

Contents

- [1 Entry Point Creation](#)
- [2 Chat Entry](#)
- [3 Standalone Routing Engine](#)
- [4 Integrated Routing Engine](#)
 - ◆ [4.1 Watchdog Timer](#)
- [5 IPTA and nIPTA for Chats](#)

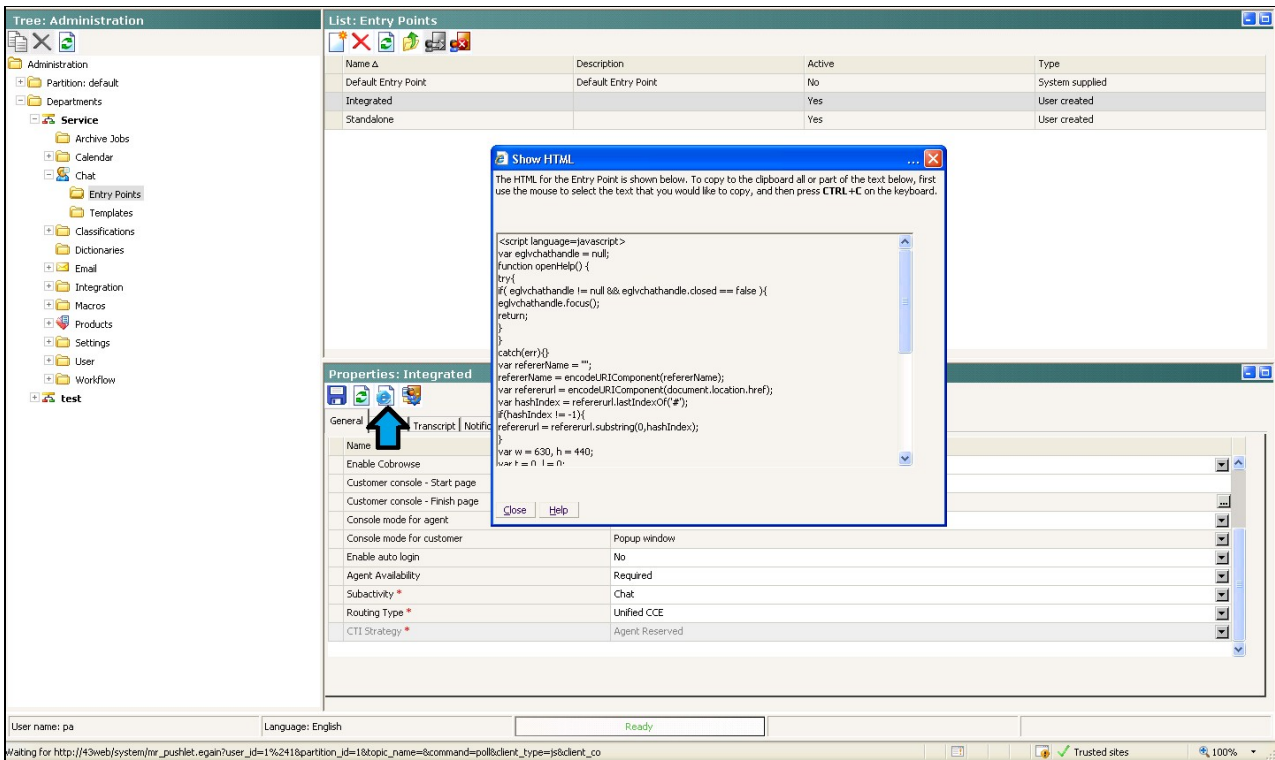
Entry Point Creation

- Entry Points are configured from the Administration Console > Departments > Service > Chat > Entry Points.
- Each Entry Point is associated with a Queue and Routing Type

Name	Value
Queue *	CIM_WIM
Template Set *	Classic Gray
Enable Cobrowse	No
Customer console - Start page	
Customer console - Finish page	Last page browsed
Console mode for agent	Popup window
Console mode for customer	Popup window
Enable auto login	No
Agent Availability	Required
Subactivity *	Chat
Routing Type *	Unified CCE

The javascript for this Entry Point can be found by clicking the "Show HTML" button.

Chat_Routing_Basics



A very basic way of running this javascript is to create a .html file with the below contents. This will display the Entry Point screen seen by customers.

```
<html>
<head>
<title> Chat Entry Point </title>
</head>
<body bgcolor="white" text="blue">

*****
** INSERT JAVASCRIPT FROM "SHOW HTML" HERE **
*****

</body>
</html>
```

Chat Entry

Customers initiate chats through a Chat Entry Point that an administrator makes available on the company website.

