

The following overview lists the steps that are required to install digital certificates. The following sections describe each step in detail.

To install digital certificates and configure REN SSL:

1. Install the CA server certificate.

   See Appendix B, "Installing Digital Certificates for REN SSL," Installing the CA Server Certificate, page 472.

2. Installing REN server certificate.

   See Appendix B, "Installing Digital Certificates for REN SSL," Installing REN Server Certificate, page 472.

3. Configure digital certificates.

   See Appendix B, "Installing Digital Certificates for REN SSL," Configuring Digital Certificates, page 472.

4. Import certificates in Java keystore.

   See Appendix B, "Installing Digital Certificates for REN SSL," Importing Certificates in Java Key Store, page 474.

5. Configure the REN server.

   See Appendix B, "Installing Digital Certificates for REN SSL," Configuring REN Server, page 474.

Copyright © 1988-2007, Oracle. All rights reserved.

6. Configure the REN clusters.

   See Appendix B, "Installing Digital Certificates for REN SSL," Configuring REN Clusters, page 474.

7. Install certificates for local node.

   See Appendix B, "Installing Digital Certificates for REN SSL," Installing Certificates for Local Node, page 475.

8. Generating client certificate

   See Appendix B, "Installing Digital Certificates for REN SSL," Generating Client Certificate, page 476.

9. Install PSMCAPI certificates.

   See Appendix B, "Installing Digital Certificates for REN SSL," Installing PSMCAPI Certificates, page 476.

Installing the CA Server Certificate

To install the CA server certificate:

1. Generate the RSA private key for certificate authority.
2. Generate the Certificate Signing Request (CSR) for certificate authority.
3. Generate the PEM file.

   Note. If there is already an existing CA certificate in PEM format, these three steps can be omitted.

4. Import CA Certificate in PEM format in PeopleTools, Security, Security Objects, Digital Certificates.

The above steps are explained in details in the section, Configuring Digital Certificates.

See Appendix B, "Installing Digital Certificates for REN SSL," Configuring Digital Certificates, page 472 and *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker*, "Setting Up Secure Integration Environments," Implementing Client Authentication.

Installing REN Server Certificate

To install REN server certificate:

1. Generate REN server Certificate Signing Request (CSR) using PeopleTools, Security, Security Objects, Digital Certificates.
2. Get the CSR signed by CA.

   Note. The certificate must be in PEM format.

3. Import the certificate in PeopleTools, Security, Security Objects, Digital Certificates.

The above steps are explained in details in the following section, Configuring Digital Certificates.

See Appendix B, "Installing Digital Certificates for REN SSL," Configuring Digital Certificates, page 472.

Configuring Digital Certificates

Before configuring digital certificates, you must generate the private keys, CSR, and PEM file.

 Copyright © 1988-2007, Oracle. All rights reserved.

To configure digital certificates:

1. Select PeopleTools, Security, Security Objects, Digital Certificates.
2. Click +.
3. Select *ROOTCA* from Type drop-down list.
4. Enter an alias name for the CA in Alias, and click Add Root.

   The Add Root Certificate dialog box appears.
5. Open the ca.pem file.

   The root CA certificate is generated.
6. Copy the contents of ca.pem file, paste them in the Add Root Certificate dialog box, and click OK.
7. Click +.
8. Select *Cert* from the Type drop-down list.
9. Enter an alias name in Alias, such as, PSFTCA. Click Add Root.
10. Select CA certificate alias of step 4 from the Issuer Alias lookup button.
11. Click Request.

    The Request New certificate dialog box appears.
12. Enter Common Name, Org Unit, Organization, Locality, State/Province, Country, Algorithm, Key Size, Email Address, and Challenge Pswd.

    Note. Common name must be the machine name of REN server's machine, for example PTA112.peoplesoft.com where PTA112 is the machine name and .peoplesoft.com is the domain name
13. Click OK.

    The Certificates Signing Request dialog box appears.
14. Copy and paste the text from the Certificates Signing Request dialog box, and save the text in a file named ren.csr in <PS_HOME>\appserv\<domain name>\.
15. Click OK.

    The Import link appears.
16. Submit ren.csr to the CA that issued the selected root certificate.

    The CA may send you the signed public key certificate by email or require you to download it from a specified web page.
17. Open the saved certificate file in a text editor, and then highlight and copy its entire contents.
18. Select PeopleTools, Security, Security Objects, Digital Certificates.
19. Click Import.

    The Import Certificate page appears.
20. Paste the copied certificate content into the long edit box, and click OK.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker*, "Setting Up Secure Integration Environments," Implementing Node Authentication.

Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker, "Setting Up Secure Integration Environments," Implementing Nonrepudiation

Importing Certificates in Java Key Store

To import certificates in Java Key Store:

1. Open a command prompt.
2. Issue the following command:

```
<PS_HOME>\jre\bin\keytool -import -trustcacerts -alias <alias-name> -file <CA⇒
 Certificate Pem file> -keystore <PS_HOME>\jre\lib\security\cacerts -storepass⇒
 changeit
```

Example:

```
<PS_HOME>\jre\bin\keytool -import -trustcacerts -alias PSFTCA -file ca.pem -⇒
keystore <PS_HOME>\jre\lib\security\cacerts -storepass changeit
```

Note. You will get an error, sslv3 alert certificate unknown, if the certificate is not imported correctly.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker*, "Setting Up Secure Integration Environments," Understanding Client Authentication.

Configuring REN Server

To configure REN server for SSL:

1. Select PeopleTools, REN Server Configuration, REN Server Definition.

   The REN Server Definition page appears.
2. Select the SSL Only check box.
3. Select *RENSERVER* from the Certificate Alias drop-down list.
4. Click *Save*.

See Also

Chapter 5, "Configuring REN Servers," Defining REN Servers, page 66

Configuring REN Clusters

To configure REN clusters:

1. Select PeopleTools, REN Server Configuration, REN Server Cluster.

   The REN Server Cluster page appears.
2. Update REN Server Cluster URL using https and the SSL port.
3. Update REN Server Browser URL using https and the SSL port.
4. Click Save.

 Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Chapter 5, "Configuring REN Servers," Clustering REN Servers and SSL-Enabled REN Servers, page 69

Installing Certificates for Local Node

Apart from CA and REN server certificate, client authentication requires local node certificates, client certificate for the browser, and PSMCAPI certificate.

To install certificates for local node:

1. Select PeopleTools, Security, Security Objects, Digital Certificates.
2. Click +.
3. Select *Local Node* from the Type drop-down list.
4. Enter local node name in Alias. Click Add Root.
5. Select alias of CA certificate from the Issuer Alias lookup button.
6. Click Request.
7. Enter Common Name, Org Unit, Organization, Locality, State/Province, Country, Algorithm, Key Size, Email Address, and Challenge Pswd.

   Note. Common name must be the machine name of REN server's machine, for example PTA112.peoplesoft.com where PTA112 is the machine name and .peoplesoft.com is the domain name

8. Click OK.

   The Certificates Signing Request dialog box appears.

9. Copy and paste the text from the Certificates Signing Request dialog box, and save the text in a file named qelocal.csr in <PS_HOME>\appserv\<domain name>\.

   Note. The file name, qelocal.csr is used as an example only.

10. Click OK.

    The Import link appears.

11. To obtain your local node certificate, submit the qelocal.csr to the CA that issued the selected root certificate.

    The process of obtaining digital certificates varies, depending on the CA. Typically, a CA requires you to paste the content of the PEM-formatted CSR into a form that you submit online. The CA may send you the signed public key certificate by email or require you to download it from a specified web page.

12. Open the saved certificate file in a text editor, and then highlight and copy its entire contents.
13. Select PeopleTools, Security, Security Objects, Digital Certificates.
14. Click the Import link.

    The Import Certificate page appears.

15. Paste the copied certificate content into the long edit box, and click OK.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker*, "Setting Up Secure Integration Environments," Implementing Node Authentication.

Copyright © 1988-2007, Oracle. All rights reserved.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker, "Setting Up Secure Integration Environments," Implementing Nonrepudiation

Generating Client Certificate

Client certificate can be generated by openssl or keytool in P12 format and imported in the browser. Importing the certificates is dependent on the browser.

The following steps are an example to generate a client certificate using openssl. Clients can use keytool or Microsoft CA to generate client certificate.

To generate client certificate using openssl:

1. Generate RSA private key.

   ```
   openssl genrsa -des3 -out <Private key file>
   ```

   Example:

   ```
   openssl genrsa des3 out renclient.key passout pass:pass 1024
   ```

2. Generate Certificate Signing Request file.

   ```
   openssl req -config <ps_home>\src\openssl\openssl- 0.9.7b\apps\
   openssl.cnf -new -M;key <Private key file> -out <CSR file>
   ```

   Example:

   ```
   openssl req config ..\apps\openssl.cnf -new -key renclient.key out
   renclient.csr passout pass:pass
   ```

3. Generate PEM format file.

   ```
   openssl x509 -req -days 365 in <CSR file> -CA <CA PEM File> -CAkey<CA
   Key File> -M;CAcreateserial -out <RenServer PEM file> -outform PEM
   ```

   Note. <CA key file> and <CA PEM file> are Certificate Authority Key file and Certificate Authority in PEM format respectively.

   Example:

   ```
   openssl x509 req days 365 in RENCLIENT.csr -CA ca.pem -CAkey ca.key -⇒
   CAcreateserial -out renclient1.pem -outform PEM -passout pass:pass
   ```

4. Generate .p12 certificate for browser.

   ```
   openssl pkcs12 -export -in <RenServer PEM file>  -out <.p12 file> -inkey
   <Private key file> -name <alias name>
   ```

   Example:

   ```
   openssl pkcs12 -export -in renclient.pem -out renclient.p12 -inkey
   renclient.key -M;name renclient
   ```

Installing PSMCAPI Certificates

To install PSMCAPI certificates:

1. Generate a private key in the keystore using the following command:

 Copyright © 1988-2007, Oracle. All rights reserved.

```
<PS_HOME>\jre\bin\keytool genkey dname CN=Company Name,
OU=Organization Unit, O=M;Organization, L=Locality, S=State/Provenance,
C=Country alias <alias Name>  -M;keyalg RSA validity 365 keystore
<PS_HOME>\jre\lib\cacerts storepass changeit keypass password
```

2. Generate the CSR using the following command:

```
PS_HOME>\jre\bin\keytool certreq alias <alias name> -file <certificate
file name> -keystore <PS_HOME>\jre\lib\security\cacerts storepass
changeit keypass password
```

3. To obtain your certificate, submit the CSR to the CA that issued the selected root certificate.
4. Import the signed certificates into Java keystore using the following command:

```
<PS_HOME>\jre\bin\keytool import alias <alias name> -file <certificate
file .pem> -keystore <PS_HOME>\jre\lib\security\cacerts storepass changeit
keypass password
```

Note. The clients must import the CA certificate in Java Keystore of JRE of PSMCAPI for SSL communication with REN server using KeyTool command.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Integration Broker*, "Setting Up Secure Integration Environments," Implementing Client Authentication.

Copyright © 1988-2007, Oracle. All rights reserved.

Copyright © 1988-2007, Oracle. All rights reserved.

Glossary of PeopleSoft Enterprise Terms

absence entitlement	This element defines rules for granting paid time off for valid absences, such as sick time, vacation, and maternity leave. An absence entitlement element defines the entitlement amount, frequency, and entitlement period.
absence take	This element defines the conditions that must be met before a payee is entitled to take paid time off.
academic career	In PeopleSoft Enterprise Campus Solutions, all course work that a student undertakes at an academic institution and that is grouped in a single student record. For example, a university that has an undergraduate school, a graduate school, and various professional schools might define several academic careers—an undergraduate career, a graduate career, and separate careers for each professional school (law school, medical school, dental school, and so on).
academic institution	In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.
academic organization	In PeopleSoft Enterprise Campus Solutions, an entity that is part of the administrative structure within an academic institution. At the lowest level, an academic organization might be an academic department. At the highest level, an academic organization can represent a division.
academic plan	In PeopleSoft Enterprise Campus Solutions, an area of study—such as a major, minor, or specialization—that exists within an academic program or academic career.
academic program	In PeopleSoft Enterprise Campus Solutions, the entity to which a student applies and is admitted and from which the student graduates.
accounting class	In PeopleSoft Enterprise Performance Management, the accounting class defines how a resource is treated for generally accepted accounting practices. The Inventory class indicates whether a resource becomes part of a balance sheet account, such as inventory or fixed assets, while the Non-inventory class indicates that the resource is treated as an expense of the period during which it occurs.
accounting date	The accounting date indicates when a transaction is recognized, as opposed to the date the transaction actually occurred. The accounting date and transaction date can be the same. The accounting date determines the period in the general ledger to which the transaction is to be posted. You can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date.
accounting split	The accounting split method indicates how expenses are allocated or divided among one or more sets of accounting ChartFields.
accumulator	You use an accumulator to store cumulative values of defined items as they are processed. You can accumulate a single value over time or multiple values over time. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated.
action reason	The reason an employee's job or employment information is updated. The action reason is entered in two parts: a personnel action, such as a promotion, termination, or change from one pay group to another—and a reason for that action. Action reasons are used by PeopleSoft Enterprise Human Resources, PeopleSoft Enterprise Benefits

Copyright © 1988-2007, Oracle. All rights reserved.

Administration, PeopleSoft Enterprise Stock Administration, and the COBRA Administration feature of the Base Benefits business process.

action template In PeopleSoft Enterprise Receivables, outlines a set of escalating actions that the system or user performs based on the period of time that a customer or item has been in an action plan for a specific condition.

activity In PeopleSoft Enterprise Learning Management, an instance of a catalog item (sometimes called a class) that is available for enrollment. The activity defines such things as the costs that are associated with the offering, enrollment limits and deadlines, and waitlisting capacities.

In PeopleSoft Enterprise Performance Management, the work of an organization and the aggregation of actions that are used for activity-based costing.

In PeopleSoft Enterprise Project Costing, the unit of work that provides a further breakdown of projects—usually into specific tasks.

In PeopleSoft Workflow, a specific transaction that you might need to perform in a business process. Because it consists of the steps that are used to perform a transaction, it is also known as a step map.

address usage In PeopleSoft Enterprise Campus Solutions, a grouping of address types defining the order in which the address types are used. For example, you might define an address usage code to process addresses in the following order: billing address, dormitory address, home address, and then work address.

adjustment calendar In PeopleSoft Enterprise Campus Solutions, the adjustment calendar controls how a particular charge is adjusted on a student's account when the student drops classes or withdraws from a term. The charge adjustment is based on how much time has elapsed from a predetermined date, and it is determined as a percentage of the original charge amount.

administrative function In PeopleSoft Enterprise Campus Solutions, a particular functional area that processes checklists, communication, and comments. The administrative function identifies which variable data is added to a person's checklist or communication record when a specific checklist code, communication category, or comment is assigned to the student. This key data enables you to trace that checklist, communication, or comment back to a specific processing event in a functional area.

admit type In PeopleSoft Enterprise Campus Solutions, a designation used to distinguish first-year applications from transfer applications.

agreement In PeopleSoft Enterprise eSettlements, provides a way to group and specify processing options, such as payment terms, pay from a bank, and notifications by a buyer and supplier location combination.

allocation rule In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure from the current node to the root node, checking each node for plans that contain allocation rules.

alternate account A feature in PeopleSoft Enterprise General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.

analysis database In PeopleSoft Enterprise Campus Solutions, database tables that store large amounts of student information that may not appear in standard report formats. The analysis database tables contain keys for all objects in a report that an application program can use to reference other student-record objects that are not contained in the printed report. For instance, the analysis database contains data on courses that are considered

Copyright © 1988-2007, Oracle. All rights reserved.

	for satisfying a requirement but that are rejected. It also contains information on courses captured by global limits. An analysis database is used in PeopleSoft Enterprise Academic Advisement.
Application Messaging	PeopleSoft Application Messaging enables applications within the PeopleSoft Enterprise product family to communicate synchronously or asynchronously with other PeopleSoft Enterprise and third-party applications. An application message defines the records and fields to be published or subscribed to.
AR specialist	Abbreviation for *receivables specialist.* In PeopleSoft Enterprise Receivables, an individual in who tracks and resolves deductions and disputed items.
arbitration plan	The arbiter when multiple price rules match the transaction. This plan determines the order in which the price rules are applied to the transaction base price.
assessment rule	In PeopleSoft Enterprise Receivables, a user-defined rule that the system uses to evaluate the condition of a customer's account or of individual items to determine whether to generate a follow-up action.
asset class	An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.
attribute/value pair	In PeopleSoft Enterprise Directory Interface, relates the data that makes up an entry in the directory information tree.
auction event	In PeopleSoft Strategic Sourcing, a sourcing event where bidders actively compete against one another to achieve the best price or score.
audience	In PeopleSoft Enterprise Campus Solutions, a segment of the database that relates to an initiative, or a membership organization that is based on constituent attributes rather than a dues-paying structure. Examples of audiences include the Class of '65 and Undergraduate Arts & Sciences.
authentication server	A server that is set up to verify users of the system.
base time period	In PeopleSoft Enterprise Business Planning, the lowest level time period in a calendar.
benchmark job	In PeopleSoft Enterprise Workforce Analytics Solution, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.
bid response	In PeopleSoft Strategic Sourcing, the response by a bidder to an event.
billing career	In PeopleSoft Enterprise Campus Solutions, the one career under which other careers are grouped for billing purposes if a student is active simultaneously in multiple careers.
bio bit or bio brief	In PeopleSoft Enterprise Campus Solutions, a report that summarizes information stored in the system about a particular constituent. You can generate standard or specialized reports.
book	In PeopleSoft Enterprise Asset Management, used for storing financial and tax information, such as costs, depreciation attributes, and retirement information on assets.
branch	A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.
budgetary account only	An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called "system-maintained account."

Copyright © 1988-2007, Oracle. All rights reserved.

budget check In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.

budget control In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.

budget period The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.

business activity The name of a subset of a detailed business process. This might be a specific transaction, task, or action that you perform in a business process.

business event In PeopleSoft Enterprise Receivables, defines the processing characteristics for the Receivable Update process for a draft activity.

In PeopleSoft Enterprise Sales Incentive Management, an original business transaction or activity that may justify the creation of a PeopleSoft Enterprise Incentive Management event (a sale, for example).

business process A standard set of 17 business processes are defined and maintained by the PeopleSoft Enterprise product families and are supported by the Business Process Engineering group. An example of a business process is Order Fulfillment, which is a business process that manages sales orders and contracts, inventory, billing, and so forth.

See also *detailed business process.*

business unit constraints In PeopleSoft Strategic Sourcing, these constraints apply to a selected Strategic Sourcing business unit. Spend is tracked across all of the events within the selected Strategic Sourcing business unit.

business task The name of the specific function depicted in one of the business processes.

business unit A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions.

buyer In PeopleSoft Enterprise eSettlements, an organization (or business unit, as opposed to an individual) that transacts with suppliers (vendors) within the system. A buyer creates payments for purchases that are made in the system.

buy event In PeopleSoft Strategic Sourcing, for event creators, the purchase of goods or services, most typically associated with a request for quote, proposal, or reverse auction.For bidders, the sale of goods or services.

campus In PeopleSoft Enterprise Campus Solutions, an entity that is usually associated with a distinct physical administrative unit, that belongs to a single academic institution, that uses a unique course catalog, and that produces a common transcript for students within the same academic career.

cash drawer A repository for monies and payments taken locally.

catalog item In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, "Introduction to Microsoft Word." A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods. A catalog item can have one or more learning activities.

catalog map In PeopleSoft Enterprise Catalog Management, translates values from the catalog source data to the format of the company's catalog.

Copyright © 1988-2007, Oracle. All rights reserved.

catalog partner In PeopleSoft Enterprise Catalog Management, shares responsibility with the enterprise catalog manager for maintaining catalog content.

categorization Associates partner offerings with catalog offerings and groups them into enterprise catalog categories.

category In PeopleSoft Enterprise Campus Solutions, a broad grouping to which specific comments or communications (contexts) are assigned. Category codes are also linked to 3C access groups so that you can assign data-entry or view-only privileges across functions.

channel In PeopleSoft MultiChannel Framework, email, chat, voice (computer telephone integration [CTI]), or a generic event.

