Troubleshoot Agent Login and Availability in Enterprise Chat and Email

Contents

Introduction

This document describes how Enterprise Chat and Email(ECE) identifies the agent availability status when clients initiate chat sessions.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Enterprise Chat and Email
- Web browsers Developer Tools
- Unified Intelligent Contact Management Enterprise

Components Used

The information in this document is based on the software version ECE 11.6.

The information in this document were created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration.

Functional Overview

In order for Cisco Unified Intelligent Contact Management Enterprise(ICM) to manage the agent activities and properly route tasks, ICM must monitor all the agents that are logged into ICM. The application instances, like ECE, report the agent activities and agent status through the extended ICM CTI/ARM (Agent Report and Management) interface.

The ARM service is built upon the current CTI Server functionality and allows a client application to monitor application agents and task activity. The ARM Interface allows a client application to monitor a specified set of agents (workstation mode) or all agents (bridge mode) associated with an application.
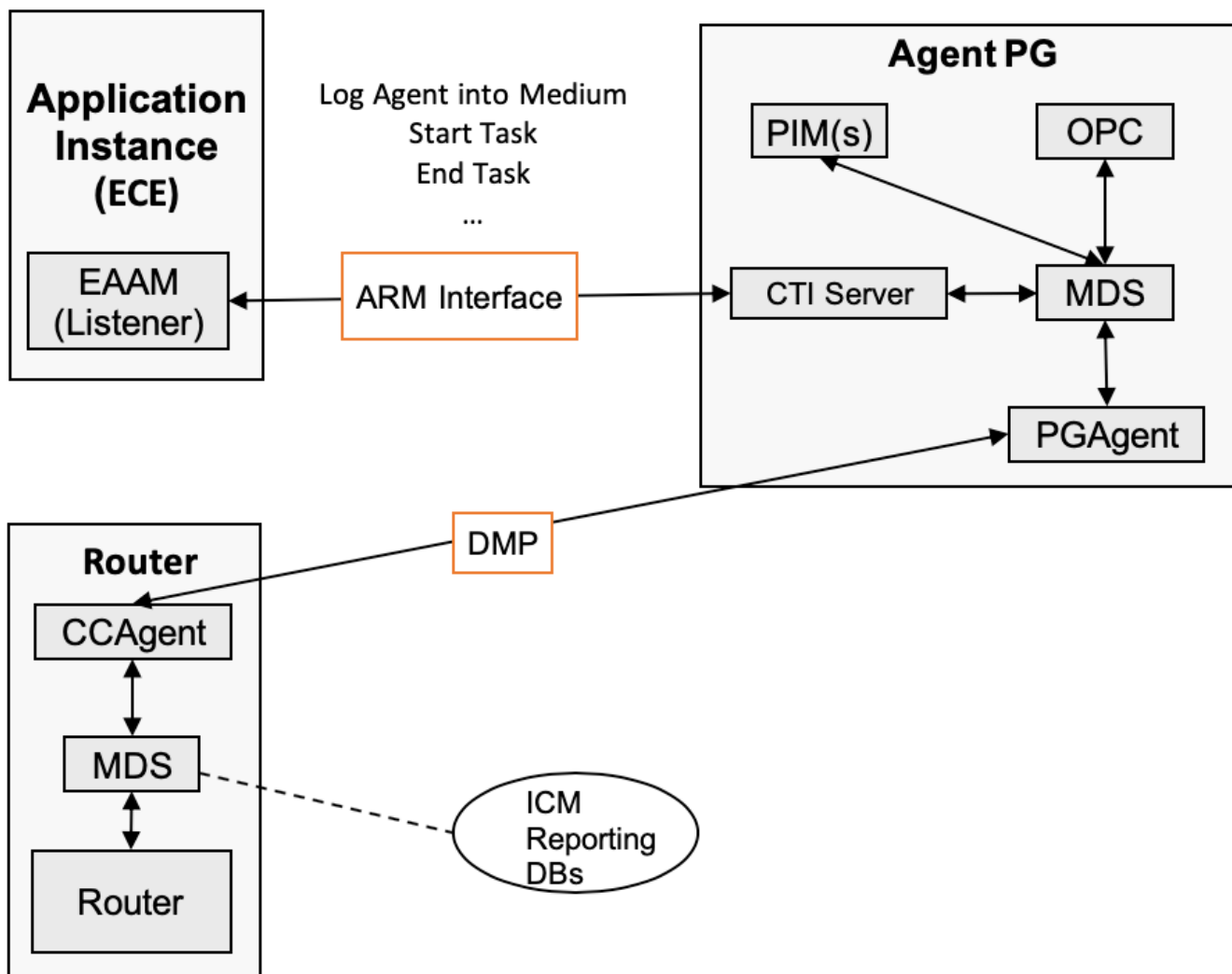
The image shows more details of the ARM interfaces. An application instance uses the ARM interface to manage agents on one or more Agent PGs (log them in and out of media, etc) and to report on their task activity (Start Task, End Task, etc).