Customer Entry - Mozilla Firefox

http://43web/system/LiveCustomerServlet.egain?eglvcmd=CustEntry&entryPo

Live help | Powered by Cisco Chat™

To help us assist you, please enter the information below and click 'Start Chat'.

(* denotes required field)

Full Name: *

Email Address: *

Phone Number:

How can we help you? *

Delay Time(in min)

The WIM Application Server then inserts the activity into the eActiveDB through a JMS Session Pool worker connection.

Standalone Routing Engine

The Agent Assignment Service handles all standalone chat routing. These are the primary logs to look at when tracing chats.

When a chat is initiated, AAS finds the longest available agent:

```
2011-06-01 16:10:07.097 GMT-0400 <@> INFO <@> [47:1] <@> ProcessId:3380 <@> PID:1 <@> UID:12 <@> H
```

Pushes the chat to and notifies the agent:

```
2011-06-01 16:10:07.160 GMT-0400 <@> INFO <@> [47:1] <@> ProcessId:3380 <@> PID:1 <@> UID:12 <@> H
```

And completes the assignment:

```
2011-06-01 16:10:07.160 GMT-0400 <@> INFO <@> [47:1] <@> ProcessId:3380 <@> PID:1 <@> UID:12 <@> H
```

Integrated Routing Engine

Integrated Chat routing is more complicated than Standalone, but uses the same EAAS-MR PIM and CTI Server-Listener connections that are used with Email.

- The NEW_TASK is sent from EAAS to MR PIM. The activity is now queued to UCCE, waiting for the routing script to determine an available agent.

```
15:40:10:341 pg2A-pim2 Trace: [ 7392]Application->PG:
Message = NEW_TASK; Length = 71 bytes
  DialogueID = (24) Hex 00000018
  SendSeqNo = (1) Hex 00000001
  MRDomainID = (5002) Hex 0000138a
  PreviousTask = -1:-1:-1
  PreferredAgent = Undefined
  Service = (0) Hex 00000000
  CiscoReserved = (0) Hex 00000000
  ScriptSelector: 8100
ECC Variable Name: user.cim.activity.id
Value: 1254
```

- A pre-call notification is sent to CTI server signifying that an agent belonging to this Agent PG has been picked for assignment.

```
15:40:10:356 cg1A-ctisvr Trace: DEVICE_TARGET_PRE_CALL_IND - Instrument= RouterCallKey=149900 5005
15:40:10:356 cg1A-ctisvr Trace: SvSkTargID=-1(-1) SkGroupSkTarID=5039(541)
15:40:10:356 cg1A-ctisvr Trace: AgSkTargID=5005 ANI= CED=
15:40:10:356 cg1A-ctisvr Trace: Var1= Var2= Var3= Var4= Var5=
15:40:10:356 cg1A-ctisvr Trace: Var6= Var7= Var8= Var9= Var10=
15:40:10:356 cg1A-ctisvr SESSION 1: MsgType:AGENT_PRE_CALL_EVENT (MonitorID:0 NumNamedVariables:1
15:40:10:356 cg1A-ctisvr SESSION 1:           ServiceID:N/A SkillGroupNumber:541 SkillGroupID:5039 S
15:40:10:356 cg1A-ctisvr SESSION 1:           AgentInstrument:"" RouterCallKeyDay:149900 RouterCallK
15:40:10:356 cg1A-ctisvr SESSION 1:           RouterCallKeySequenceNum:1 DialedNumber:"8100" )
```

- MR PIM sends the DO_THIS_WITH_TASK back to EAAS with routing instructions.

```
15:40:10:356 pg2A-pim2 Trace: [ 5192]PG->Application:
Message = DO_THIS_WITH_TASK; Length = 90 bytes
  DialogueID = (24) Hex 00000018
  SendSeqNo = (1) Hex 00000001
  IcmTaskID = 149900:5005201: 1
  SkillGroup = (5039) Hex 000013af
  Service = Undefined
  Agent = (5005) Hex 0000138d
  AgentInfo: 1002
  Label:
  ApplicationString2:
  Call Variable 1:
  Call Variable 2:
  Call Variable 3:
  Call Variable 4:
  Call Variable 5:
  Call Variable 6:
  Call Variable 7:
  Call Variable 8:
  Call Variable 9:
  Call Variable 10:
ECC Variable Name: user.cim.activity.id
```