ChartField A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft Enterprise application. ChartField values represent individual account numbers, department codes, and so forth.

ChartField balancing You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction.

ChartField combination edit The process of editing journal lines for valid ChartField combinations based on user-defined rules.

ChartKey One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.

checkbook In PeopleSoft Enterprise Promotions Management, enables you to view financial data (such as planned, incurred, and actual amounts) that is related to funds and trade promotions.

checklist code In PeopleSoft Enterprise Campus Solutions, a code that represents a list of planned or completed action items that can be assigned to a staff member, volunteer, or unit. Checklists enable you to view all action assignments on one page.

claimback In the wholesale distribution industry, a contract between supplier and distributor, in which monies are paid to the distributor on the sale of specified products or product groups to targeted customers or customer groups.

class In PeopleSoft Enterprise Campus Solutions, a specific offering of a course component within an academic term.

See also *course.*

Class ChartField A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called *sub-classification.*

clearance In PeopleSoft Enterprise Campus Solutions, the period of time during which a constituent in PeopleSoft Enterprise Contributor Relations is approved for involvement in an initiative or an action. Clearances are used to prevent development officers from making multiple requests to a constituent during the same time period.

clone In PeopleCode, to make a unique copy. In contrast, to *copy* may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.

cohort In PeopleSoft Enterprise Campus Solutions, the highest level of the three-level classification structure that you define for enrollment management. You can define a cohort level, link it to other levels, and set enrollment target numbers for it.

See also *population* and *division*.

Copyright © 1988-2007, Oracle. All rights reserved.

collection To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleTools maintains a set of collections (one per language code) for each search index object.

collection rule In PeopleSoft Enterprise Receivables, a user-defined rule that defines actions to take for a customer based on both the amount and the number of days past due for outstanding balances.

comm key See *communication key.*

communication key In PeopleSoft Enterprise Campus Solutions, a single code for entering a combination of communication category, communication context, communication method, communication direction, and standard letter code. Communication keys (also called *comm keys* or *speed keys*) can be created for background processes as well as for specific users.

compensation object In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.

compensation structure In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.

component interface A component interface is a set of application programming interfaces (APIs) that you can use to access and modify PeopleSoft Enterprise database information using a program instead of the PeopleSoft client.

condition In PeopleSoft Enterprise Receivables, occurs when there is a change of status for a customer's account, such as reaching a credit limit or exceeding a user-defined balance due.

configuration parameter catalog Used to configure an external system with PeopleSoft Enterprise. For example, a configuration parameter catalog might set up configuration and communication parameters for an external server.

configuration plan In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.

constituents In PeopleSoft Enterprise Campus Solutions, friends, alumni, organizations, foundations, or other entities affiliated with the institution, and about which the institution maintains information. The constituent types delivered with PeopleSoft Enterprise Contributor Relations Solutions are based on those defined by the Council for the Advancement and Support of Education (CASE).

constraint A business policy or rule that affects how a sourcing event is awarded. There are three types of constraints: business, global, and event.

content reference Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.

context In PeopleCode, determines which buffer fields can be contextually referenced and which is the current row of data on each scroll level when a PeopleCode program is running.

In PeopleSoft Enterprise Campus Solutions, a specific instance of a comment or communication. One or more contexts are assigned to a category, which you link to

Copyright © 1988-2007, Oracle. All rights reserved.

	3C access groups so that you can assign data-entry or view-only privileges across functions. In PeopleSoft Enterprise Incentive Management, a mechanism that is used to determine the scope of a processing run. PeopleSoft Enterprise Incentive Management uses three types of context: plan, period, and run-level.
control table	Stores information that controls the processing of an application. This type of processing might be consistent throughout an organization, or it might be used only by portions of the organization for more limited sharing of data.
cost plus contract line	A rate-based contract line associated with a fee component of Award, Fixed, Incentive, or Other. Rate-based contract lines associated with a fee type of None are not considered cost-plus contract lines.
cost plus pricing	In PeopleSoft Enterprise Pricer, a pricing method that begins with cost of goods as the basis.
cost profile	A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.
cost row	A cost transaction and amount for a set of ChartFields.
counter sale	A face-to-face customer transaction where the customer typically selects items from the storefront or picks up products that they ordered ahead of time. Customers pay for the goods at the counter and take the goods with them instead of having the goods shipped from a warehouse.
course	In PeopleSoft Enterprise Campus Solutions, a course that is offered by a school and that is typically described in a course catalog. A course has a standard syllabus and credit level; however, these may be modified at the class level. Courses can contain multiple components such as lecture, discussion, and lab. See also *class.*
course share set	In PeopleSoft Enterprise Campus Solutions, a tag that defines a set of requirement groups that can share courses. Course share sets are used in PeopleSoft Enterprise Academic Advisement.
current learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's in-progress learning activities and programs.
data acquisition	In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).
data cube	In PeopleSoft Analytic Calculation Engine, a data cube is a container for one kind of data (such as Sales data) and works with in tandem with one or more dimensions. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and online analytical processing (OLAP) cubes in PeopleSoft Cube Manager.
data elements	Data elements, at their simplest level, define a subset of data and the rules by which to group them. For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.
dataset	A data grouping that enables role-based filtering and distribution of data. You can limit the range and quantity of data that is displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data that is appropriate for the user's roles.

Copyright © 1988-2007, Oracle. All rights reserved.

delivery method In PeopleSoft Enterprise Learning Management, identifies the primary type of delivery method in which a particular learning activity is offered. Also provides default values for the learning activity, such as cost and language. This is primarily used to help learners search the catalog for the type of delivery from which they learn best. Because PeopleSoft Enterprise Learning Management is a blended learning system, it does not enforce the delivery method.

In PeopleSoft Enterprise Supply Chain Management, identifies the method by which goods are shipped to their destinations (such as truck, air, and rail). The delivery method is specified when creating shipment schedules.

delivery method type In PeopleSoft Enterprise Learning Management, identifies how learning activities can be delivered—for example, through online learning, classroom instruction, seminars, books, and so forth—in an organization. The type determines whether the delivery method includes scheduled components.

detailed business process A subset of the business process. For example, the detailed business process named Determine Cash Position is a subset of the business process called Cash Management.

dimension In PeopleSoft Analytic Calculation Engine, a dimension contains a list of one kind of data that can span various contexts, and it is a basic component of an analytic model. Within the analytic model, a dimension is attached to one or more data cubes. In PeopleSoft Cube Manager, a dimension is the most basic component of an OLAP cube and specifies the PeopleSoft metadata to be used to create the dimension's rollup structure. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and OLAP cubes in PeopleSoft Cube Manager.

direct receipt Items shipped from a warehouse or vendor to another warehouse.

direct ship Items shipped from the vendor or warehouse directly to the customer (formerly referred to as *drop ship*).

directory information tree In PeopleSoft Enterprise Directory Interface, the representation of a directory's hierarchical structure.

division In PeopleSoft Enterprise Campus Solutions, the lowest level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a division level, link it to other levels, and set enrollment target numbers for it.

See also *population* and *cohort*.

document sequencing A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in the system for statutory reporting and for tracking commercial transaction activity.

dynamic detail tree A tree that takes its detail values—dynamic details—directly from a table in the database, rather than from a range of values that are entered by the user.

edit table A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft Enterprise application, they can be validated against an edit table to ensure data integrity throughout the system.

effective date A method of dating information in PeopleSoft Enterprise applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.

EIM ledger Abbreviation for *Enterprise Incentive Management ledger.* In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.

Copyright © 1988-2007, Oracle. All rights reserved.

elimination set In PeopleSoft Enterprise General Ledger, a related group of intercompany accounts that is processed during consolidations.

entry event In PeopleSoft Enterprise General Ledger, Receivables, Payables, Purchasing, and Billing, a business process that generates multiple debits and credits resulting from single transactions to produce standard, supplemental accounting entries.

equitization In PeopleSoft Enterprise General Ledger, a business process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations.

equity item limit In PeopleSoft Enterprise Campus Solutions, the amounts of funds set by the institution to be awarded with discretionary or gift funds. The limit could be reduced by amounts equal to such things as expected family contribution (EFC) or parent contribution. Students are packaged by Equity Item Type Groups and Related Equity Item Types. This limit can be used to assure that similar student populations are packaged equally.

event A predefined point either in the Component Processor flow or in the program flow. As each point is encountered, the event activates each component, triggering any PeopleCode program that is associated with that component and that event. Examples of events are FieldChange, SavePreChange, and RowDelete.

In PeopleSoft Enterprise Human Resources, also refers to an incident that affects benefits eligibility.

event constraints In PeopleSoft Strategic Sourcing, these constraints are associated with a specific sourcing event. Spend is tracked within the selected event.

event propagation process In PeopleSoft Enterprise Sales Incentive Management, a process that determines, through logic, the propagation of an original PeopleSoft Enterprise Incentive Management event and creates a derivative (duplicate) of the original event to be processed by other objects. PeopleSoft Enterprise Enterprise Sales Incentive Management uses this mechanism to implement splits, roll-ups, and so on. Event propagation determines who receives the credit.

exception In PeopleSoft Enterprise Receivables, an item that either is a deduction or is in dispute.

exclusive pricing In PeopleSoft Enterprise Order Management, a type of arbitration plan that is associated with a price rule. Exclusive pricing is used to price sales order transactions.

fact In PeopleSoft Enterprise applications, facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table.

financial aid term In PeopleSoft Enterprise Campus Solutions, a combination of a period of time that the school determines as an instructional accounting period and an academic career. It is created and defined during the setup process. Only terms eligible for financial aid are set up for each financial aid career.

financial sanctions For U.S. based companies and their foreign subsidiaries, a federal regulation from the Office of Foreign Assets Control (OFAC) requires that vendors be validated against a Specially Designated Nationals (SDN) list prior to payment.

For PeopleSoft Payables, eSettlements, Cash Management, and Order to Cash, you can validate your vendors against any financial sanctions list (for example, the SDN list, a European Union list, and so on).

forecast item A logical entity with a unique set of descriptive demand and forecast data that is used as the basis to forecast demand. You create forecast items for a wide range of uses, but they ultimately represent things that you buy, sell, or use in your organization and for which you require a predictable usage.

Copyright © 1988-2007, Oracle. All rights reserved.

fund In PeopleSoft Enterprise Promotions Management, a budget that can be used to fund promotional activity. There are four funding methods: top down, fixed accrual, rolling accrual, and zero-based accrual.

gap In PeopleSoft Enterprise Campus Solutions, an artificial figure that sets aside an amount of unmet financial aid need that is not funded with Title IV funds. A gap can be used to prevent fully funding any student to conserve funds, or it can be used to preserve unmet financial aid need so that institutional funds can be awarded.

generic process type In PeopleSoft Process Scheduler, process types are identified by a generic process type. For example, the generic process type SQR includes all SQR process types, such as SQR process and SQR report.

gift table In PeopleSoft Enterprise Campus Solutions, a table or so-called *donor pyramid* describing the number and size of gifts that you expect will be needed to successfully complete the campaign in PeopleSoft Enterprise Contributor Relations. The gift table enables you to estimate the number of donors and prospects that you need at each gift level to reach the campaign goal.

GDS Abbreviation for *Global Distribution System.* Broad-based term to describe all computer reservation systems for making travel plans.

GL business unit Abbreviation for *general ledger business unit.* A unit in an organization that is an independent entity for accounting purposes. It maintains its own set of accounting books.

See also *business unit.*

GL entry template Abbreviation for *general ledger entry template.* In PeopleSoft Enterprise Campus Solutions, a template that defines how a particular item is sent to the general ledger. An item-type maps to the general ledger, and the GL entry template can involve multiple general ledger accounts. The entry to the general ledger is further controlled by high-level flags that control the summarization and the type of accounting—that is, accrual or cash.

GL Interface process Abbreviation for *General Ledger Interface process.* In PeopleSoft Enterprise Campus Solutions, a process that is used to send transactions from PeopleSoft Enterprise Student Financials to the general ledger. Item types are mapped to specific general ledger accounts, enabling transactions to move to the general ledger when the GL Interface process is run.

global constraints In PeopleSoft Strategic Sourcing, these constraints apply across multiple Strategic Sourcing business units. Spend is tracked across all of the events from the multiple Strategic Sourcing business units.

group In PeopleSoft Enterprise Billing and Receivables, a posting entity that comprises one or more transactions (items, deposits, payments, transfers, matches, or write-offs).

In PeopleSoft Enterprise Human Resources Management and Supply Chain Management, any set of records that are associated under a single name or variable to run calculations in PeopleSoft business processes. In PeopleSoft Enterprise Time and Labor, for example, employees are placed in groups for time reporting purposes.

ideal response In PeopleSoft Strategic Sourcing, a question that requires the response to match the ideal value for the bid to be considered eligible for award. If the response does not match the ideal value, you can still submit the bid, but it will be disqualified and ineligible for award.

incentive object In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, and user interaction objects.

Copyright © 1988-2007, Oracle. All rights reserved.

incentive rule In PeopleSoft Enterprise Sales Incentive Management, the commands that act on transactions and turn them into compensation. A rule is one part in the process of turning a transaction into compensation.

incur In PeopleSoft Enterprise Promotions Management, to become liable for a promotional payment. In other words, you owe that amount to a customer for promotional activities.

initiative In PeopleSoft Enterprise Campus Solutions, the basis from which all advancement plans are executed. It is an organized effort targeting a specific constituency, and it can occur over a specified period of time with specific purposes and goals. An initiative can be a campaign, an event, an organized volunteer effort, a membership drive, or any other type of effort defined by the institution. Initiatives can be multipart, and they can be related to other initiatives. This enables you to track individual parts of an initiative, as well as entire initiatives.

inquiry access In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user only to view data.

See also *update access.*

institution In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.

integration A relationship between two compatible integration points that enables communication to take place between systems. Integrations enable PeopleSoft Enterprise applications to work seamlessly with other PeopleSoft Enterprise applications or with third-party systems or software.

integration point An interface that a system uses to communicate with another PeopleSoft Enterprise application or an external application.

integration set A logical grouping of integrations that applications use for the same business purpose. For example, the integration set ADVANCED_SHIPPING_ORDER contains all of the integrations that notify a customer that an order has shipped.

item In PeopleSoft Enterprise Inventory, a tangible commodity that is stored in a business unit (shipped from a warehouse).

In PeopleSoft Enterprise Demand Planning, Inventory Policy Planning, and Supply Planning, a noninventory item that is designated as being used for planning purposes only. It can represent a family or group of inventory items. It can have a planning bill of material (BOM) or planning routing, and it can exist as a component on a planning BOM. A planning item cannot be specified on a production or engineering BOM or routing, and it cannot be used as a component in a production. The quantity on hand will never be maintained.

In PeopleSoft Enterprise Receivables, an individual receivable. An item can be an invoice, a credit memo, a debit memo, a write-off, or an adjustment.

item shuffle In PeopleSoft Enterprise Campus Solutions, a process that enables you to change a payment allocation without having to reverse the payment.

itinerary In PeopleSoft Expenses, a collection of travel reservations. Itineraries can have reservations that are selected and reserved with the travel vendor. These itineraries are not yet paid for and can be referred to as *pending reservations.* Reservations that have been paid for are referred to as *confirmed reservations.*

joint communication In PeopleSoft Enterprise Campus Solutions, one letter that is addressed jointly to two people. For example, a letter might be addressed to both Mr. Sudhir Awat and Ms. Samantha Mortelli. A relationship must be established between the two individuals in the database, and at least one of the individuals must have an ID in the database.

Copyright © 1988-2007, Oracle. All rights reserved.

keyword In PeopleSoft Enterprise Campus Solutions, a term that you link to particular elements within PeopleSoft Enterprise Student Financials, Financial Aid, and Contributor Relations. You can use keywords as search criteria that enable you to locate specific records in a search dialog box.

KPI An abbreviation for *key performance indicator.* A high-level measurement of how well an organization is doing in achieving critical success factors. This defines the data value or calculation upon which an assessment is determined.

KVI Abbreviation for *Known Value Item.* Term used for products or groups of products where the selling price cannot be reduced or increased.

landlord In PeopleSoft Real Estate Management, an entity that owns real estate and leases the real estate to tenants.

LDIF file Abbreviation for *Lightweight Directory Access Protocol (LDAP) Data Interchange Format file.* Contains discrepancies between PeopleSoft Enterprise data and directory data.

learner group In PeopleSoft Enterprise Learning Management, a group of learners who are linked to the same learning environment. Members of the learner group can share the same attributes, such as the same department or job code. Learner groups are used to control access to and enrollment in learning activities and programs. They are also used to perform group enrollments and mass enrollments in the back office.

learning components In PeopleSoft Enterprise Learning Management, the foundational building blocks of learning activities. PeopleSoft Enterprise Learning Management supports six basic types of learning components: web-based, session, webcast, test, survey, and assignment. One or more of these learning component types compose a single learning activity.

learning environment In PeopleSoft Enterprise Learning Management, identifies a set of categories and catalog items that can be made available to learner groups. Also defines the default values that are assigned to the learning activities and programs that are created within a particular learning environment. Learning environments provide a way to partition the catalog so that learners see only those items that are relevant to them.

learning history In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities and programs.

lease In PeopleSoft Real Estate Management, a legally binding agreement between a landlord and a tenant, where the tenant rents all or part of a physical property from the landlord.

lease abstract In PeopleSoft Real Estate Management, a summarized version of the complete lease contract with only the important terms. The lease abstract usually fits on one page and does not include legal terminology.

ledger mapping You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as *rates*) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.

library section In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.

line In PeopleSoft Strategic Sourcing, an individual item or service upon which there can be a bid.

Copyright © 1988-2007, Oracle. All rights reserved.

linked section In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.

linked variable In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.

LMS Abbreviation for *learning management system.* In PeopleSoft Enterprise Campus Solutions, LMS is a PeopleSoft Enterprise Student Records feature that provides a common set of interoperability standards that enable the sharing of instructional content and data between learning and administrative environments.

load In PeopleSoft Enterprise Inventory, identifies a group of goods that are shipped together. Load management is a feature of PeopleSoft Enterprise Inventory that is used to track the weight, the volume, and the destination of a shipment.

local functionality In PeopleSoft Enterprise HRMS, the set of information that is available for a specific country. You can access this information when you click the appropriate country flag in the global window, or when you access it by a local country menu.

location Locations enable you to indicate the different types of addresses—for a company, for example, one address to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each address has a different location number. The primary location—indicated by a *1*—is the address you use most often and may be different from the main address.

logistical task In PeopleSoft Enterprise Services Procurement, an administrative task that is related to hiring a service provider. Logistical tasks are linked to the service type on the work order so that different types of services can have different logistical tasks. Logistical tasks include both preapproval tasks (such as assigning a new badge or ordering a new laptop) and postapproval tasks (such as scheduling orientation or setting up the service provider email). The logistical tasks can be mandatory or optional. Mandatory preapproval tasks must be completed before the work order is approved. Mandatory postapproval tasks, on the other hand, must be completed before a work order is released to a service provider.

market template In PeopleSoft Enterprise Incentive Management, additional functionality that is specific to a given market or industry and is built on top of a product category.

mass change In PeopleSoft Enterprise Campus Solutions, mass change is a SQL generator that can be used to create specialized functionality. Using mass change, you can set up a series of Insert, Update, or Delete SQL statements to perform business functions that are specific to the institution.

See also *3C engine.*

match group In PeopleSoft Enterprise Receivables, a group of receivables items and matching offset items. The system creates match groups by using user-defined matching criteria for selected field values.

MCF server Abbreviation for *PeopleSoft MultiChannel Framework server.* Comprises the universal queue server and the MCF log server. Both processes are started when *MCF Servers* is selected in an application server domain configuration.

merchandising activity In PeopleSoft Enterprise Promotions Management, a specific discount type that is associated with a trade promotion (such as off-invoice, billback or rebate, or lump-sum payment) that defines the performance that is required to receive the discount. In the industry, you may know this as an offer, a discount, a merchandising event, an event, or a tactic.

