Agent Functionality

Wednesday, October 3, 2018 11:38 AM

We have agents log into our system that will fill real time queue member tables with the users agent ID. Asterisk will get all queues and queue members from these real time tables. When incoming calls come in, they are directed to a queue where agents will eventually connect to when available. When the agent or caller hangs up, the call is disconnected and the agent is thrown into the 'agentpool'. There they will be available to take another call when it comes in.

Everything seems to be built in to automatically place the agent back in the 'agentpool' upon completion of a call. However, we need the agent to have the ability to have a 3 or 4 way call occasionally. Asterisk allows a 3 way call in the Agent functionality but one of the requirements at my company was to have all telephony interaction be done via the application interface the agents use to enter data into. So in these situations, I have programmed the sending of AMI commands to redirect the channels to a conference room bridge (ConfBridge application).

Once in the ConfBridge the agent leaves the "Agent Realm (I call it)" because when they leave the conference room bridge they are **not** automatically sent back to the 'agentpool' to take another call. Therefore, when they leave the conference room bridge, I re-log them back in using the AgentLogin application (in silent mode).

Here is the functionality available in our system to the agent:

1. All calls to 3rd parties start with the AMI action: **ATXFER**.
    - Here the agent connects to a 3rd party and goes to context 'atxfer' extension {phone number dialed} priority 1.
    - If the agent wants to disconnect the 3rd party call, I send the AMI command: **Hangup** on the 3rd party channel.

2. After the agent connects to the 3rd party, they have multiple options on proceeding:

    A. They can complete a transfer where the caller and 3rd party are on the line and the agent returns to the 'agentpool'.
    - ➢ This is performed by the following AMI actions:
        - **BRIDGE** (caller channel and 3rd Party Channel)
        - **BRIDGEKICK** (agent channel)

    B. They can create a 3 way call by moving all parties to a conference room (ConfBridge)
    - ➢ This is performed by the following AMI actions:
        - **BRIDGEKICK** (agent channel)
        - **REDIRECT** (caller channel and 3rd party to context 'xfer',, extension 11, priority 1)
        - **REDIRECT**(agent channel to context 'xfer', , extension 11, priority 1)

    C. They can move the caller into conference room bridge A and the 3rd party into conference room bridge B (with themselves). This is for the agent to toggle back and forth on speaking to each yet keeping them from speaking to each other (until they are ready to).
    - ➢ This is performed by the following AMI actions: *NOTE: these are the same as a in B but channel variables already set will direct them into the different conference room bridges.*
        - **BRIDGEKICK** (agent channel)
        - **REDIRECT** (caller channel and 3rd party to context 'xfer', extension 11, priority 1)
        - **REDIRECT**(agent channel to context 'xfer', extension 11, priority 1)

    NOTE: All crashes (if not most) have come from the agent clicking the button to perform A, B, or C above.

    NOTE: This crash scenario I gave, crashed under option A above. [I wrote this code many months ago, and am not sure why I have BRIDGE right before the BRIDGEKICK, it doesn't hurt, because it works most of the time and the crashes are not just in option A. I have witnessed crashes in all 3 options].

    NOTE: This all works most of the time, but occasionally it crashes.

3. Once the agent leaves the conference room bridge, they are re-directed to context 'agentrelogin' extension 11 priority 1 in order log back into the agent pool as such:

```
; ------ when agent leaves the realm they will call this to get back in.
[agentrelogin]
exten => 11,1,NoOp()
exten => 11,n,AgentLogin(${agentid},s)
```

Agent Login Contexts

```
[proxy-incoming]
include => spy

exten => 1999,1,Dial(PJSIP/6126706994@${CALLROUTER})


; ---- login in the agent -------
exten => _[127]XXX,1,NoOp()
same => n,Set(CHANNEL(dtmf_features)=TH)
same => n,Set(CHANNEL(musicclass)=silence)
same => n,Answer()
same => n,NoOp(${CALLERID(num)})
same => n,NoOp(caller id name is ${CALLERID(name)})
same => n,Set(sipaccount=${GET_CHECK_AMP_LOGIN(${EXTEN})})
same => n,NoOp(sip account is ${sipaccount})
same => n,Gotoif($[${LEN(${sipaccount})}>0]?:emptysipaccount)
same => n,GotoIf($[${sipaccount}=${CALLERID(name)}]?:invalidsipaccount)
same => n,Set(CDR_PROP(disable)=1)
same => n,AgentLogin(${EXTEN})
same => n,Hangup()
same => n(invalidsipaccount),NoOp(Sip Account is Invalid)
same => n,Playback(im-sorry&tt-somethingwrong&vm-and&is-in-use&goodbye)
same => n,Hangup()
same => n(emptysipaccount),NoOp(Sip Account is Empty)
same => n,Playback(im-sorry&tt-somethingwrong&goodbye)
same => n,Hangup()

; ------- tranfer numbers -------
exten => _NXXNXXXXXX,1,NoCDR()
exten => _NXXNXXXXXX,n,Set(CALLERID(num)=)
exten => _NXXNXXXXXX,n,Set(CALLERID(name)=)
exten => _NXXXXXXXXX,n,Dial(PJSIP/${EXTEN}@${CALLROUTER})
exten => _NXXNXXXXXX,n,Hangup()

; ------ when agent leaves the realm they will call this to get back in.
[agentrelogin]
exten => 11,1,NoOp()
exten => 11,n,AgentLogin(${agentid},s)
```