Agent Log in Flow

Agent availability is identified from CTI Server side. When an agent logs in to agent console, ECE Listener process sends the request to CTI Server. The request indicates that the agent is logged in and marked himself as available.

These are the indicators which are sent by ECE application to the CTI Server:

CTI State Management

Step	Action
Agent logs into eGain	• Listener sends MEDIA_LOGIN_REQ • Listener sends START_APPLICATION_TASK_REQ for all open activities
Listener pushes activity to agent inbox	• Listener sends OFFER_TASK_IND
Agent clicks on new activity	• Listener sends START_TASK_IND
Agent switches from activity A1 to A2	• For A1, Listener sends PAUSE_TASK_IND • For A2, Listener sends RESUME_TASK_IND or START_TASK_IND (if new)
Agent transfers activity to queue	• Listener sends END_TASK_IND
Agent completes activity in inbox (agent has wrap-up role)	• Listener sends WRAPUP_TASK_IND on complete • Listener sends END_TASK_IND on end wrap-up
Agent completes activity in inbox	• Listener sends END_TASK_IND
Agent logs out of eGain	• Listener sends MEDIA_LOGOUT_IND

Whenever an agent is logged in, a listener sends a MEDIA_LOGIN_REQ. The MEDIA_LOGIN_REQ logs the specified agent into an Media Route Domain(MRD) (logs agent into all skills configured for that MRD and agent). When an agent marks himself as available, the listener does send two more requests that indicate that the agent is ROUTABLE or NOT ROUTABLE and READY or NOT READY, and provides client-defined agent information. The CTI client must have specified the Application Path for the related MRD Peripheral pair in the Open Request message or the log in is rejected. For the log in to succeed, the agent must also be configured to belong to at least one Skill Group(SG) that belongs to the indicated MRD.