Copyright © 1988-2007, Oracle. All rights reserved.

meta-SQL Meta-SQL constructs expand into platform-specific SQL substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, and PeopleSoft Application Engine programs.

metastring Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform.

multibook In PeopleSoft Enterprise General Ledger, multiple ledgers having multiple-base currencies that are defined for a business unit, with the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers).

multicurrency The ability to process transactions in a currency other than the business unit's base currency.

national allowance In PeopleSoft Enterprise Promotions Management, a promotion at the corporate level that is funded by nondiscretionary dollars. In the industry, you may know this as a national promotion, a corporate promotion, or a corporate discount.

NDP Abbreviation for *Non-Discountable Products.* Term used for products or groups of products where the selling price cannot be decreased.

need In PeopleSoft Enterprise Campus Solutions, the difference between the cost of attendance (COA) and the expected family contribution (EFC). It is the gap between the cost of attending the school and the student's resources. The financial aid package is based on the amount of financial need. The process of determining a student's need is called *need analysis.*

node-oriented tree A tree that is based on a detail structure, but the detail values are not used.

Optimization Engine A PeopleTools component that Strategic Sourcing leverages to evaluate bids and determine an ideal award allocation. The award recommendation is based on maximizing the value while adhering to purchasing and company objectives and constraints.

pagelet Each block of content on the home page is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provide users with a snapshot of their most relevant PeopleSoft Enterprise and non-PeopleSoft Enterprise content.

participant In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process.

participant object Each participant object may be related to one or more compensation objects.

See also *compensation object.*

partner A company that supplies products or services that are resold or purchased by the enterprise.

pay cycle In PeopleSoft Enterprise Payables, a set of rules that define the criteria by which it should select scheduled payments for payment creation.

payment shuffle In PeopleSoft Enterprise Campus Solutions, a process allowing payments that have been previously posted to a student's account to be automatically reapplied when a higher priority payment is posted or the payment allocation definition is changed.

pending item In PeopleSoft Enterprise Receivables, an individual receivable (such as an invoice, a credit memo, or a write-off) that has been entered in or created by the system, but hasn't been posted.

Copyright © 1988-2007, Oracle. All rights reserved.

PeopleCode PeopleCode is a proprietary language, executed by the PeopleSoft Enterprise component processor. PeopleCode generates results based on existing data or user actions. By using various tools provided with PeopleTools, external services are available to all PeopleSoft Enterprise applications wherever PeopleCode can be executed.

PeopleCode event See *event.*

PeopleSoft Pure Internet Architecture The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of a relational database management system (RDBMS), an application server, a web server, and a browser.

performance measurement In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.

period context In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.

person of interest A person about whom the organization maintains information but who is not part of the workforce.

personal portfolio In PeopleSoft Enterprise Campus Solutions, the user-accessible menu item that contains an individual's name, address, telephone number, and other personal information.

phase A level 1 task, meaning that if a task had subtasks, the level 1 task would be considered the phase.

pickup quantity The product quantity that the customer is taking with them from the counter sales environment.

plan In PeopleSoft Enterprise Sales Incentive Management, a collection of allocation rules, variables, steps, sections, and incentive rules that instruct the PeopleSoft Enterprise Incentive Management engine in how to process transactions.

plan context In PeopleSoft Enterprise Incentive Management, correlates a participant with the compensation plan and node to which the participant is assigned, enabling the PeopleSoft Enterprise Incentive Management system to find anything that is associated with the node and that is required to perform compensation processing. Each participant, node, and plan combination represents a unique plan context—if three participants are on a compensation structure, each has a different plan context. Configuration plans are identified by plan contexts and are associated with the participants that refer to them.

plan template In PeopleSoft Enterprise Incentive Management, the base from which a plan is created. A plan template contains common sections and variables that are inherited by all plans that are created from the template. A template may contain steps and sections that are not visible in the plan definition.

planned learning In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's planned learning activities and programs.

planning instance In PeopleSoft Enterprise Supply Planning, a set of data (business units, items, supplies, and demands) constituting the inputs and outputs of a supply plan.

population In PeopleSoft Enterprise Campus Solutions, the middle level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and

Copyright © 1988-2007, Oracle. All rights reserved.

	Admissions for enrollment management. You can define a population level, link it to other levels, and set enrollment target numbers for it. See also *division* and *cohort.*
portal registry	In PeopleSoft Enterprise applications, the portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that defines both the structure and content of a portal through a hierarchical, tree-like structure of folders useful for organizing and securing content references.
predecessor task	A task that you must complete before you start another task.
price breaks	In PeopleSoft Strategic Sourcing, a price discount or surcharge that a bidder may apply based on the quantity awarded.
price components	In PeopleSoft Strategic Sourcing, the various components, such as material costs, labor costs, shipping costs, and so on that make up the overall bid price.
price list	Enables you to select products and conditions for which the price list applies to a transaction. During a transaction, the system either determines the product price based on the predefined search hierarchy for the transaction or uses the product's lowest price on any associated, active price lists. This price is used as the basis for any further discounts and surcharges.
price rule	The conditions that must be met for adjustments to be applied to the base price. Multiple rules can apply when conditions of each rule are met.
price rule conditions	Conditions that select the price-by fields, the values for the price-by fields, and the operator that determines how the price-by fields relate to the transaction.
price rule key	The fields that are available to define price rule conditions (which are used to match a transaction) on the price rule.
primacy number	In PeopleSoft Enterprise Campus Solutions, a number that the system uses to prioritize financial aid applications when students are enrolled in multiple academic careers and academic programs at the same time. The Consolidate Academic Statistics process uses the primacy number indicated for both the career and program at the institutional level to determine a student's primary career and program. The system also uses the number to determine the primary student attribute value that is used when you extract data to report on cohorts. The lowest number takes precedence.
primary name type	In PeopleSoft Enterprise Campus Solutions, the name type that is used to link the name stored at the highest level within the system to the lower-level set of names that an individual provides.
process category	In PeopleSoft Process Scheduler, processes that are grouped for server load balancing and prioritization.
process group	In PeopleSoft Enterprise Financials, a group of application processes (performed in a defined order) that users can initiate in real time, directly from a transaction entry page.
process definition	Process definitions define each run request.
process instance	A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.
process job	You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.

Copyright © 1988-2007, Oracle. All rights reserved.

process request A single run request, such as a Structured Query Report (SQR), a COBOL or Application Engine program, or a Crystal report that you run through PeopleSoft Process Scheduler.

process run control A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.

product A PeopleSoft Enterprise or third-party product. PeopleSoft organizes its software products into product families and product lines. Interactive Services Repository contains information about every release of every product that PeopleSoft sells, as well as products from certified third-party companies. These products appear with the product name and release number.

product adds The pricing functionality where buying product A gets product B for free or at a price (formerly referred to as *giveaways*).

product bidding In PeopleSoft Strategic Sourcing, the placing of a bid on behalf of the bidder, up or down to the bidder's specified amount, so that the bidder can be the leading bidder.

product category In PeopleSoft Enterprise Incentive Management, indicates an application in the PeopleSoft Enterprise Incentive Management suite of products. Each transaction in the PeopleSoft Enterprise Incentive Management system is associated with a product category.

product family A group of products that are related by common functionality. The family names that can be searched using Interactive Service Repository are Oracle's PeopleSoft Enterprise, PeopleSoft EnterpriseOne, PeopleSoft World, and third-party, certified partners.

product line The name of a PeopleSoft Enterprise product line or the company name of a third-party certified partner. Integration Services Repository enables you to search for integration points by product line.

programs In PeopleSoft Enterprise Learning Management, a high-level grouping that guides the learner along a specific learning path through sections of catalog items. PeopleSoft Enterprise Learning Systems provides two types of programs—curricula and certifications.

progress log In PeopleSoft Enterprise Services Procurement, tracks deliverable-based projects. This is similar to the time sheet in function and process. The service provider contact uses the progress log to record and submit progress on deliverables. The progress can be logged by the activity that is performed, by the percentage of work that is completed, or by the completion of milestone activities that are defined for the project.

project transaction In PeopleSoft Enterprise Project Costing, an individual transaction line that represents a cost, time, budget, or other transaction row.

promotion In PeopleSoft Enterprise Promotions Management, a trade promotion, which is typically funded from trade dollars and used by consumer products manufacturers to increase sales volume.

prospects In PeopleSoft Enterprise Campus Solutions, students who are interested in applying to the institution.

In PeopleSoft Enterprise Contributor Relations, individuals and organizations that are most likely to make substantial financial commitments or other types of commitments to the institution.

proxy bidding In PeopleSoft Strategic Sourcing, the placing of a bid on behalf of the bidder, up or down to the bidder's specified amount, so that the bidder can be the leading bidder.

Copyright © 1988-2007, Oracle. All rights reserved.

publishing In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.

rating components In PeopleSoft Enterprise Campus Solutions, variables used with the Equation Editor to retrieve specified populations.

record group A set of logically and functionally related control tables and views. Record groups help enable TableSet sharing, which eliminates redundant data entry. Record groups ensure that TableSet sharing is applied consistently across all related tables and views.

record input VAT flag Abbreviation for *record input value-added tax flag.* Within PeopleSoft Enterprise Purchasing, Payables, and General Ledger, this flag indicates that you are recording input VAT on the transaction. This flag, in conjunction with the record output VAT flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. For all cases within Purchasing and Payables where VAT information is tracked on a transaction, this flag is set to Yes. This flag is not used in PeopleSoft Enterprise Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in PeopleSoft Enterprise Expenses, where it is assumed that you are always recording only input VAT.

record output VAT flag Abbreviation for *record output value-added tax flag.*

See *record input VAT flag.*

recname The name of a record that is used to determine the associated field to match a value or set of values.

recognition In PeopleSoft Enterprise Campus Solutions, the recognition type indicates whether the PeopleSoft Enterprise Contributor Relations donor is the primary donor of a commitment or shares the credit for a donation. Primary donors receive hard credit that must total 100 percent. Donors that share the credit are given soft credit. Institutions can also define other share recognition-type values such as memo credit or vehicle credit.

reference data In PeopleSoft Enterprise Sales Incentive Management, system objects that represent the sales organization, such as territories, participants, products, customers, and channels.

reference object In PeopleSoft Enterprise Incentive Management, this dimension-type object further defines the business. Reference objects can have their own hierarchy (for example, product tree, customer tree, industry tree, and geography tree).

reference transaction In commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition.

regional sourcing In PeopleSoft Enterprise Purchasing, provides the infrastructure to maintain, display, and select an appropriate vendor and vendor pricing structure that is based on a regional sourcing model where the multiple ship to locations are grouped. Sourcing may occur at a level higher than the ship to location.

relationship object In PeopleSoft Enterprise Incentive Management, these objects further define a compensation structure to resolve transactions by establishing associations between compensation objects and business objects.

remote data source data Data that is extracted from a separate database and migrated into the local database.

Copyright © 1988-2007, Oracle. All rights reserved.

REN server Abbreviation for *real-time event notification server* in PeopleSoft MultiChannel Framework.

requester In PeopleSoft Enterprise eSettlements, an individual who requests goods or services and whose ID appears on the various procurement pages that reference purchase orders.

reservations In PeopleSoft Expenses, travel reservations that have been placed with the travel vendor.

reversal indicator In PeopleSoft Enterprise Campus Solutions, an indicator that denotes when a particular payment has been reversed, usually because of insufficient funds.

RFI event In PeopleSoft Strategic Sourcing, a request for information.

RFx event In PeopleSoft Strategic Sourcing, a request for proposal or request for a quote event when bidders submit their overall best bids and during which bidders do not actively compete against one another.

role Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.

role user A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs.

roll up In a tree, to roll up is to total sums based on the information hierarchy.

run control A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data.

run control ID A unique ID to associate each user with his or her own run control table entries.

run-level context In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.

saved bid In PeopleSoft Strategic Sourcing, a bid that has been created but not submitted. Only submitted bids are eligible for award.

score In PeopleSoft Strategic Sourcing, the numerical sum of answers (percentages) to bid factors on an event. Scores appear only to bidders on auction events.

SCP SCBM XML message Abbreviation for *Supply Chain Planning Supply Chain Business Modeler Extensible Markup Language message.* Supply Chain Business Modeler uses XML as the format for all data that it imports and exports.

search query You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.

search/match In PeopleSoft Enterprise Campus Solutions and PeopleSoft Enterprise Human Resources Management Solutions, a feature that enables you to search for and identify duplicate records in the database.

seasonal address In PeopleSoft Enterprise Campus Solutions, an address that recurs for the same length of time at the same time of year each year until adjusted or deleted.

Copyright © 1988-2007, Oracle. All rights reserved.

section In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.

security event In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.

sell event In PeopleSoft Strategic Sourcing, for event creators, the sale of goods or services most typically associated with forward auctions. For bidders, the purchase of goods or services.

serial genealogy In PeopleSoft Enterprise Manufacturing, the ability to track the composition of a specific, serial-controlled item.

serial in production In PeopleSoft Enterprise Manufacturing, enables the tracing of serial information for manufactured items. This is maintained in the Item Master record.

service impact In PeopleSoft Enterprise Campus Solutions, the resulting action triggered by a service indicator. For example, a service indicator that reflects nonpayment of account balances by a student might result in a service impact that prohibits registration for classes.

service indicator In PeopleSoft Enterprise Campus Solutions, indicates services that may be either withheld or provided to an individual. Negative service indicators indicate holds that prevent the individual from receiving specified services, such as check-cashing privileges or registration for classes. Positive service indicators designate special services that are provided to the individual, such as front-of-line service or special services for disabled students.

session In PeopleSoft Enterprise Campus Solutions, time elements that subdivide a term into multiple time periods during which classes are offered. In PeopleSoft Enterprise Contributor Relations, a session is the means of validating gift, pledge, membership, or adjustment data entry . It controls access to the data entered by a specific user ID. Sessions are balanced, queued, and then posted to the institution's financial system. Sessions must be posted to enter a matching gift or pledge payment, to make an adjustment, or to process giving clubs or acknowledgements.

In PeopleSoft Enterprise Learning Management, a single meeting day of an activity (that is, the period of time between start and finish times within a day). The session stores the specific date, location, meeting time, and instructor. Sessions are used for scheduled training.

session template In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise Learning Management activity—characteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.

setup relationship In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.

share driver expression In PeopleSoft Enterprise Business Planning, a named planning method similar to a driver expression, but which you can set up globally for shared use within a single planning application or to be shared between multiple planning applications through PeopleSoft Enterprise Warehouse.

short-term customer A customer not in the system who is entered during sales order entry using a template.

Copyright © 1988-2007, Oracle. All rights reserved.

single signon With single signon, users can, after being authenticated by a PeopleSoft Enterprise application server, access a second PeopleSoft Enterprise application server without entering a user ID or password.

source key process In PeopleSoft Enterprise Campus Solutions, a process that relates a particular transaction to the source of the charge or financial aid. On selected pages, you can drill down into particular charges.

source transaction In commitment control, any transaction generated in a PeopleSoft Enterprise or third-party application that is integrated with commitment control and which can be checked against commitment control budgets. For example, a pre-encumbrance, encumbrance, expenditure, recognized revenue, or collected revenue transaction.

sourcing objective For constraints, the option to designate whether a business rule is required (mandatory) or is only recommended (target).

speed key See *communication key.*

SpeedChart A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.

SpeedType A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.

staging A method of consolidating selected partner offerings with the offerings from the enterprise's other partners.

standard letter code In PeopleSoft Enterprise Campus Solutions, a standard letter code used to identify each letter template available for use in mail merge functions. Every letter generated in the system must have a standard letter code identification.

statutory account Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft Enterprise, this is equivalent to the Alternate Account (ALTACCT) ChartField.

step In PeopleSoft Enterprise Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.

storage level In PeopleSoft Enterprise Inventory, identifies the level of a material storage location. Material storage locations are made up of a business unit, a storage area, and a storage level. You can set up to four storage levels.

subcustomer qualifier A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.

Summary ChartField You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).

summary ledger An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.

summary time period In PeopleSoft Enterprise Business Planning, any time period (other than a base time period) that is an aggregate of other time periods, including other summary time periods and base time periods, such as quarter and year total.

Copyright © 1988-2007, Oracle. All rights reserved.

summary tree A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the *basis* tree). A summary tree structure specifies the details on which the summary trees are to be built.

syndicate To distribute a production version of the enterprise catalog to partners.

system function In PeopleSoft Enterprise Receivables, an activity that defines how the system generates accounting entries for the general ledger.

system source The system source identifies the source of a transaction row in the database. For example, a transaction that originates in PeopleSoft Enterprise Expenses contains a system source code of BEX (Expenses Batch).

When PeopleSoft Enterprise Project Costing prices the source transaction row for billing, the system creates a new row with a system source code of PRP (Project Costing pricing), which represents the system source of the new row. System source codes can identify sources that are internal or external to the PeopleSoft Enterprise system. For example, processes that import data from Microsoft Project into PeopleSoft Enterprise applications create transaction rows with a source code of MSP (Microsoft Project).

TableSet A means of sharing similar sets of values in control tables, where the actual data values are different but the structure of the tables is the same.

TableSet sharing Shared data that is stored in many tables that are based on the same TableSets. Tables that use TableSet sharing contain the SETID field as an additional key or unique identifier.

target currency The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.

task A deliverable item on the detailed sourcing plan.

tax authority In PeopleSoft Enterprise Campus Solutions, a user-defined element that combines a description and percentage of a tax with an account type, an item type, and a service impact.

template A template is HTML code associated with a web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft Enterprise, you use templates to build a page by combining HTML from a number of sources. For a PeopleSoft Enterprise portal, all templates must be registered in the portal registry, and each content reference must be assigned a template.

tenant In PeopleSoft Real Estate Management, an entity that leases real estate from a landlord.

territory In PeopleSoft Enterprise Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.

third party A company or vendor that has extensive PeopleSoft Enterprise product knowledge and whose products and integrations have been certified and are compatible with PeopleSoft Enterprise applications.

tiered pricing Enables different portions of a schedule to be priced differently from one another.

time span A relative period, such as year-to-date or current period, that various PeopleSoft General Ledger functions and reports can use when a rolling time frame, rather than a specific date, is required.

total cost In PeopleSoft Strategic Sourcing, the estimated dollar cost (sum of real price dollars and potential “soft” or non-price dollars) of a particular award approach.

Copyright © 1988-2007, Oracle. All rights reserved.

travel group In PeopleSoft Expenses, the organization's travel rules and polices that are associated with specific business units, departments, or employees. You must define at least one travel group when setting up the PeopleSoft Expenses travel feature. You must define and associate at least one travel group with a travel vendor.

travel partner In PeopleSoft Expenses, the travel vendor with which the organization has a contractual relationship.

3C engine Abbreviation for *Communications, Checklists, and Comments engine.* In PeopleSoft Enterprise Campus Solutions, the 3C engine enables you to automate business processes that involve additions, deletions, and updates to communications, checklists, and comments. You define events and triggers to engage the engine, which runs the mass change and processes the 3C records (for individuals or organizations) immediately and automatically from within business processes.

3C group Abbreviation for *Communications, Checklists, and Comments group.* In PeopleSoft Enterprise Campus Solutions, a method of assigning or restricting access privileges. A 3C group enables you to group specific communication categories, checklist codes, and comment categories. You can then assign the group inquiry-only access or update access, as appropriate.

trace usage In PeopleSoft Enterprise Manufacturing, enables the control of which components will be traced during the manufacturing process. Serial- and lot-controlled components can be traced. This is maintained in the Item Master record.

transaction allocation In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables.

transaction state In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and "picked up" by a different section for further processing.

Translate table A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own.

tree The graphical hierarchy in PeopleSoft Enterprise systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.

tuition lock In PeopleSoft Enterprise Campus Solutions, a feature in the Tuition Calculation process that enables you to specify a point in a term after which students are charged a minimum (or *locked*) fee amount. Students are charged the locked fee amount even if they later drop classes and take less than the normal load level for that tuition charge.

unclaimed transaction In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator.

universal navigation header Every PeopleSoft Enterprise portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user.

update access In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user to edit and update data.