Chat_Routing_Basics

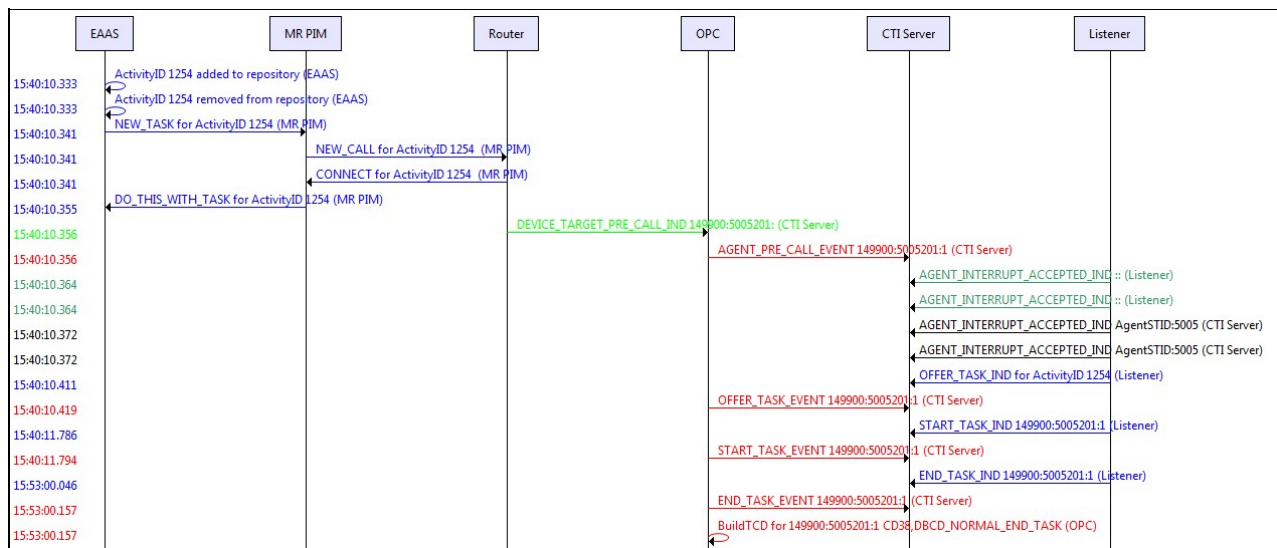
Value: 1254

- EAAS alerts the Listener

```
2011-06-01 15:40:10.464 GMT-0400 <@> INFO <@> [228539:RMI TCP Connection(11883)-10.77.30.32] <@> P
instance_id = 2000 : Type = java.lang.Long
customer_home = 43app : Type = java.lang.String
activity_sub_type = 2001 : Type = java.lang.Integer
true = true : Type = java.lang.String
do_this_with_task_type = 0 : Type = java.lang.Integer
mrd_id = 5002 : Type = java.lang.Long
cim_user_id = 1002 : Type = java.lang.Long
do_this_with_task = com.cisco.ipcc.mr.message.DoThisWithTaskMsg@138a184 : Type = com.cisco.ipcc.mr
activity_id = 1254 : Type = java.lang.Long
expired_status = false : Type = java.lang.Boolean
listenerInstanceId= 2000 <@>
```

- The Listener offers the task to the Agent, who starts the activity shortly thereafter.

```
2011-06-01 15:40:10.511 GMT-0400 <@> INFO <@> [4621:pool-3-thread-2] <@> ProcessId:5916 <@> PID:1
2011-06-01 15:40:11.886 GMT-0400 <@> INFO <@> [228523:RMI TCP Connection(11881)-10.77.30.31] <@> P
```



Watchdog Timer

In an integrated environment, chats cannot stay queued forever. If the chat does not get past step #1 and waits in queue for a configurable duration, the WIM Watchdog Timer will be triggered, ending the chat session immediately. The customer must start a new session and wait in queue again.

By default, the Watchdog Timer is set to 70 seconds. This can be modified by changing a properties file on the File Server.

- File Name: C:\CIM\Service\config\ipcc\legicm_configuration.properties
- Config Option: CHAT_WATCHDOG_INSEC = 70

Note that in 4.3(1), these chats were treated as "Serviced" by the System User. A patch was delivered to treat these as "Abandoned" chats.

IPTA and nIPTA for Chats

Like email, Chat also follows the same logic of IPTA and nIPTA for activity routing. In most cases, however, customers will not use nIPTA chat queues.

With email, nIPTA queues can be checked periodically, just like the Exception queue, and do not have to consistently have agents logged in and actively handling activities. But chat is a real-time activity, which means customers would need to always have a group of separate nIPTA agents available for handling nIPTA chats that have failed IPTA routing. This is not efficient.