The image shows the messages flow diagram for log in request:
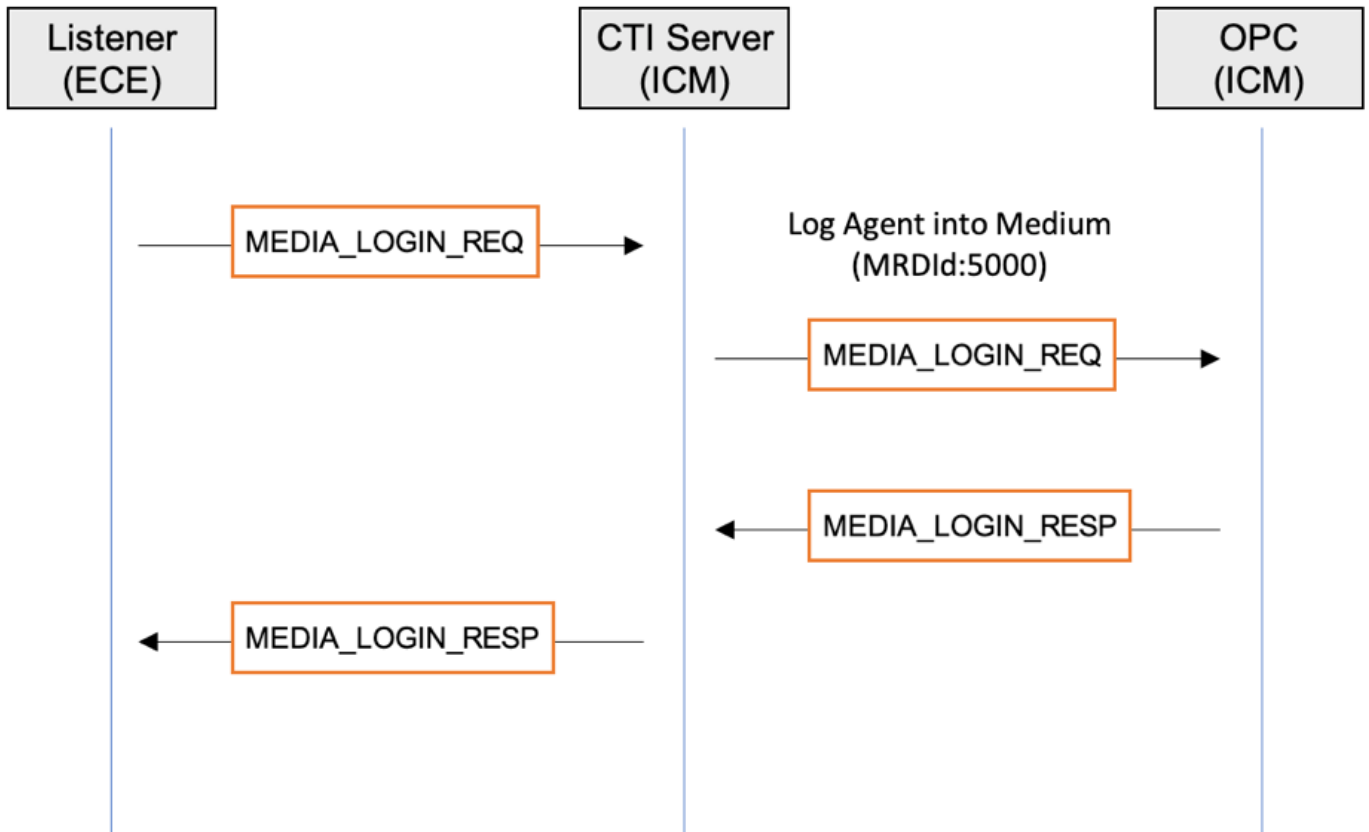
Listener log with the INFO trace level:

```
2019-07-20 18:27:31.749 GMT+0000 <@> INFO <@> [14285:listener-event-pool-priority-arm-request-
executor::-0] <@> ProcessId:4584
<@> PID:1 <@> UID:1005 <@> HttpSessionId:IrltMMd3T0prrkbhAwK8wkL5 <@>
com.ipcc.listener.arm.ARMLogger <@>
<@> Sending MEDIA_LOGIN_REQ  -> 0 0 0 27 0 0 0 -105 0 2 8 1 0 0 19 -120 0 0 19 -87 0 0 0 0 0 0 0
1 107 5 49 48 48 53 0     <@>
```

```
2019-07-20 18:27:32.037 GMT+0000 <@> INFO <@> [71:Thread-9] <@> ProcessId:4584 <@> PID:1 <@>
UID:12 <@> HttpSessionId:
<@> com.ipcc.listener.arm.ARMLogger <@>  <@> Received MEDIA_LOGIN_RESP  ->  0 0 0 8 0 0 0 -104 0
2 8 1 0 0 0 0    <@>
```

 CTIsvr log with the default trace level:

```
20:27:32:466 cg1A-ctisvr Trace: ProcessMediaLoginReq - sessionID 4
20:27:32:466 cg1A-ctisvr Trace: SendARMMsg -- InvokeID = 591309094, MRDID = 5000, ICMAgentID =
5033, AgentMode = 0
IsAvailable = 0, MaxTaskLimit = 1, AgentInfo = 1005, ApplicationPathID = 5001, PeripheralID = 0,
AgentID =
20:27:32:607 cg1A-ctisvr Trace: ProcessARMMediaLoginRespMsg -- InvokeID = 591309094, Status = 0,
AgentSkillTargetID = 5033
```

Status 0 means No errors were occurred from CTI Server side.

Agent Availability Flow

If the agent is associated to the chat SG, and this SG is associated to the ECE queue in Chat Entry point, when agent marks himself available you do see 2 requests, the MAKE_AGENT_ROUTABLE_IND and MAKE_AGENT_READY_IND**.**

Make Agent Routable Indication tells to ICM that the specified agent has been set to a ROUTABLE mode for the specified MRD.

> **Note**: The Make Agent Routable Indication message can be sent while it waits for a Make Agent Not Routable Response and cancels the pending Make Agent Not Routable Request.