Copyright © 1988-2007, Oracle. All rights reserved.

	See also *inquiry access*.
user interaction object	In PeopleSoft Enterprise Sales Incentive Management, used to define the reporting components and reports that a participant can access in his or her context. All PeopleSoft Enterprise Sales Incentive Management user interface objects and reports are registered as user interaction objects. User interaction objects can be linked to a compensation structure node through a compensation relationship object (individually or as groups).
variable	In PeopleSoft Enterprise Sales Incentive Management, the intermediate results of calculations. Variables hold the calculation results and are then inputs to other calculations. Variables can be plan variables that persist beyond the run of an engine or local variables that exist only during the processing of a section.
VAT exception	Abbreviation for *value-added tax exception.* A temporary or permanent exemption from paying VAT that is granted to an organization. This terms refers to both VAT exoneration and VAT suspension.
VAT exempt	Abbreviation for *value-added tax exempt.* Describes goods and services that are not subject to VAT. Organizations that supply exempt goods or services are unable to recover the related input VAT. This is also referred to as exempt without recovery.
VAT exoneration	Abbreviation for *value-added tax exoneration.* An organization that has been granted a permanent exemption from paying VAT due to the nature of that organization.
VAT suspension	Abbreviation for *value-added tax suspension.* An organization that has been granted a temporary exemption from paying VAT.
warehouse	A PeopleSoft Enterprise data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions.
weight or weighting	In PeopleSoft Strategic Sourcing, how important the line or question is to the overall event. Weighting is used to score and analyze bids. For RFx and RFI events, weightings may or may not appear to bidders.
work order	In PeopleSoft Enterprise Services Procurement, enables an enterprise to create resource-based and deliverable-based transactions that specify the basic terms and conditions for hiring a specific service provider. When a service provider is hired, the service provider logs time or progress against the work order.
worker	A person who is part of the workforce; an employee or a contingent worker.
workset	A group of people and organizations that are linked together as a set. You can use worksets to simultaneously retrieve the data for a group of people and organizations and work with the information on a single page.
worksheet	A way of presenting data through a PeopleSoft Enterprise Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.
worklist	The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.
XML link	The XML Linking language enables you to insert elements into XML documents to create a links between resources.
XML schema	An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.
XPI	Abbreviation for *eXtended Process Integrator.* PeopleSoft XPI is the integration infrastructure that enables both real-time and batch communication with JD Edwards EnterpriseOne applications.

Copyright © 1988-2007, Oracle. All rights reserved.

yield by operation In PeopleSoft Enterprise Manufacturing, the ability to plan the loss of a manufactured item on an operation-by-operation basis.

zero-rated VAT Abbreviation for *zero-rated value-added tax.* A VAT transaction with a VAT code that has a tax percent of zero. Used to track taxable VAT activity where no actual VAT amount is charged. Organizations that supply zero-rated goods and services can still recover the related input VAT. This is also referred to as exempt with recovery.

Copyright © 1988-2007, Oracle. All rights reserved.

Copyright © 1988-2007, Oracle. All rights reserved.

Index

A

A2AChat class
  constructor 233, 423
  fields 234, 424
  method 236
  methods 424
A2AChat class fields
  address 234
  agentID 234
  agentName 234
  appData 235
  chatType 235
  customerName 235
  id 235
  isConference 236
  jr 236
  question 236
  subject 236
  type 236
  uniqueueId 236
A2AChat class method 236
A2AChat constructor 233, 423
A2AChat object 423
A2AChat.PS_JR 312, 449
A2AChat.TYPE_ANSWER 313, 449
A2AChat.TYPE_CONSULT 313, 449
A2AChatAddress class
  callback event methods 425
  constructor 237, 424
  event method 239
    See Also A2AChatAddress class callback event methods
  fields 237, 424
  method 238
  methods 425
A2AChatAddress class callback event methods
  onChatEnded 239
  onInitiatingChat 239
  onNotify 239
A2AChatAddress class fields
  id 238
  tasks 238
A2AChatAddress constructor 237, 424
A2AChatAddress object 424
abandon
  EmailConnection class methods 280
  GenericConnection class methods 301
acceptTask 228, 422
ACD, *See* Automatic Call Distributor (ACD)
additional documentation xxxviii
addKeyValue
  AppData class 244, 427
  CallData class 249
  CallDataclass 429
  ChatData class 435
  ChatData class methods 272
  EmailData class 441
  EmailData class methods 287
  GenericData class 448
  GenericData class methods 308
  UserData class 396
address
  A2AChat class 234, 424
  Chat class 252, 430
  Email class 273, 437
  GenericTask class 310, 448
_Address callback event methods 421
_Address class
  callback event method 226
  constructor 225
  fields 226
_Address class fields
  caps 226
  id 226
_Address constructor 226, 421
_Address fields 421
_Address object 421
_Address() constructor 421
addresses
  Session class 364, 461
  User class 378, 465
agent
  Email class 438
  EmailAddress class fields 276
  GenericAddress class 445
  GenericAddress class fields 298
agent chat window 131
agent console, *See* MultiChannel Console

Copyright © 1988-2007, Oracle. All rights reserved.

Agent Console page 167
Agent Info page 37
Agent page 103
Agent State Summary component (MCF_AGENTSTATE_CMP) 116
Agent State Summary page 118
agent statistics
  monitoring 95
agent-specific URL 197
agentId
  A2AChat class 424
  Chat class 252, 430
  Email class 273, 437
  GenericTask class 310, 448
  _User class 423
agentID
  A2AChat class 234
  _User class 231
agentName 234, 424
agentPassword 378, 465
agents
  activating/viewing chats 130
  assigning tasks 89
    See Also tasks
  assigning to queues 104
  communicating by chat 6
  communicating via chat 131
  configuring CTI agents 33
  configuring MCF agents 103
  configuring windows 108
  creating 103
  customizing presence 105
  defining 103
  defining agent-specific URLs 110
  defining for third party 191
  defining optional characteristics 106
  defining presence 197
  deleting 104
  deleting queues 105
  displaying number of agents logged into queues 95
  inviting to chats 133
  logging the status 96
  monitoring 170
  moving 114
  moving physical queues 114
  personalizing chat 109
  selecting buddies 130
  selecting state masks 28
  setting maximum timeouts for queue server logout 100
  setting maximum workload 105
  setting presence 130
  setting skill level 104
  setting trace level 112
  setting up buddy lists 107
  specifying languages 199
  specifying log/chat/buddy list names 104
  specifying miscellaneous parameters 111
  specifying supported languages 105
  specifying URL for third party 197
  viewing the state summary 118
AgentStatistics class
  constructor 239, 425
  fields 240, 425
    See Also AgentStatistics class fields
AgentStatistics class fields
  averageCallDuration 240
  averageHoldDuration 240
  callsHandled 240
  data 240
  percentIdleTime 240
  percentTimeAvailable 241
  percentTimeInCurrentState 241
  percentTimeUnavailable 241
  timeCurrentLogin 241
  timeWorking 241
  totalTaskAcceptedLogin 241
  totalTaskDoneLogin 241
  totalTaskUnassignedLogin 242
  unavailableDuration 242
  waitDuration 242
AgentStatistics constructor 239
AgentStatistics object 425
AgentStatistics() constructor 425
alternate 327, 453
ani 248, 429
answer 328, 453
  ChatConnection class 433
  ChatConnection class methods 260
  EmailConnection class 440
  EmailConnection class methods 280
  GenericConnection class 446
  GenericConnection class methods 302
AOL Instant Messenger
  configuring servers 415
  detecting presence 415

Copyright © 1988-2007, Oracle. All rights reserved.

setting the server IP address 416
updating presence 418
appData
A2AChat class 235, 424
Chat class 252, 430
Email class 273, 437
GenericTask class 310, 449
AppData class 243
constructor 242, 426
fields 242, 426
See Also AppData class fields
methods 244, 427
AppData class fields
data 243
groupId 243
jr 243
question 243
strData 243
subject 243
uniqueId 243
url 244
userId 244
username 244
wizUrl 244
AppData constructor 242, 426
AppData object 426
application fundamentals xxxvii
application servers
chat architecture 8
See Also chat
configuring MCF servers 77
See Also MCF servers
describing MCF cluster architecture 78
logging 126
notifying MCF clusters of shutdown 102
specifying the authentication domain 65
understanding REN servers 55
See Also REN servers
using queue servers 84
See Also queue servers
Aspect 24, 47
attachUserData 328, 453
authentication domain 71
configuring REN servers 63, 65
authentication, server and client 58
Automatic Call Distributor (ACD)
configuring CTI queues 33
CTI architecture 14
See Also CTI
switching agent ready status 51
viewing the ACD group in the CTI console 47
availableCost 397, 469
Avaya Definity G3 24
averageCallDuration 240, 425
averageHoldDuration 240, 425
averageTaskDuration 324, 452
averageWaitTime 324, 452

B

backup queue servers 84
Balance Queue page 116
bcastinterval 95
binding, sockets 66
broadcast
configuring 137
configuring using JSMCAPI 139
implementing 139
publishing 139
subscribing 139
viewing logs 120
broadcast logs
viewing for third party 203
Broadcast page 120
broadcastSubscribe 365, 462
broadcastUnsubscribe 365, 462
browsers
authentication domain 63
browsers required for CTI 19
configuring REN server security 58
describing REN server cluster requests 57
MCF browser windows 8
MCF cluster browser URLs 83
MultiChannel Console 8
setting REN server browser URLs 71
using MCF cluster browser URLs 82
buddies 364, 461
Buddy class
callback event methods 428
constructor 245, 427, 428
event method 246
See Also Buddy class callback event methods
fields 427
Buddy class callback event methods
onStat1 246
onStat2 246

Copyright © 1988-2007, Oracle. All rights reserved.

onState 246
Buddy constructor 245, 427
Buddy List page 107
buddy lists
selecting buddies in the MultiChannel Console 130
setting up for agents 107
setting up for third party 193
Buddy object 427
buffer tests, REN server 71, 83
built-in functions, PeopleCode 175
Button Sample page 418

C

CA server certificate 472
call 326, 453
Line class 453
Call class
constructor 246
fields 247, 428
See Also Call class fields
Call class fields 247
line 247
statistics 248
Call constructor 247, 428
Call object 428
callback event
onNotify
_UQAddress class 422
onTaskRemoved 422
callback event methods
A2AChatAddress class 239, 425
_Address class 226, 421
Buddy class 246, 428
ChatAddress class 258, 432
ChatConnection class 264, 433
EmailAddress class 278, 439
EmailConnection class 282, 440
Extension class 292, 442
GenericAddress class 445
GenericConnection class 446
Group class 318, 450
Line class 340, 455
onAccepted 422
onAcceptingTask 422
onAddressRegistered 462
onAddressUnregistered 462
onAlternating 455
onAnswering 455
ChatConnection class 433
EmailConnection class 440
onAttachingUD 455
onBroadcast 461
onBuddyRegistered 462
onBuddyUnregistered 462
onCallDataChanged 455
onCancelingDnd 443
onCancelingForward 443
onCapabilitiesChanged 455, 466
ChatAddress class 432
ChatConnection class 433
EmailConnection class 440
GenericConnection class 446
onChatdataChanged
ChatConnection class 433
onChatEnded 425
onClearing 455
onClosed 462
onCompleted
EmailConnection class 440
GenericConnection class 446, 447
onCompleting 455
onConferencing 455
ChatConnection class 433
onCtiBusy 466
onCtiClear 466
onDequeueingTask 422
onDialing 455
ChatConnection class 434
onDnd 443
onDndCanceled 443
onDown 460
onDropped 455, 466
ChatConnection class 434
EmailConnection class 440
GenericConnection class 446
onError 455, 462
_Address class 421
ChatConnection class 434
EmailConnection class 440
GenericConnection class 446
User class 466
onForwardCanceled 443
onForwarded 443
ChatConnection class 434
EmailConnection class 440
GenericConnection class 446
onForwardError
ChatConnection class 434
EmailConnection class 440

Copyright © 1988-2007, Oracle. All rights reserved.

GenericConnection class 446
onForwarding 443
ChatConnection class 434
EmailConnection class 440
GenericConnection class 447
onGrabbing 455
onGroupRegistered 462
onGroupUnregistered 462
onHbLost 461
onHbRecovered 461
onHeld 455
onHistory
ChatConnection class 434
onHolding 455
onIncoming
EmailConnection class 440
GenericConnection class 447
onIncomingChat
ChatConnection class 434
onInfo 462, 466
onInitiatingChat 425
onInService 461
onJoining 455
onLoggedIn 466
onLoggedOut 466
onLoggingIn 466
onLoggingOut 466
onMediaDisabled 466
onMediaEnabled 466
onMessage
ChatConnection class 434
onMuted 455
onNotify
A2AChatAddress class 425
onNotReady 466
onOffHook 455
onOnHook 455
onOpened 463
onOutOfService 461
onParking 455
onPartyAdded 455
ChatConnection class 434
onPartyChanged 455
ChatConnection class 434
onPartyRemoved 455
ChatConnection class 434
onPresenceChanged 466
onProperties
ChatConnection class 434
onPushURL
ChatConnection class 434
onReady 466
onReconnecting 456
onRegistered 466
onRegistering 466
onRejected 456
ChatConnection class 434
EmailConnection class 440
GenericConnection class 447
onRejecting 456
onReleased
ChatConnection class 434
onReleasing 456
ChatConnection class 434
onRestart 461
onRetrieving 456
onRevoked
ChatConnection class 434
EmailConnection class 440
GenericConnection class 447
onRinging 456
onSetcallresult 456
onSetcallresultDNC 456
onSettingDnd 443
onSettingNotReady 466
onSettingPresence 466
onSettingReady 466
onSettingWorkNotReady 466
onSettingWorkReady 466
onStat
Group class 450
User class 467
onStat1
Buddy class 428
Group class 450
onStat1(
User class 467
onStat2
Buddy class 428
Group class 450
User class 467
onState 428
onTalking 456
ChatConnection class 434
onTaskAdded 450
_UQAddress class 422
onTaskRemoved 450
onTransfering 456
onTyping
ChatConnection class 434

Copyright © 1988-2007, Oracle. All rights reserved.

onUnassigned
_UQAddress class 422
onUnknown 467
onUnmuted 456
onUnregistered 467
onUnregistering 467
onUp 460
onUpdatingCD 456
onUserdataChanged
ChatConnection class 434
EmailConnection class 440
GenericConnection class 447
onUserDataChanged 456
onUserRegistered 463
onUserUnregistered 463
onWorkNotReady 467
onWorkReady 467
RenServer class 360, 460
Server class 362, 461
Session class 370, 462
_UQAddress class 228, 422
User class 386, 466
CallData class
constructor 248, 429
fields 248, 429
method 249
methods 429
CallData class fields
ani 248
callId 249
callType 249
data 249
dnis 249
CallData constructor 248, 429
CallData object 429
callId 249, 429
calls
answering 48
completing 52
CTI architecture 13
See Also CTI
dialing outbound 52
disconnecting callers 51
holding/retrieving 50
initiating conference calls 50
selecting actions in the CTI console 46
selecting agent status for incoming 51
selecting masks 28
testing outbound 41
transferring callers 48
understanding call actions 47
callsHandled 240, 425
CallStatistics class
constructor 250, 429
fields 250, 429
CallStatistics class fields
data 250
holdTime 251
queueTime 251
talkTime 251
CallStatistics constructor 250, 429
CallStatistics object 429
callType 249, 429
canAlternate 347, 456
canAnswer 347, 456
ChatConnectionCaps class 435
ChatConnectionCaps class fields 270
EmailConnection class fields 286
EmailConnectionCaps class 441
GenericConnectionCaps class 447
GenericConnectionCaps class fields 307
canAttachUserData 456
canAttachUserDate 347
canCancelDnd 294, 443
canCancelDndFwdDefault 443
cancelDnd 289, 442
cancelForwardSet 289, 442
canClear 348, 456
canComplete 348, 456
EmailConnection class fields 286
GenericConnectionCaps class fields 307
canConference 348, 457
ChatConnectionCaps class 435
ChatConnectionCaps class fields 270
EmailConnectionCaps class 441
GenericConnectionCaps class 447
canConferenceSingle 348, 457
ChatConnectionCaps class 435
ChatConnectionCaps class fields 270
canDial 294, 443
canDropParty 348, 457
canForward
ChatConnectionCaps class 435
ChatConnectionCaps class fields 270
EmailConnection class fields 286
EmailConnectionCaps class 441
GenericConnectionCaps class 447

Copyright © 1988-2007, Oracle. All rights reserved.

GenericConnectionCaps class fields 307
canFwdBusy 295, 443
canFwdBusyNoAnswer 295, 443
canFwdCancelForward 295, 443
canFwdDefault 295
canFwdNoAnswer 295, 443
canFwdUnconditional 295, 443
canHandleChat 467
UserCaps class fields 393
canHandleEmail 467
UserCaps class fields 393
canHandleGeneric 467
UserCaps class fields 393
canHandleVoice 467
UserCaps class fields 393
canHold 348, 457
canIndicateTyping
ChatConnectionCaps class 435
ChatConnectionCaps class fields 270
canLogin 393, 467
canLogout 393, 467
canMute 348, 457
canned messages 190
canned URL 190
canPark 349, 457
canPushURL
ChatConnectionCaps class 435
ChatConnectionCaps class fields 270
canReconnect 349, 457
canRefreshState 295, 394, 443, 467
canReject 349, 457
ChatConnectionCaps class 435
ChatConnectionCaps class fields 271
EmailConnection class fields 286
EmailConnectionCaps class 441
GenericConnectionCaps class 447
GenericConnectionCaps class fields 307
canRelease 349, 457
canRetrieve 349, 457
canSendDTMF 349, 457
canSendMessage
ChatConnectionCaps class 435
ChatConnectionCaps class fields 271
canSetcallresult 349, 457
canSetcallresultDNC 350, 457
canSetcallresultReschedule 350, 457
canSetDnd 296, 443
canSetNotReady 394, 467
canSetPresence 394, 467
canSetReady 394, 467
canSetWorkNotReady 394, 468
canSetWorkReady 394, 468
canTransfer 350, 457
canTransferMute 350, 457
canUnmute 350, 457
canUpdateCallData 350, 457
caps
_Address class 226, 421
ChatConnection class 433
ChatConnection class fields 259
EmailConnection class 439
EmailConnection class fields 279
GenericConnection class 446
GenericConnection class fields 300
Line class 326
_User class 231, 423
caseid 374, 463
certificate alias 67
certificates
importing in REN Java client 474
installing for local node 475
installing PSMCAPI certificates 476
channels
communicating by chat 6
See Also chat
email 401
See Also email channels
generic 7
instant messaging 7
See Also instant messaging
MultiChannel Console 8
understanding 5
voice 6
chat
activating 130
architecture 8
ChatAddress class 256, 432
ChatConnection class 433
ChatConnection class fields 259
collaborating by 6
communicating with agents 131
communicating with customers 133
defining responses 87
ending 133, 134
forwarding to other queues 133
initiating 130
instant messaging 415
See Also instant messaging

Copyright © 1988-2007, Oracle. All rights reserved.