Conference Room and Transfer Contexts

```
[atxfer]

exten => _NXXNXXXXXX,1,NoCDR()
exten => _NXXNXXXXXX,n,Set(CALLERID(num)=)
exten => _NXXNXXXXXX,n,Set(CDR(transfertype)=atxfer)
exten => _NXXNXXXXXX,n,Set(MONITOR_EXEC=soxmix)
exten => _NXXNXXXXXX,n,Dial(PJSIP/${EXTEN}@${CALLROUTER},,tT)
exten => _NXXNXXXXXX,n,Hangup()

exten => _NXXNXXXXXXXXXXXXXXXX,1,NoCDR()
exten => _NXXNXXXXXXXXXXXXXXXX,n,Set(CALLERID(num)=${EXTEN:10})
exten => _NXXNXXXXXXXXXXXXXXXX,n,NoOp(${CALLERID(num)})
exten => _NXXNXXXXXXXXXXXXXXXX,n,Set(CDR(transfertype)=atxfer)
exten => _NXXNXXXXXXXXXXXXXXXX,n,Set(MONITOR_EXEC=soxmix)
exten => _NXXXXXXXXXXXXXXXXXXX,n,Dial(PJSIP/${EXTEN:0:10}@${CALLROUTER},,tT)
exten => _NXXNXXXXXXXXXXXXXXXX,n,Hangup()
exten => _1NXXNXXXXXX,1,Set(CALLERID(num)=)
exten => _1NXXNXXXXXX,n,Set(MONITOR_EXEC=soxmix)
exten => _1NXXNXXXXXX,n,Dial(PJSIP/${EXTEN:1}@${CALLROUTER},,tT)
exten => _1NXXNXXXXXX,n,Hangup()

exten => _XXXX,1,MixMonitor(${CEID}-${STRFTIME(,,%C%y%m%d)}-TRANSFERAGENT-${UNIQUEID}.wav49,b)
exten => _XXXX,n,Userevent(TRANAGENTAUDIO,agentid:${EXTEN},audioname:${CEID}-${STRFTIME(,,%C%y%m%d)}-
TRANSFERAGENT-${UNIQUEID}.wav)
exten => _XXXX,n,Dial(Local/${EXTEN}@agents)


exten => _XXXXXXXXXXXXXX,1,MixMonitor(${CEID}-${STRFTIME(,,%C%y%m%d)}-TRANSFERAGENT-${UNIQUEID}.wav49,b)
exten => _XXXXXXXXXXXXXX,n,Userevent(TRANAGENTAUDIO,agentid:${EXTEN:0:4},audioname:${CEID}-${STRFTIME(,,%C%y%m%d)}-
TRANSFERAGENT-${UNIQUEID}.wav)
exten => _XXXXXXXXXXXXXX,n,Dial(Local/${EXTEN:0:4}@agents)


; --- tollfree numbers need to go out local t1
exten => _1800NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1822NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1833NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1844NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1855NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1866NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1877NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})
exten => _1888NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,${EXTEN})

exten => _800NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _822NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _833NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _844NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _855NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _866NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _877NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})
exten => _888NXXXXXX,1,Macro(outbound-call-local-T1,TRANSFER,1${EXTEN})


exten => _800NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _822NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _833NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _844NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _855NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _866NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _877NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})
exten => _888NXXXXXXXXXXXXXXXX,1,Macro(outbound-call-local-T1-with-callerid,TRANSFER,1${EXTEN:0:10},${EXTEN:10})


; this is for transfering to a conference room and to call nth parties to join.
[xfer]
exten => 11,1,Set(CDR(accountcode)=${transfer})
exten => 11,n,Set(CDR(transfertype)=3way)
exten => 11,n,Set(CDR(ast_id)=${CEID})
exten => 11,n,Userevent(Conference,depruniqueid:${UNIQUEID},callertype:${callertype},room:${room},agent:${agent},origuniqueid:
${transfer},origchannelid:${origchannel},actionid:${action},mandatacdr:${mandatacdr})
exten => 11,n,ConfBridge(${room})

exten => 12,1,Set(CDR(accountcode)=${outbound})
exten => 12,n,Set(CALLERID(num)=${calleridtosend})
exten => 12,n,Userevent(ConfCallOut,room:${room},dialed:${dialnumber})
exten => 12,n,Gotoif($[${LEN(${chanline})}>0]?eight:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=800]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=811]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=822]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=833]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=844]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=855]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=866]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=877]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=888]?toll:)
exten => 12,n,Gotoif($[${dialnumber:0:3}=899]?toll:)
exten => 12,n,Dial(PJSIP/${dialnumber}@${CALLROUTER},,tT)
exten => 12,n,Hangup()
exten => 12,n(toll),Dial(PJSIP/1${dialnumber}@${CALLROUTER},,tT)
exten => 12,n,Hangup()
exten => 12,n(eight),Dial(${chanline},,tT)
exten => 12,n,Hangup()
```

Agent Telephony System

- We have agents log into an Asterisk 10 (SIP Proxy) box using Eyebeam softphones via 'chansip'. The call gets routed to our main Asterisk 13 box where agents take calls. There we use PJSIP (with Asterisk 13).

My thoughts and things witnessed and things I can't explain.

1. The context **proxy-incoming** is used when the agent logs into Asterisk from their software phone. However, some backtraces from the system crash show that it crashed within this context. However, the FULL log in Asterisk only shows the initial login attempt in the morning and the login attempt **after** the crash for the agent that triggered the crash. Almost like our SIP proxy is remembering and re-logging in.

2. The AMI actions above are sent and performed consecutively and 2 involve the agent channel .
   - **BRIDGEKICK** (agent channel)
   - **REDIRECT** (caller channel and 3rd party to context 'xfer',, extension 11, priority 1)
   - **REDIRECT**(agent channel to context 'xfer', , extension 11, priority 1)

   My log files show that they are all sent basically on the same millisecond. This makes me believe there **may** be a race condition going on here.