Once Make Agent Ready Indication request is received by the listener from application server, the listener forwards the request to the CTI Server and at that moment the agent considered as available for ECE. In that case, if chat is initiated at the same time, the system allows to start and creates the chat activity for that chat.
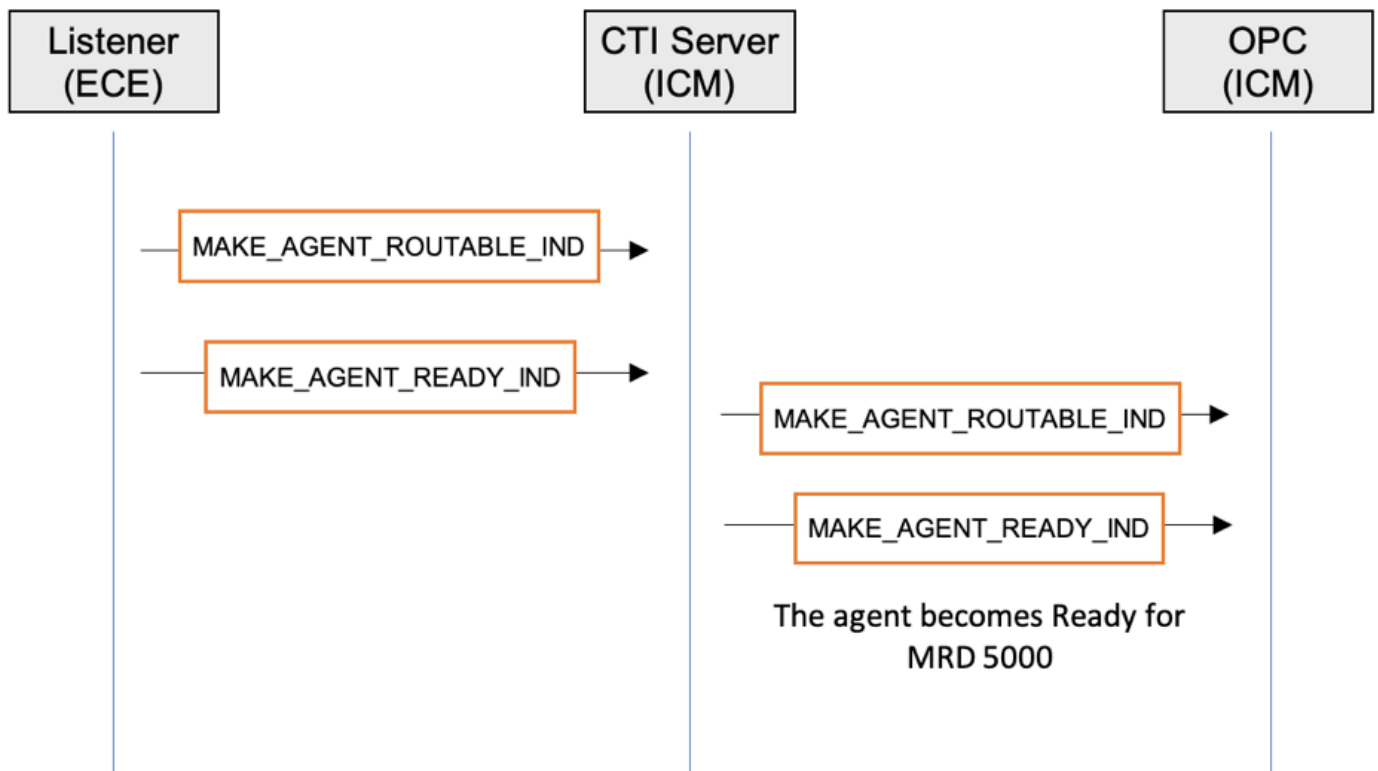
The Listener log shows those requests if the INFO trace is enabled:

```
2019-08-19 13:34:09.773 GMT+0000 <@> INFO <@> [8938:listener-event-pool-priority-arm-request-
executor::-441] <@> ProcessId:5436 <@> PID:1 <@> UID:12 <@> HttpSessionId: <@>
com.ipcc.listener.AgentAvailabilityStatusHandler <@> <@>
AgentAvailabilityStatusHandler:agentIsAvailable() MAKE_AGENT_ROUTABLE_IND to ARM
armLoginDataArraySize= ARMAgentData
================================================================
2019-08-19 13:34:09.773 GMT+0000 <@> INFO <@> [8938:listener-event-pool-priority-arm-request-
executor::-441] <@> ProcessId:5436 <@> PID:1 <@> UID:12 <@> HttpSessionId: <@>
com.ipcc.listener.arm.ARMLogger <@> <@> Sending MAKE_AGENT_ROUTABLE_IND -> 0 0 0 16 0 0 0 -102 0
1 57 43 0 0 19 -120 0 0 25 20 0 0 0 2 <@>
2019-08-19 13:34:09.774 GMT+0000 <@> INFO <@> [8938:listener-event-pool-priority-arm-request-
executor::-441] <@> ProcessId:5436 <@> PID:1 <@> UID:12 <@> HttpSessionId: <@>
com.ipcc.listener.arm.ARMLogger <@> <@> Sending MAKE_AGENT_READY_IND -> 0 0 0 14 0 0 0 -99 0 1
```

```
57 44 0 0 19 -120 0 0 25 20 0 1 <@>
2019-08-19 13:34:09.774 GMT+0000 <@> INFO <@> [8938:listener-event-pool-priority-arm-request-
executor::-441] <@> ProcessId:5436 <@> PID:1 <@> UID:12 <@> HttpSessionId: <@>
com.ipcc.listener.AgentAvailabilityStatusHandler <@> <@> PRINT_STATE after sending
MAKE_AGENT_READY_IND to ARM:
```