inviting agents 133
logging 81, 98, 186
moving agents during ongoing sessions 114
personalizing 109
responding 133, 134
selecting task acceptance mode 109
sending template messages 133
sending/grabbing URLs 133
setting up buddy lists 107
specifying agent names 104
specifying greeting interval 92
using in MultiChannel Console 129
using the Customer Chat Sample page 148
viewing history 132
viewing logs 121
Chat class
constructor 251, 430
fields 252, 430
See Also Chat class fields
methods 254, 431
Chat class fields 252
address 252
agentId 252
appData 252
chatconnection 253
chatType 253
customerName 253
groupId 253
question 253
statistics 254
subject 253
userData 254
Chat class methods
gettpUrl 254
getUrl 254
Chat constructor 251, 430
Chat Log component (MCF_CHAT_LOG_CMP) 119
Chat object 430
Chat page 121
Chat Responses page 87
ChatAddress class
callback event methods 432
constructor 255, 431
event method 258
fields 255, 431
methods 256, 431
ChatAddress class callback event
onCapabilitiesChanged 258
ChatAddress class fields
chatconnections 256
ChatAddress class method
chat 256
getChatconnectionByConnectionId 256
getChatconnectionindexByConnectionId 257
getFreeChatconnection 257
getFreeChatconnectionIndex 257
ChatAddress constructor 255, 431
ChatAddress object 431
chatconnection
Chat class 253, 430
ChatConnection class
callback event methods 264, 433
constructor 258, 432
fields 259, 433
methods 260, 433
ChatConnection class callback event methods
onAccepted 264
onAnswering 264
onCapabilitiesChanged 265
onChatdataChanged 265
onConferencing 265
onDialing 265
onDropped 265
onError 266
onForwarded 266
onForwardError 266
onForwarding 266
onHistory 266
onIncomingChat 266
onMessage 267
onPartyAdded 267
onPartyChanged 267
onPartyRemoved 267
onProperties 267
onPushURL 268
onRejected 268
onReleased 268
onRereleasing 268
onRevoked 268
onTalking 268
onTyping 269
onUserdataChanged 269
ChatConnection class fields
caps 259
chat 259
connectionId 259

Copyright © 1988-2007, Oracle. All rights reserved.

id 259
state 259
ChatConnection class methods
answer 260
conference 260
forward 261
gethistory 261
getUrl 262
message 262
pushURL 263
reject 263
release 263
typing 264
ChatConnection constructor 432
ChatConnection object 432
ChatConnectionCaps class
constructor 269, 435
fields 270, 435
See Also ChatConnectionCaps class fields
ChatConnectionCaps class fields
canAnswer 270
canConference 270
canConferenceSingle 270
canForward 270
canIndicateTyping 270
canPushURL 270
canReject 271
canSendMessage 271
ChatConnectionCaps constructor 435
ChatConnectionCaps object 435
chatconnections
ChatAddress class 431
ChatAddress class fields 256
ChatData class
constructor 271, 435
fields 271, 435
See Also ChatData class fields
methods 271, 435
See Also ChatData class methods
ChatData class fields
data 271
ChatData class methods
addKeyValue 272
ChatData constructor 435
chatType 235, 424
Chat class 253
Cisco
components required for CTI 19
configuring the expanded call context variable 29
CTI architecture 13
See Also CTI
implementing 27
logging out 37
selecting a CTI vendor 24
selecting a switch 25
setting agent state masks 28
setting CTI client signature 34
setting masks 27
setting the heartbeat 28
setting up for pop-ups 39
transferring callers 49
Cisco switch limitations 44
clear 328, 454
clhbinterval 96
client authentication 58
close 366, 462
closeSession 354, 459
Cluster Notify component (MCF_AD_NOTIFY_CMP) 101
Cluster Summary component (MCF_RSERV_CFG_CMP) 93
Cluster Summary page 93
Cluster Tuning component (MCF_SYSTEM_NV_CMP) 93
Cluster Tuning page 94
cluster-wide broadcast 137
clusters
defining for third party 182
implementing MCF 77
See Also MCF clusters
notifying for third party 187
REN server 81
See Also REN servers
tuning for third party 184
viewing cluster IDs 189
code 358, 459
comments, submitting xlii
common elements xlii
complete 329, 454
EmailConnection class 440
EmailConnection class methods 280
GenericConnection class 446
GenericConnection class methods 302
computer telephony integration (CTI), *See* CTI
conference 329, 454
ChatConnection class 433

Copyright © 1988-2007, Oracle. All rights reserved.

ChatConnection class methods 260
conferenceSingle 330, 454
configuration ID 183, 189
Configuration page (instant messaging) 416
connectionid 326
connectionId 453
ChatConnection class 433
ChatConnection class fields 259
EmailConnection class 439
EmailConnection class fields 279
GenericConnection class 446
GenericConnection class fields 300
Connectors page 403
consoles
agent 6
See Also MultiChannel Console
CTI 13
See Also CTI console
Java 34
MultiChannel 129
See Also MultiChannel Console
constructor
A2AChat class 423
A2AChat Class 233
A2AChatAddress class 237, 424
_Address class 225, 421
AgentStatistics 239
AgentStatistics class 425
AppData class 242, 426
Buddy class 245, 427
Call class 246, 428
CallData class 248, 429
CallStatistics class 429
CallStatisticsclass 250
Chat class 251, 430
ChatAddress 255
ChatAddress class 431
ChatConnection 258
ChatConnection class 432
ChatConnectionCaps class 269, 435
ChatData class 271, 435
Email class 272, 437, 438
EmailAddress class 275
EmailConnection class 278, 439
EmailConnectionCaps class 285, 441
EmailData class 286, 441
Extension class 287, 442
ExtensionCaps class 294, 443
ForwardMode class 296, 444
GenericAddress class 297, 444
GenericConnection class 300, 446
GenericConnectionCaps class 306, 447
GenericData class 308, 447
GenericTask class 309, 448
Group class 316, 450
GroupStatistics class 319, 451
GroupStatistics1 class 321, 451
GroupStatistics2 class 324, 452
Line class 325, 453
LineCaps class 346, 456
MCEvent class 350, 457
MediaType class 352, 458
PSMC class 353, 458
Reason class 357, 459
RenServer class 359, 460
Server class 360, 460
Session class 363, 461
Task class 373, 463
TaskStatistics class 376, 464
_UQAddress class 227, 422
User class 377, 464
_User class 230, 422
UserCaps class 392, 393, 467
See Also UserCaps class fields
UserData class 394, 468
UserStatistics1 class 396, 469
UserStatistics2 class 399, 469
contact information xlii
cost 374, 463
cross-references xli
CTI
accessing 45
agents 22
See Also CTI agents
answering calls 48
applying applet protection 19
Cisco components, required 19
completing calls 52
components 14
configuring pop-up windows 38
configuring queues 32
configuring using CTI server 19
configuring using third party 19
creating configurations 22
describing Cisco switch limitations 44
dialing outbound calls 52
disconnecting callers 51
enabling 22
event logging 40

Copyright © 1988-2007, Oracle. All rights reserved.

Genesys components, required 18
holding calls 50
implementing Cisco 27
See Also Cisco
implementing free seating 40
implementing Genesys 26
See Also Genesys
initiating conference calls 50
installing 19, 21
JRE, required 19
log 99
miscellaneous 19
monitoring 96
personalizing 35
phone books, shared 19
registering with middleware 46
setting pop-up screens 30
supporting single signon 40
testing calls 41
transferring callers 48
understanding 13, 43
understanding call actions 47
understanding components 13
using 44
using console 46
See Also CTI console
using hot keys 53
using outbound call sample page 41
using phone books 34
using phone books, shared 25
using sample page 41
using the interface 44
CTI Agent Configuration component (PT_CTI_AGENT) 32
CTI Agent Configuration page 33
CTI agents
configuring 33
implementing wrap-up mode 51
logging out 37, 47
phone books, shared 25
signing on to PeopleSoft 22
switching agent ready status 51
using phone books 34
viewing information 37
viewing status 46
CTI Applet page 22
CTI Busy 165
CTI Cisco page 27
CTI configuration component (PT_CTI_CONFIG) 19
CTI console 14
applet 19, 24
browser requirements 19
components 13
floating 37
heartbeat interval 96
JSMCAPI 19
non-applet 19, 23
synchronizing with teleset 49
understanding 13
using 46
CTI Genesys page 26
CTI Non-Applet page 22
CTI queues, configuring 32
CTI Sample console
using 160
CTI Sample Pages component (PT_CTI_DEMOOUTB) 155
CTI Sample Pages component (PT_CTI_DEMOUTB) 41
CTI Type page 22
ctiBusy 397, 469
ctiBusyUq 379
currentQueue 397, 399, 469, 470
Customer Chat page 148
customer chat window 134
Customer Connection website xxxviii
customerid 374, 463
customerName 235, 424
Chat class 253, 430
Email class 273, 437
GenericTask class 310, 449
customers
accepting tasks in chat 109
chat architecture 8
See Also chat
communicating by chat 6
communicating via chat 133
using CTI 43
See Also CTI
using the Customer Chat Sample page 148

D

data
AgentStatistics class 240, 425
AppData class 243, 426
CallData class 249, 429
CallStatistics class 250, 429
ChatData class 435

Copyright © 1988-2007, Oracle. All rights reserved.

ChatData class fields 271
EmailData class 441
EmailData class fields 287
GenericData class 448
GenericData class fields 308
GroupStatistics class 319, 451
TaskStatistics class 464
TaskStatistics class fields 376
debugging
logging for application servers 126
REN servers 64
tracing level for agents 112
troubleshooting CTI 34
viewing event logs 123
default_http_port 65
default_https_port 65
defining
MCF queues 86
Definity G3, Avaya 24
DeQueue
demonstrating 153, 154
understanding 175
dequeueTask 228, 422
desc 358
description 459
dial 330, 454
digital certificates
configuring 472
describing 58
installing for REN SSL 471
digital signatures 19
disableMedia 379, 465
DND 166
DND (do not disturb) 209
dnis 249, 429
documentation
printed xxxviii
related xxxviii
updates xxxviii
domain
authenticating domain 71
domains
configuring MCF server 77
See Also MCF
describing application server domains 78
See Also application servers
describing MCF cluster architecture 78
setting screen pop-up URLs 30
setting the authentication token domain for REN servers 63
See Also authentication domain
done list, size 96
donelistsize 96
dropParty 330, 454
DTMF info 209
DTMFInfo 166
dumpagents 96
dumpinterval 97

E

ECC variables 39
email
accessing/deleting from the database 412
authorizing attachments for viewing/deleting 412
channels 401
See Also email channels
defining connector and mail server properties 404
email IDs 413
EmailConnection class 439
EmailConnection class fields 279
GetMail - Server page 410
handling headers, size determination and time zone offsets 402
HTML 7
instant messaging 415
See Also instant messaging
MailStore - DB page 412
MIME 410
See Also Multipurpose Internet Mail Extensions (MIME)
processing architecture 10
reading/deleting from mail servers 410
setting access, attachment, language and threshold properties 404
storing in/retrieving from database servers 410
understanding 7
unique identifiers (UIDs) 412
universal queue 7
using sample pages 153
using the Email window 154
email channels
configuring GETMAILTARGET 403
configuring Integration Broker 403
demonstrating 410

Copyright © 1988-2007, Oracle. All rights reserved.

understanding 401
Email class
constructor 272, 437, 438
fields 272, 437
See Also Email class fields
methods 275, 438
See Also Email class methods
Email class fields 272
address 273
agentId 273
appData 273
customerName 273
emailconnection 274
emailId 274
groupId 274
question 274
statistics 274
subject 274
userData 274
Email class methods
gettpUrl 275
getUrl 275
Email constructor 272, 437
Email object 435, 437, 441
Email page 153
Email Sample Pages component (MCFEM_DEMOERMS_CMP) 410
Email window 154
EmailAddress class
callback event methods 278, 439
constructor 275
fields 276, 438
See Also EmailAddress class fields
methods 277, 438
See Also EmailAddress class methods
EmailAddress class fields
agent 276
emailconnections 276
EmailAddress class methods
getEmailconnectionByConnectionId 277
getEmailconnectionindexByConnectionId 277
getFreeEmailconnection 277
EmailAddress constructor 276, 438
EmailAddress object 438
emailconnection
Email class 274, 437
EmailConnection class
callback event methods 282, 440
See Also EmailConnection class callback event methods
constructor 278, 439
fields 278, 285, 439
See Also EmailConnection class fields
methods 280, 440
See Also EmailConnection class methods
EmailConnection class callback event methods
onAnswering 282
onCapabilitiesChanged 282
onCompleted 283
onDropped 283
onEmaildataChanged 283
onError 283
onForwarded 283
onForwardError 283
onForwarding 284
onIncoming 284
onProcessing 284
onRejected 284
onRevoked 284
onUserdataChanged 285
onWithdraw_REQ 285
EmailConnection class fields
canAnswer 286
canComplete 286
canForward 286
canReject 286
caps 279
connectionId 279
email 279
id 279
state 279
EmailConnection class methods
abandon 280
answer 280
complete 280
forward 281
reject 281
withdraw_RES 282
EmailConnection constructor 439
EmailConnection object 439
EmailConnectionCaps class
constructor 285, 441
fields 441
EmailConnectionCaps constructor 441
EmailConnectionCaps object 441
emailconnections
Email class 438
EmailAddress class fields 276

Copyright © 1988-2007, Oracle. All rights reserved.

EmailData class
constructor 286, 441
fields 287, 441
See Also EmailData class fields
methods 287, 441
See Also EmailData class methods
EmailData class fields
data 287
EmailData class methods
addKeyValue 287
EmailData constructor 441
emailId
Email class 274, 437
enableMedia 380, 465
enableMedialogout 382
enableMediasetWorkNotReady 385
EnQueue
accepting tasks in chat 109
demonstrating generic persistent events 151
setting agent languages 105
understanding 175
errors
determining email size 402
logging for application servers 126
REN servers 64
troubleshooting CTI 34
escalation
administering tasks 128
specifying the default timeout for tasks 92
Escalation Admin (administration) page 128
Escalation Administration component (MCF_ESCAL_CMP) 127
event, *See* callback event methods
A2AChatAddress class callback 239
_Address class callback 226
Buddy class callback 246
ChatAddress class callback 258
Extension class callback 292
Group class callback 318
Line class callback 340
RenServer class callback 360
Server class callback 362
Session class callback 370
_UQAddress class callback 228
User class callback 386
Event Log component (MCF_EVENTLOGL_CMP) 119
event logs
viewing for third party 201
events
logging 40
MCF log server failover 81
types 124
understanding real-time event notification 55
See Also REN servers
using Generic Event page 151
using Generic Event window 152
viewing event logs 123
Expanded Call Context (ECC) variables 39
extension 351, 458
Extension class
callback event methods 442
constructor 287, 442
event method 292
See Also Extension class callback event methods
fields 288, 442
See Also Extension class fields
method 289
methods 442
Extension class callback event methods
onCancelingDnd 292
onCancelingForward 292
onDnd 292
onDndCanceled 293
onForwardCanceled 293
onForwarded 293
onForwarding 293
onSettingDnd 293
Extension class fields 288
forwardMode 288
isDnd 288
lines 288
numOfLines 289
Extension class method
cancelDnd 289
cancelForwardSet 289
forwardSet 290
getDialingLine 290
getFreeLine 290
getLineByConnectionId 291
getOffHookLine 291
setDnd 292
Extension constructor 287, 442
Extension object 442

Copyright © 1988-2007, Oracle. All rights reserved.

ExtensionCaps class
constructor 443
fields 294, 443
See Also ExtensionCaps class fields
ExtensionCaps class fields
canCancelDnd 294
canDial 294
canFwdBusy 295
canFwdBusyNoAnswer 295
canFwdCancelForward 295
canFwdDefault 295
canFwdNoAnswer 295
canFwdUnconditional 295
canRefreshState 295
canSetDnd 296
Extensioncaps constructor 443
ExtensionCaps constructor 294
ExtensionCaps object 443

F

failover
configuring REN servers 55
REN server clusters 57
understanding 77
using MCF log servers 81
using queue servers 79
fields
A2AChat class 424
A2AChat Class 234
See Also A2AChat class fields
A2AChat.PS_JR 449
A2AChat.TYPE_ANSWER 449
A2AChat.TYPE_CONSULT 449
A2AChatAddress class 237, 424
See Also A2AChatAddress class fields
address
A2AChat class 424
Chat class 430
Email class 437
GenericTask class 448
_Address class 226, 421
See Also _Address class fields
addresses
Session class 461
User class 465
agent
Email class 438
GenericAddress class 445
agentId
A2AChat class 424
Chat class 430
Email class 437
GenericTask class 448
_User class 423
agentName 424
agentPassword 465
AgentStatistics class 240, 425
ani 429
appData
A2AChat class 424
Chat class 430
Email class 437
GenericTask class 449
AppData class 242, 426
availableCost 469
averageCallDuration 425
averageHoldDuration 425
averageTaskDuration 452
averageWaitTime 452
buddies 461
Buddy class 427
call 453
Line class 453
Call class 247, 428
CallData class 248, 429
See Also CallData class fields
callId 429
callsHandled 425
CallStatistics class 250, 429
See Also CallStatistics class fields
callType 429
canAlternate 456
canAnswer 456
ChatConnectionCaps class 435
EmailConnectionCaps class 441
GenericConnectionCaps class 447
canAttachUserData 456
canCancelDnd 443
canClear 456
canComplete 456
canConference 457
ChatConnectionCaps class 435
EmailConnectionCaps class 441
GenericConnectionCaps class 447
canConferenceSingle 457
ChatConnectionCaps class 435
canDial 443
canDropParty 457
canForward
ChatConnectionCaps class 435

Copyright © 1988-2007, Oracle. All rights reserved.

EmailConnectionCaps class 441
GenericConnectionCaps class 447
canFwdBusy 443
canFwdBusyNoAnswer 443
canFwdCancelForward 443
canFwdDefault 443
canFwdNoAnswer 443
canFwdUnconditional 443
canHandleChat 467
canHandleEmail 467
canHandleGeneric 467
canHandleVoice 467
canHold 457
canIndicateTyping
ChatConnectionCaps class 435
canLogin 467
canLogout 467
canMute 457
canPark 457
canPushURL
ChatConnectionCaps class 435
canReconnect 457
canRefreshState 443, 467
canReject 457
ChatConnectionCaps class 435
EmailConnectionCaps class 441
GenericConnectionCaps class 447
canRelease 457
canRetrieve 457
canSendDTMF 457
canSendMessage
ChatConnectionCaps class 435
canSetcallresult 457
canSetcallresultDNC 457
canSetcallresultReschedule 457
canSetDnd 443
canSetNotReady 467
canSetPresence 467
canSetReady 467
canSetWorkNotReady 468
canSetWorkReady 468
canTransfer 457
canTransferMute 457
canUnmute 457
canUpdateCallData 457
caps
_Address class 421
ChatConnection class 433
EmailConnection class 439
GenericConnection class 446
_User class 423
caseid 463
chat
ChatConnection class 433
Chat class 252, 430
ChatAddress 255
ChatAddress class 431
chatconnection
Chat class 430
ChatConnection 259
ChatConnection class 433
ChatConnectionCaps class 270, 435
chatconnections
ChatAddress class 431
ChatData class 271, 435
chatType 424
code 459
connectionId 453
ChatConnection class 433
EmailConnection class 439
GenericConnection class 446
cost 463
ctiBusy 469
currentQueue 469, 470
customerid 463
customerName 424
Chat class 430
Email class 437
GenericTask class 449
data
AgentStatistics class 425
AppData class 426
CallData class 429
CallStatistics class 429
ChatData class 435
EmailData class 441
GenericData class 448
GroupStatistics class 451
TaskStatistics class 464
description 459
dnis 429
email
EmailConnection class 439
Email class 272, 437
EmailAddress class 276, 438
emailconnection
Email class 437
EmailConnection class 278, 285, 287, 439
EmailConnectionCaps class 441

Copyright © 1988-2007, Oracle. All rights reserved.

emailconnections
Email class 438
EmailData class 441
emailId
Email class 437
extension 458
Extension class 288, 442
ExtensionCaps class 294, 443
forwardMode 442
ForwardMode class 296, 444
generic
GenericConnection class 446
GenericAddress class 297, 309, 444
genericconnection
GenericTask class 449
GenericConnection class 300, 446
GenericConnectionCaps class 307, 447
genericconnections
GenericAddress class 445
GenericData class 308, 448
genericId
GenericTask class 449
GenericTask class 448
gettpUrl
GenericTask class 449
getUrl
GenericTask class 449
GLOBALS class 312, 449
group 458, 463
Group class 317, 450
groupId 426, 430
Email class 437
GenericTask class 449
groups 461
GroupStatistics class 319, 451
GroupStatistics1 class 322, 452
GroupStatistics2 class 324, 452
holdTime 429
TaskStatistics class 464
id
A2AChat class 424
A2AChatAddress class 425
_Address class 421
ChatConnection class 433
EmailConnection class 439
GenericConnection class 446
Group class 450
Line class 453
Server class 460
Session class 461
Task class 463
_User class 423
info
Server class 460
isConference
A2AChat class 424
isDnd 442
isMuted
Line class 453
isRunning 460
jr
A2AChat class 424
AppData class 426
language 465
line 428
Line class 326, 453
LineCaps class 347, 456
lines 442
listOfTasksInTheQueueByTaskType
GroupStatistics class 451
maxTaskCompletionTime 451
MCEvent class 351, 458
MediaType class 352, 458
mode 444
mostRecentTaskData 469
mostRecentTaskDone 452
mostRecentTaskDoneData 452
mostRecentTaskEnqueued 452
mostRecentTaskEnqueuedData 452
mostRecentTaskId 469
name
Group class 450
_User class 423
newestTask 451
newestTaskCompletionTime 451
numAgentsAvailable 452
numAgentsLoggedIn 452
numberOfAbandoned 451
numberOfLoggedIn 451
numberOfQueued 451
numEscalation 452
numOfLines 442
numOverflow 452
numTaskAccepted 452
UserStatistics1 class 469
numTaskDone 452
numTasksDone 469
numTasksUnassigned 469
numUnassignedTasks 451
oldestTask 452

Copyright © 1988-2007, Oracle. All rights reserved.

onStat 463
percentIdleTime
  AgentStatistics class 425
percentTimeAvailable
  AgentStatistics class 425
percentTimeInCurrentState
  AgentStatistics class 426
percentTimeUnavailable
  AgentStatistics class 426
presences 423
presenceText 469
priority 463
PSMC class 353, 458
question
  AppData class 426
  Chat class 430
  Email class 437
  GenericTask class 449
queuedWaitTime 451
queueTime 429
  TaskStatistics class 464
queueUpTime 451
reason 458
Reason class 358, 459
reasonData1 459
reasonData2 459
reasonData3 459
reasonFlag
  GroupStatistics1 class 452
  UserStatistics1 class 469
recentTask 452
registered
  Group class 450
registeredAddresses 465
relativeQueueLoad 451
renServer 458
RenServer class 359, 460
Server class 361, 460
Server.TYPE.CTI 449
Server.TYPE_UQ 449
serverID 461
servers 458
Session class 364, 461
sessions 458
ST_LOGGEDIN 423
ST_LOGGEDOUT 423
ST_NOTREADY 423
ST_READY 423
ST_UNKNOWN 423
ST_WORKNOTREADY 423
ST_WORKREADY 423
state
  ChatConnection class 433
  EmailConnection class 439
  GenericConnection class 446
  Line class 453
  Server class 460
  Session class 461
  UserStatistics1 class 469
states 423
statesUq 465
statistics
  Call class 428
  Chat class 430
  Email class 437
  Group class 450
  _User class 423
statistics1
  Group class 450
  _User class 423
statistics2
  Group class 450
  _User class 423
strData 426
subject
  AppData class 426
  Chat class 431
  Email class 437
  GenericTask 449
talkTime 429
  TaskStatistics class 464
Task class 374, 463
Task.TYPE_A2ACHAT 449
Task.TYPE_CHAT 449
Task.TYPE_CTI 449
Task.TYPE_EMAIL 449
Task.TYPE_GENERIC 449
taskAcceptedCurrentLogin
  AgentStatistics class 426
tasks
  A2AChatAddress class 425
  ChatAddress class 431
  _UQAddress class 422
TaskStatistics class 376, 464
taskTotalTimeInSystem 452
timeCurrentLogin
  AgentStatistics class 426
timeElapsedOldestTask 451, 453
timeElapsedRecentTask 453
timeIdle 470

Copyright © 1988-2007, Oracle. All rights reserved.

timeInCurrentState 469, 470
timeNotReady 470
timeSinceLoggedIn 469
timeSinceLogin 470
timeSinceStart 452
timeWorking
AgentStatistics class 426
totalTaskAcceptedLogin
AgentStatistics class 426
totalTaskDoneLogin
AgentStatistics class 426
totalTimeAvailable 470
totalTimeUnavailable 470
type
A2AChat class 424
MediaType class 458
Server class 460
Task class 463
unavailableDuration 426
uniqueId 426
uniqueueId 424
_UQAddress class 227, 422
url 426, 460
urlAbs 463
urlRel 463
user 458
Session class 461
User class 377, 464
_User class 231, 423
See Also _User class fields
UserCaps class 467
userData 428
Chat class 431
Email class 437
GenericTask 449
UserData class 395, 397
See Also UserStatistics1 class fields
userId 427
username 427
UserStatistics1 class 469
UserStatistics2 class 399, 470
See Also UserStatistics2 class fields
waitDuration 426
wizUrl 427
flag, third party 181
floating CTI console 35, 37
forward 209
ChatConnection class 433
ChatConnection class methods 261
EmailConnection class 440
EmailConnection class methods 281
GenericConnection class 446
GenericConnection class methods 302
Forward 166
Forward function
demonstrating 154
understanding 175
forwardMode 288, 442
ForwardMode class
constructor 296, 444
fields 296, 444
ForwardMode constructor 296, 444
ForwardMode object 444
forwardSet 290, 442
free seating 40

G

Gateways page 403
gateways, configuring for email channels 403
generic
GenericConnection class 446
GenericConnection class fields 300
Generic Event page 151
Generic Event window 152
Generic object 447
GenericAddress class
callback event methods 299, 445
constructor 297, 444
fields 297, 309, 444
See Also GenericAddress class fields; GenericTask class Fields
methods 298, 445
See Also GenericAddress class methods
GenericAddress class fields
agent 298
genericconnections 298
GenericAddress class methods
getFreeGenericconnection 298
getGenericconnectionByConnectionId 298
getGenericconnectionindexByConnectionId 29
GenericAddress constructor 297, 444
GenericAddress object 444
genericconnection
GenericTask class 310, 449
GenericConnection class
callback event methods 304, 446
See Also GenericConnection class callback event methods

Copyright © 1988-2007, Oracle. All rights reserved.

constructor 300, 446
fields 300, 446
See Also GenericConnection class fields
methods 301, 446
See Also GenericConnection class methods
GenericConnection class callback event methods
onCapabilitiesChanged 304
onCompleted 304
onDropped 304
onError 304
onForwarded 305
onForwardError 305
onForwarding 305
onGenericdataChanged 305
onIncoming 305
onProcessing 305
onRejected 306
onRevoked 306
onUserdataChanged 306
onWithdraw_REQ 306
GenericConnection class fields
caps 300
connectionId 300
generic 300
id 301
state 301
GenericConnection class methods
abandon 301
answer 302
complete 302
forward 302
reject 303
withdraw_RES 303
GenericConnection constructor 446
GenericConnection object 446
GenericConnectionCaps class
constructor 306, 447
fields 307, 447
See Also GenericConnectionCaps class fields
GenericConnectionCaps class fields
canAnswer 307
canComplete 307
canForward 307
canReject 307
GenericConnectionCaps constructor 447
GenericConnectionCaps object 447
genericconnections
GenericAddress class 445
GenericAddress class fields 298
GenericData class
constructor 308, 447
fields 308, 448
See Also GenericData class fields
methods 308, 448
See Also GenericData class methods
GenericData class fields
data 308
GenericData class methods
addKeyValue 308
GenericData constructor 447
genericId
GenericTask class 311, 449
GenericTask class
constructor 309, 448
fields 448
method 311
See Also GenericTask class methods
methods 449
GenericTask class fields 309
address 310
agentId 310
appData 310
customerName 310
genericconnection 310
genericId 311
groupId 311
question 311
statistics 311
subject 311
userdata 311
GenericTask class methods
gettpUrl 312
getUrl 312
GenericTask constructor 309, 448
GenericTask object 448
Genesys
components required for CTI 18
CTI architecture 13
See Also CTI
defining wrap-up time 52
implementing 26
logging out 47
selecting a CTI vendor 24
selecting a switch 24
setting up for pop-ups 38
getAni 331, 454

Copyright © 1988-2007, Oracle. All rights reserved.

getCallById 354, 459
getChatById 355, 459
getChatconnectionByConnectionId
ChatAddress class 256, 432
getChatconnectionindexByConnectionId
ChatAddress class 257, 432
getDescr 331, 454
getDialingLine 290, 442
getDnis 331, 454
getEmailById 355, 459
getEmailconnectionByConnectionId
EmailAddress class 439
EmailAddress class methods 277
getEmailconnectionindexByConnectionId
EmailAddress class 439
EmailAddress class methods 277
getFreeChatconnection
ChatAddress class 257, 432
getFreeChatconnectionIndex
ChatAddress class 257, 432
getFreeEmailconnection
EmailAddress class 439
EmailAddress class methods 277
getFreeGenericconnection
GenericAddress class 445
GenericAddress class methods 298
getFreeLine 290, 442
getGenericconnectionByConnectionId
GenericAddress class 445
GenericAddress class methods 298
getGenericconnectionindexByConnectionId
GenericAddress class 445
GenericAddress class methods 299
getGenericTaskById 356, 459
gethistory
ChatConnection class 433
ChatConnection class methods 261
getLineByConnectionId 291, 442
getLineById 356, 459
GetMail - Server page 410
GETMAILTARGET, configuring 403
getOffHookLine 291, 442
getPadvalue 332
getReferenceId 332, 454
gettpUrl
Chat class 431
Email class 438
Email class methods 275
GenericTask class 449
GenericTask class methods 312
getUrl
A2AChat class 424
Chat class 254, 431
ChatConnection class 433
ChatConnection class methods 262
Email class 438
Email class methods 275
GenericTask class 312, 449
Line class 332, 454
getURL
A2AChat class 237
GLOBALS class
fields 312, 449
See Also GLOBALS class fields
method 314
methods 450
GLOBALS class fields
A2AChat.PS_JR 312
A2AChat.TYPE_ANSWER 313
A2AChat.TYPE_CONSULT 313
Server.TYPE_CTI 313
Server.TYPE_UQ 313
Task.TYPE_A2ACHAT 313
Task.TYPE_CHAT 314
Task.TYPE_CTI 314
Task.TYPE_EMAIL 314
Task.TYPE_GENERIC 314
GLOBALS class methods
initJSMCAPI 315
isValid 315
MCFBroadcast 315
glossary 479
grabCall 333, 454
group 458, 463
MCEvent class 351
Task class 374
Group class 317
callback event methods 450
constructor 316, 450
event method 318
See Also Group class callback event methods
fields 317, 450
See Also Group class fields
Group class callback event methods
onStat 318
onStat1 318
onStat2 318
onTaskAdded 318
onTaskRemoved 319

Copyright © 1988-2007, Oracle. All rights reserved.

Group class fields
id 317
name 317
registered 317
statistics 317
statistics1 317
statistics2 318
Group constructor 316, 450
groupId 426, 430
AppData class 243
Chat class 253
Email class 274, 437
GenericTask class 311, 449
groups 364, 461
GroupStatistics class
constructor 319, 451
fields 319, 451
See Also GroupStatistics class fields
GroupStatistics class fields
data 319
listOfTasksInTheQueueByTaskType 320
maxTaskCompletionTime 320
newestTask 320
newestTaskCompletionTime 320
numberOfAbandoned 320
numberOfLoggedIn 320
numberOfQueued 320
numUnassignedTasks 320
queuedWaitTime 321
queueUpTime 321
relativeQueueLoad 321
timeElapsedOldestTask 321
GroupStatistics constructor 319, 451
GroupStatistics object 451
GroupStatistics1 class 452
constructor 321, 451
fields 322, 452
See Also GroupStatistics1 class fields
GroupStatistics1 class fields
mostRecentTaskDone 322
mostRecentTaskDoneData 322
mostRecentTaskEnqueued 322
mostRecentTaskEnqueuedData 322
numAgentsAvailable 322
numAgentsLoggedIn 322
numEscalation 323
numOverflow 323
numTaskAccepted 323
numTaskDone 323
numTaskQueued 323
reasonFlag 323
taskTotalTimeInSystem 323
timeSinceStart 324
GroupStatistics1 constructor 321, 451
GroupStatistics1 object 451
GroupStatistics2 class
constructor 324, 452
fields 324, 452
See Also GroupStatistics2 class fields
GroupStatistics2 class fields
averageTaskDuration 324
averageWaitTime 324
oldestTask 325
recentTask 325
timeElapsedOldestTask 325
timeElapsedRecentTask 325
GroupStatistics2 constructor 324, 452
GroupStatistics2 object 452

H

heartbeat interval, JSMCAPI client 96
heartbeat, setting Cisco 28
Hicom 300, Siemens 24
highwater 97
hold 333, 454
hold status 50
holdTime 251, 429, 464
TaskStatistics class fields 376
hot keys
using CTI hot keys 53
using MultiChannel Console 131

I

id 231, 317
A2AChat class 235, 424
A2AChatAddress class 238, 425
_Address class 226, 421
ChatConnection class 433
ChatConnection class fields 259
EmailConnection class 439
EmailConnection class fields 279
GenericConnection class 446
GenericConnection class fields 301
Group class 450
Line class 326, 453
Server class 361, 460
Session class 364, 461
Task class 374, 463
_User class 423

Copyright © 1988-2007, Oracle. All rights reserved.

IMAP4, *See* Internet Message Access Protocol 4 (IMAP4)
implementation
  MultiChannel Framework 1
info 361
  Server class 460
InitChat
  demonstrating 148
  setting agent languages 105
  understanding 175
initiateChat 238, 425
initJSMCAPI 315, 450
instant messaging
  AOL Instant Messenger 415, 416
    See Also AOL Instant Messenger
  configuring 416
  configuring servers 415
  demonstrating 417, 418
  integrating 415
  Lotus Sametime Connect 415
    See Also Lotus Sametime Connect
  understanding 7, 415
    See Also chat
  using sample pages 417
  Yahoo! Messenger 415, 416
    See Also Yahoo! Messenger
Instant Messaging Configuration component (MCF_IM_CFG_CMP) 416
Instant Messaging Sample Pages component (MCF_IM_DEMO_CMP) 417
Integration Broker
  configuring for email channels 403
  setting the email size threshold 407
Interactive Voice Response (IVR)
  CTI architecture 14
    See Also CTI
Internet Message Access Protocol 4 (IMAP4)
  calculating email size 410
  setting the connector threshold 407
io_buffer_size 64
isConference
  A2AChat class 236, 424
isDnd 288, 442
isMediaEnabled 380, 465
isMuted
  Line class 327, 453
isRunning 359, 460
isValid 315, 450
IVR, *See* Interactive Voice Response (IVR)

J

Java Console 34
Java Runtime Environment (JRE), *See* JRE
JavaScript MultiChannel API (JSMCAPI), *See* JSMCAPI
join 334
jr 243
  A2AChat class 236, 424
  AppData class 426
JRE 19
JSMCAPI 14, 18
  implementing 155
JSMCAPI broadcast
  configuring using MCF Supervisor console 139
  understanding 138
JSMCAPI Broadcast page 139
JSMCAPI classes
  A2AChat class 423
  A2AChatAddress class 424
  _Address class 421
  AgentStatistics class 425
  AppData class 426
  Buddy class 427
  Call class 428
  CallData class 429
  CallStatistics class 429
  Chat class 430
  ChatAddress class 431
  ChatConnection class 432
  ChatConnectionCaps class 434
  ChatData class 435
  Email class 436
  EmailAddress class 438
  EmailConnection class 439
  EmailConnectionCaps class 440
  EmailData class 441
  Extension class 441
  ExtensionCaps class 443
  ForwardMode class 444
  GenericAddress class 444
  GenericConnection class 445
  GenericConnectionCaps class 447
  GenericData class 447
  GenericTask class 448
  GLOBALS class 449
  Group class 450
  GroupStatistics class 451
  GroupStatistics1 class 451

Copyright © 1988-2007, Oracle. All rights reserved.

GroupStatistics2 class 452
Line class 453
LineCaps class 456
MCEvent class 457
MediaType class 458
PSMC class 458
Reason class 459
RenServer class 459
Server class 460
Session class 461
Task class 463
TaskStatistics class 464
_UQAddress class 421
User class 464
_User class 422
UserCaps class 467
UserData class 468
UserStatistics1 class 468
UserStatistics2 class 469

L

language 378, 465
languages
setting agent support for 105
setting for email 408
Languages page 105
line 247, 428
Line class
callback event methods 455
constructor 325, 453
event method 340
See Also Line class callback event methods
fields 326, 453
See Also Line class fields
method 327
methods 453
Line class callback event methods
onAlternating 340
onAnswering 340
onAttachingUD 340
onCallDataChanged 340
onCapabilitiesChanged 341
onClearing 341
onCompleting 341
onConferencing 341
onDialing 341
onDropped 342
onError 342
onGrabbing 342
onHeld 342
onHolding 342
onJoining 342
onMuted 343
onOffHook 343
onOnHook 343
onParking 343
onPartyAdded 343
onPartyChanged 344
onPartyRemoved 344
onReconnecting 344
onRejected 344
onRejecting 344
onReleasing 344
onRetrieving 345
onRinging 345
onSetcallresult 345
onSetcallresultDNC 345
onSetcallresultReschedule 345
onTalking 346
onTransfering 346
onUnmuted 346
onUpdatingCD 346
onUserDataChanged 346
Line class fields
call 326
caps 326
connectionid 326
id 326
isMuted 327
state 327
Line class methods
alternate 327
answer 328
attachUserData 328
clear 328
complete 329
conference 329
conferenceSingle 330
dial 330
dropParty 330
getAni 331
getDescr 331
getDnis 331
getPadvalue 332
getReferenceId 332
getUrl 332
grabCall 333
hold 333
join 334

Copyright © 1988-2007, Oracle. All rights reserved.

mute 334
park 334
reconnect 335
reject 335
release 336
retrieve 336
sendDTMF 336
setcallresult 337
setcallresultDNC 337
setcallresultReschedule 338
transfer 338
transferMute 338
unmute 339
updateCallData 339
Line constructor 325, 453
Line object 453
LineCaps class
constructor 346, 456
fields 347, 456
See Also LineCaps class fields
LineCaps class fields
canAlternate 347
canAnswer 347
canAttachUserDate 347
canClear 348
canComplete 348
canConference 348
canConferenceSingle 348
canDropParty 348
canHold 348
canMute 348
canPark 349
canReconnect 349
canReject 349
canRelease 349
canRetrieve 349
canSendDTMF 349
canSetcallresult 349
canSetcallresultDNC 350
canSetcallresultReschedule 350
canTransfer 350
canTransferMute 350
canUnmute 350
canUpdateCallData 350
LineCaps constructor 347, 456
LineCaps object 456
lines 288, 442
listOfTasksInTheQueueByTaskType
GroupStatistics class 320, 451
local node 475
log_broadcast 98, 186
log_chat_ses 98, 186
log_cti 99
logDMPQ 98, 185
LogFence settings 126
logging
agent status 96
chat 98, 132, 186
CTI 99
CTI events 40
MCF log server failover 81
queue server status 97
REN servers 64, 98, 185
specifying agent names for logs 104
statistics 98, 186
tracing level for agents 112
using application servers 126
viewing broadcast logs 120
viewing chat logs 121
viewing event logs 123
logical queues
assigning agents 104
balancing agents/tasks across 116
deleting 86
implementing scalability 81
moving agents 114
moving physical queues within 114
understanding 80
See Also queues
viewing physical queue IDs 86
login 381, 465
loginUq 381, 465
logout 465
logoutUq 382, 465
logStat 98, 186
Lotus Sametime Connect
configuring instant messaging servers 415
detecting presence 418
instant messaging 415
setting the server IP address 416
lowwater 99

M

mail servers
configuring GETMAILTARGET properties 403
reading/deleting email 410
MailStore - DB page 412
masks, Cisco 27

Copyright © 1988-2007, Oracle. All rights reserved.

master and slave servers
determining master MCF server failure 100, 187
implementing MCF log server failover 81
implementing queue server failover 79
master servers, *See* master and slave servers
masterinterval 100, 187
max_no_reply 100
max_refresh 100
maxTaskCompletionTime 451
GroupStatistics class 320
MCEvent class
constructor 350, 457
fields 351, 458
See Also MCEvent class fields
MCEvent class fields
extension 351
group 351
reason 351
user 352
MCEvent constructor 351, 457
MCEvent object 457
MCF, *See* MultiChannel Framework (MCF)
configuring broadcast 137
configuring for third party 177
monitoring agent 170
working with sample pages 147
MCF Agent component (MCF_AGENT_CMP) 103
MCF agents, *See* agents
creating for third party 192
defining for third party 191
MCF cluster
tuning parameters 184
using third party routing 182
MCF Cluster component (MCF_UQCLUSTER) 82
MCF Cluster page 83
MCF clusters
configuring 77, 81, 82
defining 83
deleting 84
describing architecture 78
implementing queue server scalability 81
notifying for third party 187
notifying of changed parameters 101
tuning parameters 93
using queue servers within 84
viewing cluster IDs 87
viewing summary information 93
MCF log servers
configuring MCF clusters 82
implementing failover 81
MCF Queue component (MCF_Q_CONFIG_CMP) 85
MCF queues
configuring and defining 85
defining 86
defining chat responses 87
defining for third party 188
defining static push URLs 88
MCF Sample Console 205
MCF Sample Pages component (MCF_DEMO_CMP) 147, 155
MCF servers
architecture 8
configuring 77, 81
determining master failure 100, 187
MCF Supervisor Console 139
MCF task
configuring for third party 200
MCF Task component (MCF_TASKCFG_CMP) 89
MCF Task Configuration page 89
MCF_ACCPT_TASK_CMP component 113
MCF_AD_NOTIFY_CMP component 101
MCF_AGENT_CMP component 103
MCF_AGENTSTATE_CMP component 116
MCF_CHAT_LOG_CMP component 119
MCF_DEMO_CMP component 147, 155
MCF_ESCAL_CMP component 127
MCF_EVENTLOGL_CMP component 119
MCF_IM_CFG_CMP component 416
MCF_IM_DEMO_CMP component 417
MCF_OVERFLOWL_CMP component 127
MCF_Q_CONFIG_CMP component 85
MCF_QSERVSTATE_CMP component 116
MCF_QUEUESTATE_CMP 116
MCF_RSERV_CFG_CMP component 93
MCF_SYSTEM_NV_CMP component 93

Copyright © 1988-2007, Oracle. All rights reserved.