The output from CTI Server and OPC processes logs:

```
### CTI Server
15:34:09:841 cg1A-ctisvr Trace: ProcessMakeAgentRoutableInd - sessionID 6
15:34:09:841 cg1A-ctisvr Trace: SendARMMsg -- InvokeID = 80171, MRDID = 5000, ICMAgentID = 6420,
MaxTasks = 2, SessionID = 6
15:34:09:841 cg1A-ctisvr Trace: ProcessMakeAgentReadyInd - sessionID 6
15:34:09:841 cg1A-ctisvr Trace: SendARMMsg -- InvokeID = 80172, MRDID = 5000, ICMAgentID = 6420,
MakeRoutable = 1, SessionID = 6

### OPC
15:34:09:841 PG1A-opc Trace: MakeAgentRoutableInd - InvokeID = 80171, MRDID = 5000, ICMAgentID =
6420, MaxTasks = 2, SessionID = 6
15:34:09:841 PG1A-opc Trace: MakeAgentReadyInd - InvokeID = 80172, MRDID = 5000, ICMAgentID =
6420, MakeRoutable = 1, SessionID = 6
```

As the result OPC process removes agent from AS_NOT_READY state and puts to
AS_NOT_ACTIVE state. The NewState=AS_NOT_ACTIVE is actually the Ready state for
Chat/Email.

```
15:34:09:841 PG1A-opc Trace: SetAgentState: ASTID=6420 Periph#=15003 MRDomainID=5000 SGSTID=6928
SG#=70518(0x11376) OldState=AS_NOT_READY NewState=AS_NOT_ACTIVE Duration=0 CurLine=-1
ReasonCode=0 AgentObj=0x44535b8
```

At this moment the agent is Routable and Available from the Router perspective. The best way to
check this is to use the rttest utility:

```
rttest: agent_status /agent 6420

### 6520 is ICMAgtID

Agent CUCM.Agent_test (6420, periph# 15003)

  domain: Cisco_Voice (1), state = [nr-0:1,R], 411 secs

    CL nr   TEST_SG (6274, periph# 70520)

     L nr   CUCM_PIM1.Cisco_Voice.defa.88025 (5000, periph# 31858)

  domain: ECE_Chat (5000), state = [na-0:2,RA], 383 secs

    CL na   TEST_Chat (6928, periph# 70518)

     L na   CUCM.ECE_Chat.default.11006 (6909, periph# 54839)
```
na - NotActive

0:2 – AciteTasks:ConcurentTaskLimit

RA - R is routable (if set), A indicated the router considers the agent available for new work in this domain

> **Caution**: In ICM 11.5, 11.6 and 12.0 you can hit the defect [CSCvq11852](#) Chat and emails are not get assigned toagents even they are available. In such scenarios you do see in the rttest output [na-0:2,RD], where D means domain unavailable (as reported by app path).

Beyond that, you can check the agent state from OPCtest and Agent PG procmon utilities.

Examples:

```
opctest /cust <inst> /node PG1A

opctest: dump_agent 5000 15003


C:\icm\pcc12\ra\logfiles>procmon <inst> PG1A pim1
11:38:40 Trace: EMT Creating Mutex Global\IMTConnect_DisconnectLock
>>>>dagent 15003
```
Where 5000 is Peripheral ID where the agent is created, and 15003 is the agent PeripheralNumber.