MCF_TASKCFG_CMP component 89
MCF_UQCLUSTER component 82
MCFBroadcast 315
MCFEM_DEMOERMS_CMP component 410
media, agents 198
MediaType class
  constructor 352, 458
  fields 352, 458
MediaType constructor 352, 458
MediaType object 458
Meridian, Nortel 24
message
  ChatConnection class 433
  ChatConnection class methods 262
messages
  defining 195
  defining canned queue 190
messaging, instant, *See* instant messaging
method
  _Address class callback event 226
  alternate 453
  answer 453
  attachUserData 453
  broadcastSubscribe 462
  broadcastUnsubscribe 462
  clear 454
  close 462
  closeSession 459
  complete 454
  conference 454
  conferenceSingle 454
  dial 454
  disableMedia 465
  dropParty 454
  enableMedia 465
  getAni 454
  getCallById 459
  getChatById 459
  getDescr 454
  getDnis 454
  getEmailById 459
  getGenericTaskById 459
  getLineById 459
  getReferenceId 454
  getUrl
    Line class 454
  grabCall 454
  hold 454
  isMediaEnabled 465
  login 465
  loginUq 465
  logout 465
  logoutUq 465
  open 462
  openSession 459
  park 454
  reconnect 454
  register 465
  registerAddress 462
  registerBuddy 462
  registerGroup 462
  registerUser 462
  reject 454
  release 454
  retrieve 454
  sendDTMF 454
  setAutoRecovery 462
  setcallresult 454
  setcallresultDNC 454
  setcallresultReschedule 454
  setNotReady 465
  setPresence 465
  setPresenceUq 465
  setReady 465
  setWorkNotReady 465
  setWorkReady 466
  start 459
  statPublish 462
  stop 459
  transfer 454
  transferMute 454
  unmute 455
  unRegister 466
    UserData class 468
  unregisterAddress 462
  unregisterBuddy 462
  unregisterGroup 462
  unregisterUser 462
  updateCallData 455
methods
  A2AChat class 424
  A2AChat Class 236
  A2AChatAddress class 238, 425
  A2AChatAddress class callback event 239
  acceptTask 422
  addKeyValue
    AppData class 427
    CallData class 429

Copyright © 1988-2007, Oracle. All rights reserved.

ChatData class 435
EmailData class 441
GenericData class 448
answer
ChatConnection class 433
EmailConnection class 440
GenericConnection class 446
AppData class 244, 427
Buddy class callback event 246
CallData class 249, 429
cancelDnd 442
cancelDndforwardSet 442
cancelForwardSet 442
chat
ChatAddress class 432
Chat class 254, 431
ChatAddress class 256, 431
ChatAddress class callback event 258
ChatConnection 260
ChatConnection class 433
ChatData class 271, 435
complete
EmailConnection class 440
GenericConnection class 446
conference
ChatConnection class 433
dequeueTask 422
Email class 275, 438
EmailAddress class 277, 438
EmailConnection class 280, 287, 440
EmailData class 441
Extension class 442
Extension Class 289
See Also Extension class method
Extension class callback event 292
forward
ChatConnection class 433
EmailConnection class 440
GenericConnection class 446
GenericAddress class 298, 445
GenericConnection class 301, 446
GenericData class 308, 448
GenericTask class 311, 449
getChatconnectionByConnectionId
ChatAddress class 432
getChatconnectionindexByConnectionId
ChatAddress class 432
getDialingLine 442
getEmailconnectionByConnectionId
EmailAddress class 439
getEmailconnectionindexByConnectionId
EmailAddress class 439
getFreeChatconnection
ChatAddress class 432
getFreeChatconnectionIndex
ChatAddress class 432
getFreeEmailconnection
EmailAddress class 439
getFreeGenericconnection
GenericAddress class 445
getFreeLine 442
getGenericconnectionByConnectionId
GenericAddress class 445
getGenericconnectionindexByConnectionId
GenericAddress class 445
gethistory
ChatConnection class 433
getLineByConnectionId 442
getOffHookLine 442
gettpUrl
Chat class 431
Email class 438
getUrl
A2AChat class 424
Chat class 431
ChatConnection class 433
Email class 438
GLOBALS class 314, 450
See Also GLOBALS class methods
Group class callback event 318
initiateChat 425
initJSMCAPI 450
isValid 450
Line class 327, 453
See Also Line class methods
Line class callback event 340
message
ChatConnection class 433
PSMC class 354, 459
See Also PSMC class methods
pushURL
ChatConnection class 433
reject
ChatConnection class 433
EmailConnection class 440
GenericConnection class 446
release
ChatConnection class 433
RenServer class callback event 360
Server class callback event 362

Copyright © 1988-2007, Oracle. All rights reserved.

Session class 365, 462
See Also Session class methods
Session class callback event 370
setDnd 442
typing
ChatConnection class 433
_UQAddress 227
_UQAddress class 422
_UQAddress class callback event 228
User class 379, 465
See Also User class methods
User class callback event 386
UserData class 396, 468
withdraw_RES
EmailConnection class 440
GenericConnection class 446
MIME, *See* Multipurpose Internet Mail Extensions (MIME)
Miscellaneous component (PT_CTI_MISC) 19
Miscellaneous page 30, 111
miscellaneous parameters 200
mode 444
mode field 296
Monitor Agents page 170
Monitor Queues page 172
mostRecentTaskData 397, 469
mostRecentTaskDone 322, 452
mostRecentTaskDoneData 322, 452
mostRecentTaskEnqueued 322, 452
mostRecentTaskEnqueuedData 322, 452
mostRecentTaskId 397, 469
Move Agent page 114
Move Queue page 114
MultiChannel Console
channels 6
CTI interface 44
displaying logged agents/unassigned tasks per queue 95
managing tasks/chat 129
understanding 8
using hot keys 131
MultiChannel Framework (MCF)
administering 113
configuring 81
configuring Broadcast 137
configuring failover 77
configuring scalability 77
elements 5
getting started 1
implementation 1
overview 1
server architecture 8
understanding 5
working with sample pages 137
Multipurpose Internet Mail Extensions (MIME) 7, 410
mute 209, 334
Mute 165

N

name
Group class 317, 450
_User class 423
_User class fields 231
NameValue Pairs
implementing JSMCAPI 142
implementing PeopleCode 145
newestTask 451
GroupStatistics class 320
newestTaskCompletionTime 451
GroupStatistics class 320
nodes
configuring for email channels 403
configuring for email languages 408
configuring MCF_GETMAIL properties 403
Nortel Meridian 24
notes xli
notifications
notifying clusters of changed parameters 101
understanding real-time event 55
See Also REN servers
Notify Cluster page 101
notifyinterval 187
NotifyQ
demonstrating generic persistent events 151
understanding 175
numAgentsAvailable 322, 452
numAgentsLoggedIn 322, 452
numberOfAbandoned 320, 451
numberOfLoggedIn 320, 451
numberOfQueued 320, 451
numEscalation 323, 452
numOfLines 289, 442
numOverflow 323, 452
numTaskAccepted 452
GroupStatistics1 class 323

Copyright © 1988-2007, Oracle. All rights reserved.

UserStatistics1 class 397, 469
numTaskDone 323, 452
numTaskQueued 323
numTasksDone 398, 469
numTasksUnassigned 398, 469
numUnassignedTasks 320, 451

O

Object 421
oldestTask 325, 452
onAccepted 422
ChatConnection class callback event methods 264
onAccepted event 229
onAcceptingTask 229, 422
onAddressRegistered 370, 462
onAddressUnregistered 370, 462
onAlternating 340, 455
onAnswering 340, 455
ChatConnection class 433
ChatConnection class callback event methods 264
EmailConnection class 440
EmailConnection class callback event methods 282
onAttachingUD 340, 455
onBroadcast 362, 461
onBuddyRegistered 370, 462
onBuddyUnregistered 370, 462
onCallDataChanged 340, 455
onCancelingDnd 292, 443
onCancelingForward 292, 443
onCapabilitiesChanged 341, 386, 455, 466
ChatAddress class 432
ChatAddress class callback event 258
ChatConnection class 433
ChatConnection class callback event methods 265
EmailConnection class 440
EmailConnection class callback event methods 282
GenericConnection class 446
GenericConnection class callback event methods 304
onChatdataChanged
ChatConnection class 433
ChatConnection class callback event methods 265
onChatEnded 239, 425
onClearing 341, 455
onClosed 371, 462
onCompleted
EmailConnection class 440
EmailConnection class callback event methods 283
GenericConnection class 446, 447
GenericConnection class callback event methods 304
onCompleting 341, 455
onConferencing 341, 455
ChatConnection class 433
ChatConnection class callback event methods 265
onCtiBusy 387, 466
onCTIBUSYUq 387
onCtiClear 387, 466
onDequeueingTask 229, 422
onDialing 341, 455
ChatConnection class 434
ChatConnection class callback event methods 265
onDnd 292, 443
onDndCanceled 293, 443
onDown 360, 460
onDropped 455, 466
ChatConnection class 434
ChatConnection class callback event methods 265
EmailConnection class 440
EmailConnection class callback event methods 283
GenericConnection class 446
GenericConnection class callback event methods 304
Line class 342
User class 387
onEmaildataChanged
EmailConnection class callback event methods 283
onError 455, 462
_Address class 226, 421
ChatConnection class 434
ChatConnection class callback event methods 266
EmailConnection class 440
EmailConnection class callback event methods 283
GenericConnection class 446

Copyright © 1988-2007, Oracle. All rights reserved.

GenericConnection class callback event methods 304
Line class 342
Session class 371
User class 387, 466
onForwardCanceled 293, 443
onForwarded 293, 443
ChatConnection class 434
ChatConnection class callback event methods 266
EmailConnection class 440
EmailConnection class callback event methods 283
GenericConnection class 446
GenericConnection class callback event methods 305
onForwardError
ChatConnection class 434
ChatConnection class callback event methods 266
EmailConnection class 440
EmailConnection class callback event methods 283
GenericConnection class 446
GenericConnection class callback event methods 305
onForwarding 293, 443
ChatConnection class 434
ChatConnection class callback event methods 266
EmailConnection class 440
EmailConnection class callback event methods 284
GenericConnection class 447
GenericConnection class callback event methods 305
onGenericdataChanged
GenericConnection class callback event methods 305
onGrabbing 342, 455
onGroupRegistered 371, 462
onGroupUnregistered 371, 462
onHbLost 362, 461
onHbRecovered 362, 461
onHeld 342, 455
onHistory
ChatConnection class 434
ChatConnection class callback event methods 266
onHolding 342, 455
onIncoming
EmailConnection class 440
EmailConnection class callback event methods 284
GenericConnection class 447
GenericConnection class callback event methods 305
onIncomingChat
ChatConnection class 434
ChatConnection class callback event methods 266
onInfo 462, 466
Session class 371
User class 387
onInitiatingChat 239, 425
onInService 363, 461
onJoining 342, 455
onLoggedIn 388, 466
onLoggedOut 388, 466
onLoggingIn 388, 466
onLoggingOut 388, 466
onMediaDisabled 388, 466
onMediaEnabled 389, 466
onMessage
ChatConnection class 434
ChatConnection class callback event methods 267
onMuted 343, 455
onNotify
A2AChatAddress class 239, 425
_UQAddress class 229, 422
onNotReady 389, 466
onOffHook 343, 455
onOnHook 343, 455
onOpened 372, 463
onOutOfService 363, 461
onParking 343, 455
onPartyAdded 343, 455
ChatConnection class 434
ChatConnection class callback event methods 267
onPartyChanged 344, 455
ChatConnection class 434
ChatConnection class callback event methods 267
onPartyRemoved 344, 455
ChatConnection class 434
ChatConnection class callback event methods 267
onPresenceChanged 389, 466

Copyright © 1988-2007, Oracle. All rights reserved.

onProcessing
EmailConnection class callback event methods 284
GenericConnection class callback event methods 305
onProperties
ChatConnection class 434
ChatConnection class callback event methods 267
onPushURL
ChatConnection class 434
ChatConnection class callback event methods 268
onReady 389, 466
onReconnecting 344, 456
onRegistered 389, 466
onRegistering 389, 466
onRejected 344, 456
ChatConnection class 434
ChatConnection class callback event methods 268
EmailConnection class 440
EmailConnection class callback event methods 284
GenericConnection class 447
GenericConnection class callback event methods 306
onRejecting 344, 456
onReleased
ChatConnection class 434
ChatConnection class callback event methods 268
onReleasing 344, 456
ChatConnection class 434
ChatConnection class callback event methods 268
onRestart 363, 461
onRetrieving 345, 456
onRevoked
ChatConnection class 434
ChatConnection class callback event methods 268
EmailConnection class 440
EmailConnection class callback event methods 284
GenericConnection class 447
GenericConnection class callback event methods 306
onRinging 345, 456
onSetcallresult 345, 456
onSetcallresultDNC 345, 456
onSetcallresultReschedule 345
onSettingDnd 293, 443
onSettingNotReady 390, 466
onSettingPresence 390, 466
onSettingReady 390, 466
onSettingWorkNotReady 390, 466
onSettingWorkReady 390, 466
onStat 450, 463
Group class 318, 450
Task class 374
User class 391, 467
onStat1
Buddy class 246, 428
Group class 318, 450
User class 391
onStat1(
User class 467
onStat2
Buddy class 246, 428
Group class 318, 450
User class 391, 467
onState 246, 428
onTalking 346, 456
ChatConnection class 434
ChatConnection class callback event methods 268
onTaskAdded 450
Group class 318
_UQAddress class 422
__UQAddress class 229
onTaskRemoved
Group class 319
__UQAddress class 229
onTransfering 346, 456
onTyping
ChatConnection class 434
ChatConnection class callback event methods 269
onUnassigned 230
_UQAddress class 422
onUnknown 391, 467
onUnmuted 346, 456
onUnregistered 391, 467
onUnregistering 391, 467
onUp 360, 460
onUpdatingCD 346, 456
onUserdataChanged
ChatConnection class 434

Copyright © 1988-2007, Oracle. All rights reserved.

ChatConnection class callback event methods 269
EmailConnection class 440
EmailConnection class callback event methods 285
GenericConnection class 447
GenericConnection class callback event methods 306
onUserDataChanged 346, 456
onUserRegistered 372, 463
onUserUnregistered 372, 463
onWithdraw_REQ
EmailConnection class callback event methods 285
GenericConnection class callback event methods 306
onWorkNotReady 392, 467
onWorkReady 392, 467
open 366, 462
openSession 356, 459
Outbound Call page 41
outbound call sample page 41
overflow
administering overflow tasks 127
checking for overflowed persistent tasks 101
specifying task overflow default timeout 91
Overflow Administration component (MCF_OVERFLOWL_CMP) 127
Overflow Tasks page 127
overview
MultiChannel Framework 1

P

parameters 200
park 334, 454
PeopleBooks
ordering xxxviii
PeopleCode
sample page 417
using built-in functions 175
PeopleCode broadcast
configuring 144
PeopleCode Broadcast page 144
PeopleCode Sample page 417
PeopleCode, typographical conventions xl
PeopleSoft CTI, *See* CTI
PeopleSoft Integration Broker, *See* Integration Broker
PeopleSoft MultiChannel API (PSMCAPI), *See* PSMCAPI
percentIdleTime
AgentStatistics class 240, 425
percentTimeAvailable
AgentStatistics class 241, 425
percentTimeInCurrentState
AgentStatistics class 241, 426
percentTimeUnavailable
AgentStatistics class 241, 426
performance issues
clustering REN servers 57
recording agent state information 118
restarting MCF servers 81
setting CTI agent tracing 34
tuning MCF clusters 94
permission lists, REN server 59
personalization 35
chat 109
CTI pop-ups 50
dialing phone numbers on pages 41
floating CTI console 35
Personalization page (CTI) 35
Personalize Chat page 109
Personalize Presence page 105
phone books 34
using CTI shared 25
physical queues
active/inactive 87
administering 113
assigning agents 104
balancing agents/tasks across 116
deleting 86
displaying number of unassigned tasks/logged agents 95
forwarding chats 133
implementing scalability 81
moving 114
moving agents 114
selecting MCF clusters 87
understanding 80
See Also queues
viewing MCF cluster IDs 87
viewing physical queue IDs 86
viewing state summary 117
Physical Queues component (MCF_ACCPT_TASK_CMP) 113
ping tests, REN server 71, 83

Copyright © 1988-2007, Oracle. All rights reserved.

pop-ups 36
configuring 38
notifying CTI agent of calls 50
selecting pop-up mode for task/chat acceptance windows 109
setting up Cisco 39
setting up Genesys 38
using signon screens 30
POP3, *See* Post Office Protocol 3 (POP3)
Post Office Protocol 3 (POP3)
calculating email size 410
setting the connector threshold 407
prerequisites xxxvii
presence, agents 197
presences 232, 423
presenceText 398, 469
primary queue servers 84
printed documentation xxxviii
priority 375, 463
PSADMIN
configuring REN servers 64
setting application server tracing 126
PSMC class
constructor 353, 458
fields 353, 458
See Also PSMC class fields
method 354
methods 459
PSMC class fields
renserver 353
servers 354
sessions 354
PSMC class methods
closeSession 354
getCallById 354
getChatById 355
getEmailById 355
getGenericTaskById 356
getLineById 356
openSession 356
start 357
stop 357
PSMC constructor 353, 458
PSMC object 458
PSMCAPI 14
components 16
configuring 16
CTI architecture 13
file locations 16
security 17
PSMCAPI certificates 476
PT_CTI_AGENT component 32
PT_CTI_DEMOOUTB component 41, 155
PT_CTI_MISC component 19
PT_CTI_QUEUE component 32
pushURL
ChatConnection class 433
ChatConnection class methods 263

Q

question
A2AChat class 236
AppData class 243, 426
Chat class 253, 430
Email class 274, 437
GenericTask class 311, 449
queue canned messages 190
queue canned URL 190
Queue Configuration component (PT_CTI_QUEUE) 32
Queue Configuration page 32
queue IDs 189
logical 104
physical 104
understanding character requirements 86
viewing physical queue IDs 86
Queue page 86
queue server
state 101
Queue Server State component (MCF_QSERVSTATE_CMP) 116
Queue Server State page 117
queue servers
assigning tasks to agents 89
See Also tasks
changing for MCF clusters 84
configuring MCF clusters 82
defining failover 79
implementing scalability 81
logging the status 97
master/slave 79
See Also master and slave servers
optimizing performance 95
primary/backup within MCF clusters 84
setting maximum agent timeouts 100
specifying interval for clearing deleted tasks from memory 101

Copyright © 1988-2007, Oracle. All rights reserved.

viewing the server state 117
Queue State Summary component (MCF_QUEUESTATE_CMP) 116
Queue State Summary page 117
queue statistics
monitoring 95
queue statistics, monitoring 172
queue-based broadcast 137
queuedWaitTime 451
GroupStatistics class fields 321
queues
active/inactive 190
configuring 32
defining canned URL 190
defining for third party 188
defining logical queues 80
See Also logical queues
deleting 189
MCF 85
See Also MCF queues
monitoring 172
physical 80
See Also physical queues
task queues, refreshing 100
universal 7
using queue servers 79
See Also queue servers
viewing cluster IDs 189
queueTime 251, 429, 464
TaskStatistics class fields 376
queueUpTime 451
GroupStatistics class fields 321

R

real-time event notification (REN) servers, *See* REN servers
reason 351, 458
Reason class
constructor 357, 459
fields 358, 459
See Also Reason class fields
Reason class fields
code 358
desc 358
reasonData1 358
reasonData2 359
reasonData3 359
Reason Code page 31
Reason constructor 358, 459
Reason object 459
reasonData1 358, 459
reasonData2 359, 459
reasonData3 359, 459
reasonFlag 452
GroupStatistics1 class 323
UserStatistics1 class 398, 469
recentTask 325, 452
reconnect 335, 454
reeperinterval 101
register 382, 465
registerAddress 366, 462
registerBuddy 367, 462
registered 317
Group class 450
registeredAddresses 378, 465
registerGroup 367, 462
registerUser 367, 462
reject 335, 454
ChatConnection class 433
ChatConnection class methods 263
EmailConnection class 440
EmailConnection class methods 281
GenericConnection class 446
GenericConnection class methods 303
related documentation xxxviii
relativeQueueLoad 451
GroupStatistics class fields 321
release 336, 454
ChatConnection class 433
ChatConnection class methods 263
REN Cluster component (REN_CLUSTER_CMP) 68
REN Java client
importing certificates 474
REN Permissions page 59
REN server
configuring in PSADMIN 64
security 17
REN Server Cluster Members page 72
REN Server Cluster Owner page 71
REN Server Cluster page 69
REN Server Cluster Root Path 70
REN Server Cluster URL 70
REN Server component (REN_SERVER_CMP) 60
REN Server Configuration page 66
REN servers
clustering 69
configuring 60, 63, 81
configuring cluster members 72

Copyright © 1988-2007, Oracle. All rights reserved.

configuring clusters 68
configuring failover 55
configuring MCF clusters 82
configuring ownership 71
configuring reverse proxy 73
See Also RPS
configuring scalability 55
configuring security 55, 58
configuring SSL-enabled clusters 68
creating 63
defining 66
defining permission lists 59
determining configuration options 60
logging 98, 185
performing buffer/ping tests 71, 83
understanding 55
understanding clusters 57
understanding SSL-enabled 57
REN_CLUSTER_CMP component 68
REN_SERVER_CMP component 60
renserver 353
renServer 458
RenServer class
callback event methods 460
constructor 359, 460
event method 360
See Also RenServer class callback event methods
fields 359, 460
See Also RenServer class fields
RenServer class callback event methods
onDown 360
onUp 360
RenServer class fields
isRunning 359
url 360
RenServer constructor 359, 460
RenServer object 460
retrieve 336, 454
reverse proxy server, *See* RPS
reverse proxy servers
configuring REN server security 58
configuring REN servers with SSL 62
setting REN server browser URLs 71
routing, third party 177
RPS
configuring 73
configuring OAS 75
configuring WebLogic 8.1 73
understanding configuration 73

S

Sample Console page 204
Sample Monitor - Agent States 170
Sample Monitor - Queue Statistics 172
sample pages
email 410
instant messaging 417
instant messaging button 418
Monitor Agents page 170
PeopleCode 417
Sample Monitor - Agent States 170
using customer chat 148
using Email page 153
using Email window 154
using Generic Event page 151
using Generic Event window 152
using Monitor Queues page 172
using Sample Monitor - Queue Statistics 172
using URL wizard 149
working with 147
sample pages, third party 203
scalability
configuring MCF 81
configuring REN server clusters 68
configuring REN servers 55
understanding 77
using queue servers 81
Secure Socket Layer (SSL)
configuring REN server clusters 57
configuring REN server security 58
configuring REN/reverse proxy servers 62
using the MultiChannel Console 130
security
accessing CTI 45
applying CTI applet protection 19
configuring REN servers 55, 58
opening MultiChannel consoles 130
PSMCAPI 17
REN server 17
setting security zones for web/REN servers 66, 130
setting the authentication domain for REN servers 63
specifying internal/external URLs for MCF clusters 82
sendDTMF 336, 454
server authentication 58

Copyright © 1988-2007, Oracle. All rights reserved.

Server class
  callback event methods 461
  constructor 360, 460
  event method 362
    See Also Server class callback event methods
  fields 361, 460
    See Also Server class fields
Server class callback event methods
  onBroadcast 362
  onHbLost 362
  onHbRecovered 362
  onInService 363
  onOutOfService 363
  onRestart 363
Server class fields
  id 361
  info 361
  state 361
  type 361
Server constructor 360, 460
Server object 460
Server.TYPE.CTI 449
Server.TYPE_CTI 313
Server.TYPE_UQ 313, 449
serverId 364
serverID 461
servers 354, 458
  application 8
    See Also application servers
  configuring reverse proxy 58
    See Also RPS
  instant messaging 415
    See Also instant messaging
  mail 403
    See Also mail servers
  master/slave 79
    See Also master and slave servers
  MCF 8
    See Also MCF servers
  queue 79
    See Also queue servers
  REN 8
  understanding REN servers 55
    See Also REN servers
  understanding web servers 55
    See Also REN servers
  using MCF log 81
    See Also MCF log servers
service request masks 29
Session class
  callback event methods 462
  constructor 363, 461
  event method 370
    See Also Session class callback event methods
  fields 364, 461
    See Also Session class fields
  method 365
  methods 462
Session class callback event methods
  onAddressRegistered 370
  onAddressUnregistered 370
  onBuddyRegistered 370
  onBuddyUnregistered 370
  onClosed 371
  onError 371
  onGroupRegistered 371
  onGroupUnregistered 371
  onInfo 371
  onOpened 372
  onUserRegistered 372
  onUserUnregistered 372
Session class fields
  addresses 364
  buddies 364
  groups 364
  id 364
  serverId 364
  state 364
  user 365
Session class methods
  broadcastSubscribe 365
  broadcastUnsubscribe 365
  close 366
  open 366
  registerAddress 366
  registerBuddy 367
  registerGroup 367
  registerUser 367
  setAutoRecovery 368
  statPublish 368
  unregisterAddress 368
  unregisterBuddy 369
  unregisterGroup 369
  unregisterUser 369
Session constructor 363, 461
Session object 461
sessions 354, 458
setAutoRecovery 368, 462

Copyright © 1988-2007, Oracle. All rights reserved.

setcallresult 337, 454
setcallresultDNC 337, 454
setcallresultReschedule 338, 454
setDnd 292, 442
setNotReady 383, 465
setPresence 383, 465
setPresenceUq 384, 465
setReady 384, 465
setWorkNotReady 465
setWorkReady 385, 466
Shared Phone Book page 25
Siemens Hicom 300 24
signon, single, *See* single signon
Simple Mail Transfer Protocol (SMTP) 408
single signon
  accessing REN servers 59
  setting the default URL 30
  supporting on CTI 40
slave servers, *See* master and slave servers
SMTP 408
socket binding 66
SSL, *See* Secure Socket Layer (SSL)
SSL only 67
SSL Port 67
ST_LOGGEDIN 232, 423
ST_LOGGEDOUT 232, 423
ST_NOTREADY 232, 423
ST_READY 232, 423
ST_UNKNOWN 232, 423
ST_WORKNOTREADY 232, 423
ST_WORKREADY 233, 423
start 357, 459
statdump 101
state
  ChatConnection class 433
  ChatConnection class fields 259
  EmailConnection class 439
  EmailConnection class fields 279
  GenericConnection class 446
  GenericConnection class fields 301
  Line class 327, 453
  Server class 361, 460
  Session class 364, 461
  UserStatistics1 class 398, 469
states 232, 423
statesUq 379, 465
static push URLs
  defining 88
  defining agent-specific 110
statistics
  Call class 248, 428
  Chat class 254, 430
  Email class 274, 437
  GenericTask class 311
  Group class 317, 450
  _User class 233, 423
statistics, logging 98, 186
statistics1
  Group class 317, 450
  _User class 233, 423
statistics2
  Group class 318, 450
  _User class 233, 423
statPublish 368, 462
status
  detecting instant messaging presence 418, 419
  dialing outbound calls 52
  logging for agents 96
  logging for queue servers 97
  switching Agent Ready status 51
  understanding hold status 50
  viewing for CTI agents 46
stop 357, 459
strData 243, 426
subject
  A2AChat class 236
  AppData class 243, 426
  Chat class 253, 431
  Email class 274, 437
  GenericTask class 311, 449
suggestions, submitting xlii
system-wide broadcast 137

T

talkTime 251, 429, 464
  TaskStatistics class fields 377
task
  configuring for third party 200
Task class
  constructor 373, 463
  fields 374, 463
    See Also Task class fields
  Hierarchy 372
Task class fields
  caseid 374
  cost 374
  customerid 374
  group 374

Copyright © 1988-2007, Oracle. All rights reserved.

id 374
onStat 374
priority 375
type 375
urlAbs 375
urlRel 375
Task constructor 373, 463
Task object 463
task queues, refreshing 100
Task.TYPE_A2ACHAT 313, 449
Task.TYPE_CHAT 314, 449
Task.TYPE_CTI 314, 449
Task.TYPE_EMAIL 314, 449
Task.TYPE_GENERIC 314, 449
taskAcceptedCurrentLogin
AgentStatistics class 426
tasks
A2AChatAddress class 238, 425
activating 130
active/inactive physical queues 87, 190
administering escalation/overflow 127
ChatAddress class 431
checking for expired/overflowed persistent tasks 101
configuring 89
delivering/assigning 8
displaying number of unassigned tasks per queue 95
managing via MultiChannel Console 129
managing with MultiChannel Console 8
moving agents 114
moving queues associated with agents 114
refreshing the task queue 100
selecting acceptance mode in chat 109
selecting task assignment actions 111
setting agent skill level 104
specifying chat greeting interval 92
specifying default acceptance timeout 91
specifying default cost 90
specifying default escalation/overflow timeout 91
specifying default priority/agent skill level 90
specifying how many/often to read persistent tasks 97, 99
specifying interval for clearing deleted tasks from memory 101
specifying range for parameter values 92
tuning clusters 94
universal queue 7
_UQAddress class 422
using PeopleCode built-in functions 175
viewing the enqueued task count 92
Tasks field 227
TaskStatistics class
constructor 376, 464
fields 376, 464
See Also TaskStatistics class fields
TaskStatistics class fields
data 376
holdTime 376
queueTime 376
talkTime 377
TaskStatistics constructor 464
TaskStatistics object 464
taskTotalTimeInSystem 323, 452
TCP_NODELAY 66
teleset 49
terms 479
third party
configuring MCF 180
configuring task 200
creating agents 192
defining agents 191
defining MCF cluster page 182
defining outing rules 180
defining queues 188
defining system requirements 180
defining the flag 181
notifying clusters 187
setting up buddy list 193
specifying agent's media 198
specifying languages that agent supports 199
tuning MCF cluster parameters 184
understanding routing 177
using Reason Code page 31
viewing broadcast logs 203
viewing event logs 201
working with sample pages 203
third-party flag 181
time zone offsets 402
timeCurrentLogin

Copyright © 1988-2007, Oracle. All rights reserved.

AgentStatistics class 241, 426
timeElapsedOldestTask 325, 451, 453
GroupStatistics class fields 321
timeElapsedRecentTask 325, 453
timeIdle 399, 470
timeInCurrentState 398, 399, 469, 470
timeNotReady 400, 470
timeouts
setting maximum agent timeouts for queue server logout 100
specifying for chats 92
specifying the default for task acceptance 91
task escalation/overflow 91, 127
timeSinceLoggedIn 398, 469
timeSinceLogin 400, 470
timeSinceStart 324, 452
timeWorking
AgentStatistics class 241, 426
timinginterval 101
totalTaskAcceptedLogin
AgentStatistics class 241, 426
totalTaskDoneLogin
AgentStatistics class 241, 426
totalTaskUnassignedLogin
AgentStatistics class 242
totalTimeAvailable 400, 470
totalTimeUnavailable 400, 470
tracing
setting for agents 112
setting for CTI agents 34
using application servers 126
transfer 338, 454
transfer conference setup masks 29
transferMute 338, 454
transferring callers 48
troubleshooting CTI errors 34
tuning clusters 184
type
A2AChat class 236, 424
MediaType class 352, 458
Server class 361, 460
Task class 375, 463
typing
ChatConnection class 433
ChatConnection class methods 264
typographical conventions xl

U

unavailableDuration 242, 426
uniqueId 243, 426
uniqueueId 236, 424
universal queue 7
universal queue classes 175
unmute 339, 455
unregister 386
unRegister 466
UserData class 468
unregisterAddress 368, 462
unregisterBuddy 369, 462
unregisterGroup 369, 462
unregisterUser 369, 462
updateCallData 339, 455
UQ Address constructor 227
UQ Cluster (universal queue cluster) page 82
_UQAddress class
callback event methods 422
constructor 227, 422
event method 228
See Also _UQAddress class callback event methods
fields 227, 422
methods 227, 422
See Also _UQAddress methods
_UQAddress class callback event methods
onAccepted 229
onAcceptingTask 229
onDequeueingTask 229
onNotify 229
onTaskAdded 229
onTaskRemoved 229
onUnassigned 230
_UQAddress class constructor 227
_UQAddress class fields 227
_UQAddress methods
acceptTask 228
dequeueTask 228
_UQAddress object 422
_UQAddress() constructor 422
url 426, 460
AppData class 244
RenServer class 360
URL Wizard page 149
urlAbs 375, 463
urlRel 375, 463
URLs
defining agent-specific 110
defining REN server cluster 57
defining static push 88

Copyright © 1988-2007, Oracle. All rights reserved.

grabbing 149
sending/grabbing via chat 133
specifying internal/external URLs for MCF clusters 82
using the URL wizard 149
viewing details for MCF clusters 93
user 458
MCEvent class 352
Session class 365, 461
User class
callback event methods 466
constructor 377, 464
event method 386
See Also User class callback event methods
fields 377, 464
See Also User class fields
method 379
methods 465
_User class 231
constructor 230, 422
fields 231, 423
User class callback event methods
onCapabilitiesChanged 386
onCtiBusy 387
onCTIBUSYUq 387
onCtiClear 387
onDropped 387
onError 387
onInfo 387
onLoggedIn 388
onLoggedOut 388
onLoggingIn 388
onLoggingOut 388
onMediaDisabled 388
onMediaEnabled 389
onNotReady 389
onPresenceChanged 389
onReady 389
onRegistered 389
onRegistering 389
onSettingNotReady 390
onSettingPresence 390
onSettingReady 390
onSettingWorkNotReady 390
onSettingWorkReady 390
onStat 391
onStat1 391
onStat2 391
onUnknown 391
onUnregistered 391
onUnregistering 391
onWorkNotReady 392
onWorkReady 392
_User class field
ST_LOGGEDIN 232
states 232
User class fields
addresses 378
agentPassword 378
language 378
registeredAddresses 378
statesUq 379
_User class fields
agentID 231
caps 231
id 231
name 231
presences 232
ST_LOGGEDOUT 232
ST_NOTREADY 232
ST_READY 232
ST_UNKNOWN 232
ST_WORKNOTREADY 232
ST_WORKREADY 233
statistics 233
statistics1 233
statistics2 233
User class methods
ctiBusyUq 379
disableMedia 379
enableMedia 380
enableMediaregister 382
enableMediasetNotReady 383
isMediaEnabled 380
login 381
loginUq 381
logout 382
logoutUq 382
setPresence 383
setPresenceUq 384
setReady 384
setWorkNotReady 385
setWorkReady 385
unregister 386
User constructor 377, 464
_User constructor 230
User object 464
_User object 422
_User() constructor 422

Copyright © 1988-2007, Oracle. All rights reserved.

UserCaps class
  constructor 392, 393, 467
  fields 467
UserCaps class fields
  canHandleChat 393
  canHandleEmail 393
  canHandleGeneric 393
  canHandleVoice 393
  canLogin 393
  canLogout 393
  canRefreshState 394
  canSetNotReady 394
  canSetPresence 394
  canSetReady 394
  canSetWorkNotReady 394
  canSetWorkReady 394
UserCaps constructor 392, 467
UserCaps object 467
userdata
  GenericTask class 311
userData 428
  Chat class 254, 431
  Email class 274, 437
  GenericTask class 449
UserData class
  constructor 394, 468
  fields 395, 397
  methods 396, 468
    See Also UserData class methods
UserData class field constant 395
UserData class methods
  addKeyValue 396
UserData constructor 395, 468
UserData object 468
userId 244, 427
username 244, 427
UserStatistics1 class
  constructor 396, 469
  fields 469
UserStatistics1 class fields
  availableCost 397
  ctiBusy 397
  currentQueue 397
  mostRecentTaskData 397
  mostRecentTaskId 397
  numTaskAccepted 397
  numTasksDone 398
  numTasksUnassigned 398
  presenceText 398
  reasonFlag 398
  state 398
  timeInCurrentState 398
  timeSinceLoggedIn 398
UserStatistics1 constructor 396, 469
UserStatistics1 object 469
UserStatistics2 class
  constructor 399, 469
  fields 399, 470
UserStatistics2 class fields
  currentQueue 399
  timeIdle 399
  timeInCurrentState 399
  timeNotReady 400
  timeSinceLogin 400
  totalTimeAvailable 400
  totalTimeUnavailable 400
UserStatistics2 constructor 399, 469
UserStatistics2 object 469

V

visual cues xli
voice channels 6
  See Also channels

W

waitDuration 242, 426
warnings xli
web servers*, See* REN servers
Window Config (configuration) page 108
windows
  configuring for agents 108
  configuring pop-ups 38
  personalizing pop-ups 36
  setting up Cisco for pop-ups 39
  setting up Genesys for pop-ups 38
withdraw_RES
  EmailConnection class 440
  EmailConnection class methods 282
  GenericConnection class 446
  GenericConnection class methods 303
wizUrl 244, 427
workload
  implementing queue server scalability 81
  setting maximum for agents 105
  specifying for agents 90
    See Also tasks
  universal queue 7
wrap-up mode 51

Copyright © 1988-2007, Oracle. All rights reserved.

wrap-up time, Genesys 52

Y

Yahoo! Messenger
configuring servers 415
detecting presence 415
setting the server IP address 416
updating presence 418

Copyright © 1988-2007, Oracle. All rights reserved.

Copyright © 1988-2007, Oracle. All rights reserved.