Availability Require in Chat Entry Point

In chat initializations, your clients can see the message "Thank you for your inquiry. Our service hours are 9am-5pm PST, Monday-Friday.". Such a message can appear even when there is an agent in Ready state for a chat. To identify the agent availability the system sends the API call when clients run the Entry Point URL. The API request goes through ECE Web server to ECE application server. This availability is determined by the sessions created on the application server.

In ECE 11.6 the Availability Require looks into the MRD availability and if there is any agent available in MRD, then Chat is available. The problem here comes, that if you have 2 SG in CHAT MRD, then if there are an agent available in one of the SG, your MRD becomes active and CHAT is offered. This problem is solved in ECE 12.0 and later versions. The enhancement was done by the use of the SG in the configuration. In that case, the system also counts the Skill Groups for agents who mark themselves available for the specific MRD.
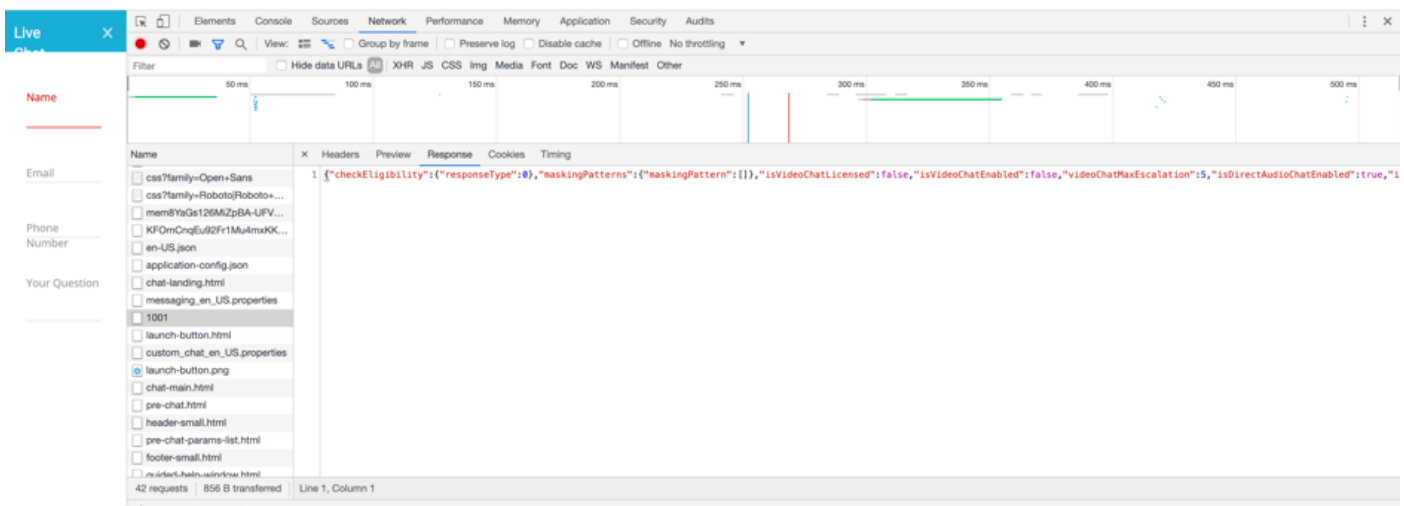
API Request:
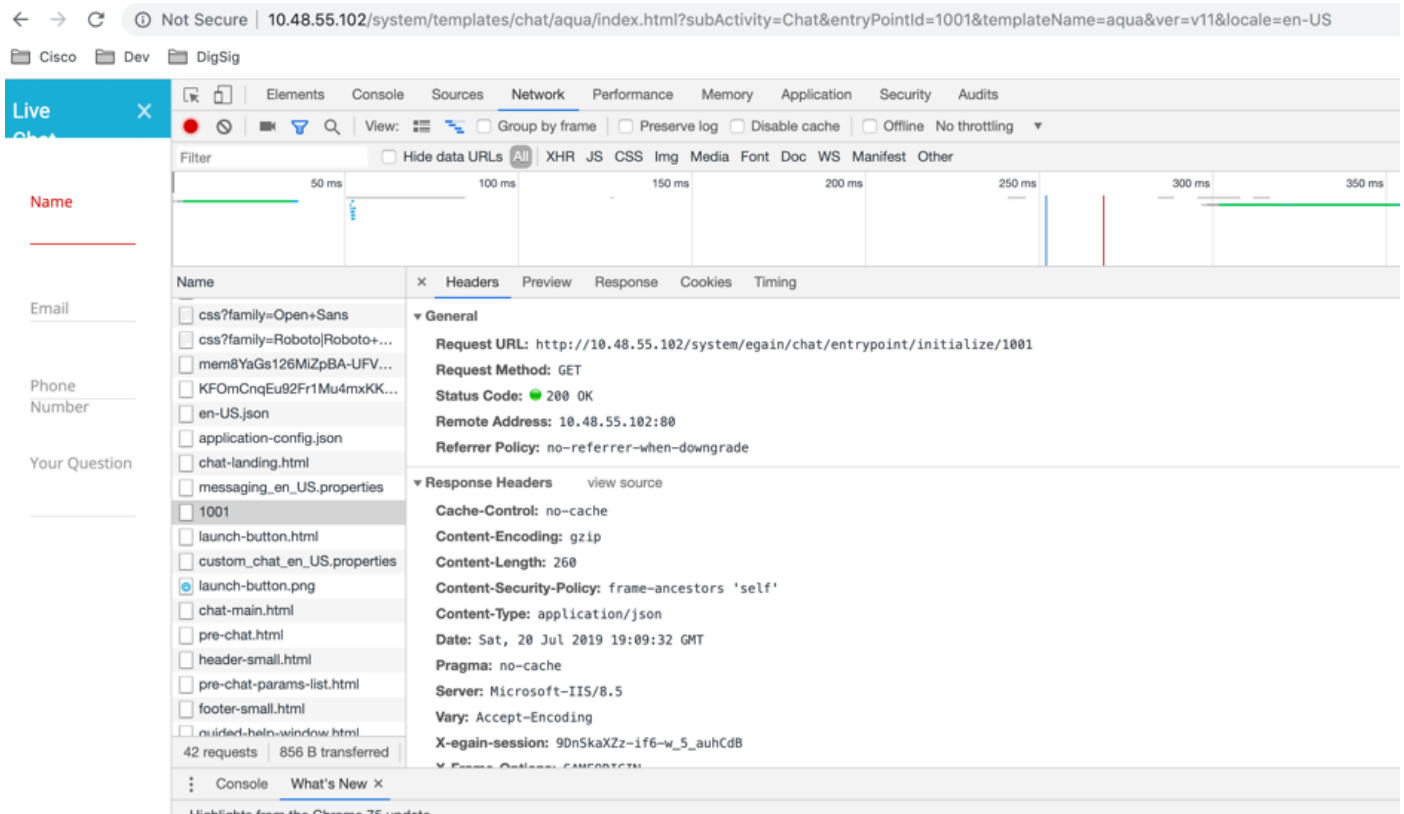
http://<ECE_WEB_Server_IP>/system/egain/chat/entrypoint/initialize/1001

where 1001 is the Entry Point ID.

API Response:

{"checkEligibility":{"responseType":0},"maskingPatterns":{"maskingPattern":[]},"isVideoChatLicen
sed":false,"isVideoChatEnabled":false,"videoChatMaxEscalation":5,"isDirectAudioChatEnabled":true
,"isChatAttachmentEnabled":false,"maxChatAttachmentSize":3,"isBlackListType":false,"isOffRecordE
nabled":false,"htmlTagMatcherRegEx":"((?:[\\r\\n|\\n]*(?:<[^>]*>)*[\\r\\n|\\n]*)*)","htmlTagMatc
herIncr":1,"isOneTagOff":true}

There are two options for how the system defines that the agent is available. Either the agent is available for a chat or there is a Queue Depth which allows to do this. The queue depth configuration allows the number of customers which can be queued when all agents are busy.

In the API response pay attention on checkEligibility: responseType value. It indicates what is an agent availability at that time.

- If it comes as **0**, it means that there is either the agent is available to take the chat or the queue depth is not reached.
- If it is **1**, it indicates that there are no agents available.
- **2** means that the max queue depth has been reached.

    Note: There are no options here to see how many agents are available at the specific time.

If an agent is available the other .js files are received by the web browser. As a result, a client does see the initial page with log in name and subject parameters for the Entry Point.

Log Collection

The API responses are available either from the client side (from the web browser Network trace) or from the ECE Application server with debug or trace level which is not recommended to keep for a long time because of the High IO that is consumed.