# ACD – Automatic Call Distribution

Right after Auto Attendants and Voice Mail, this is probably the most sought after telephony application and by far the most complicated.

ACD is the application you reach when you dial a number such as a sales or support line and get a message like "*Thank you for calling, calls are answered in the order they are received. Please stay on the line and you will be connected to our next available agent*."

Usually there are a group of agents who take a particular type of call but with the capabilities of Asterisk, they can actually share more than one type of queue such as sales and support.

This was rarely possible with legacy telephone equipment such as the Nortel Networks Meridian Option 81 C's. They provided an ACD function but it was very limited in its capabilities. It typically had three keys programmed on a display phone. The three keys were the ACD key or line that they actually answered a Make Set Busy Key (MSB) and a Not Ready Key (NRD).


The agents would log in with a fixed password and as long as the MSB and NRD keys were off, they were in the queue to take calls. If they needed a few extra seconds to finish up entering data after taking a call, they could hit the NRD key and it would temporarily block incoming calls to that agent but not take them out of the queue. When they left for lunch or in the evening, they were supposed to hit the MSB and NRD keys which would log them completely out of the queue. If they forgot, a late night caller might actually get in the queue but never actually reach a live person.

Asterisk is much more flexible and can even take an agent out of the queue if calls presented to their phone go unanswered for a period of time. This allows calls to go somewhere else like a night queue or just a voice mailbox.

Asterisk also provides many more options such as announcing the caller's position in the queue, the anticipated wait time, etc. It also provides the capability of presenting different hold music for different queues. For example, the sales queue might have promotional messages on hold while the support queue might have troubleshooting tips.

While legacy phone systems often used expensive ACD only sets, Asterisk can use anything from a single line analog set to a sophisticated SIP set. Most often, agents actually have no physical set at all but rather a "softphone" application on their computer and a USB headset.

This eliminates one more piece of hardware on their desktop and with the right application; the incoming call can even read the caller ID of the caller and pop up their account information before the agent answers the call.

While you could use an analog phone, it is not recommended because one of the critical features called system state is only available with a SIP phone. The system state actually counts how many calls have been presented to an agent and whether they are on a call or not. This can lead to sophisticated reporting capabilities and well as enhancing the feature set. You can take advantage of this by adding callcounter=yes to the general section of sip.conf.

Let's get started building an ACD queue and learning the how's and whys as we do so. Much of the information for this exercise was derived from [http://tinyurl.com/9zrbxae](http://tinyurl.com/9zrbxae) which is [http://www.asteriskdocs.org](http://www.asteriskdocs.org) the html online version of the third edition of the Asterisk the Definitive Guide book. Chapter 13 deals with ACD and covers many more possibilities and options than we can cover in one class. It is recommended that you take a look at it once you have done this exercise and you will probably get more ideas on how to enhance your queues.

The important configuration file is located at /etc/asterisk/ and is called queues.conf.

To make it easier to understand what we are doing, we will rename this file by using the mv command**: cd /etc/asterisk** and **mv queues.conf queus.orig** This will save the original file for reference and let us start with a clean slate.

**vi queues.conf** to create a new file, hit I for insert and add the following material:

```
[general]
autofill=yes                ; distribute all waiting callers to all available
                            ; members
shared_lastcall=yes         ; respect the wrapup time for members logged into
                            ; more than one queue

[StandardQueue](!)          ; template to provide common features
musicclass=default          ; play [default] music
strategy=rrmemory           ; use the Round Robin Memory strategy
joinempty=no                ; do not join the queue when no members available
```

```
leavewhenempty=yes       ; leave the queue when no members available
ringinuse=no             ; don't ring members when already in use (prevents
                         ; multiple calls to an agent)

[sales](StandardQueue)   ; create the sales queue using the parameters in the
                         ; StandardQueue template
```

**Esc:wq** to save it and quit.

You don't have to type the semicolons and text after them as they are only comments but it helps to document your code heavily so when you go back later, you or someone else can figure out easily what you were doing.

This creates our first queue which is called "sales".

In order to get Asterisk to actually enable this queue, we need to change to the Asterisk command line, **asterisk –rvvv** and run this command from the Asterisk CLI>: **module reload app_queue.so** You should see a message like this:

-- Reloading module 'app_queue.so' (True Call Queueing)

This loads your queues into memory. Now you can check them by entering this command from the Asterisk CLI>:

**queue show**

You should get a response similar to this:

sales has 0 calls (max unlimited) in 'rrmemory' strategy
(0s holdtime, 0s talktime), W:0, C:0, A:0, SL:0.0% within 0s
No Members
No Callers

Now we have a queue but we need to create a dialplan to use it.

We are going to create a separate context for our queue and point a dummy extension to it. This will allow us to use our Polycom phone as an ACD agent and test each other's queues by calling your partner. In the real world, your menu or Auto Attendant might have an option to "*Press 2 for sales, 3 for support.etc*"
**cd /etc/asterisk** if you are not already there and **vi extensions.conf**

Add this text somewhere above your default context

```
[queues]
exten => 7001,1,Verbose(2,${CALLERID(all)} entering the sales queue)
same => n,Answer()     ; Very Important, won't play MOH or announcements
                       ; if you don't answer the call first.
same => n,queue(sales)
same => n,Hangup()
```

```
exten => 7002,1,Verbose(2,${CALLERID(all)} entering the support queue)
same => n,Answer()      ; Very Important, won't play MOH or announcements
                        ; if you don't answer the call first.
same => n,queue(support)
same => n,Hangup()
```

Now in your default context above your _XXXX stanza that allows you to dial other systems in the room, add something like this:

```
exten => NN67,1,GoTo(queues,7001,1)
```

This creates a dummy extension that will point to your queue. Replace the **NN** in the example with the first two digits of **YOUR** dialplan.

**esc:wq** to save your extensions.conf and then **asterisk –rvvv** to get back to an Asterisk CLI>

(Or you can also have a terminal open on the F1 key to edit your configurations and a terminal open on the F2 key at the Asterisk CLI>)

At the Asterisk CLI> **dialplan reload**

Try calling your dummy number from your phone and watch the Asterisk CLI> The call will fail and you will get some messages about entering the sales queue and unable to join the sales queue. Why does this happen?

You have a queue and you have defined a number to get to it but who is the agent that will answer the calls for that queue?

There are several ways to add an agent to the queue. You can do it from the Asterisk CLI> or you can create another stanza in extensions.conf to allow a person to log into the queue as an agent. We will use the Asterisk CLI> method first.

From the Asterisk CLI> **queue add member SIP/MACADDRESS to sales** Look at the label on back of your phone for the mac address or copy it from elsewhere in your extensions.conf file.

Now from the Asterisk CLI> **`queue show sales`**

```
sales       has 0 calls (max unlimited) in 'rrmemory' strategy
(0s holdtime, 0s talktime), W:0, C:0, A:0, SL:0.0% within 0s
   Members:
      SIP/MACADDRESS (dynamic) (Not in use) has taken no calls yet
   No Callers
```

Now call your dummy number or your neighbors. Your phone (or your neighbors) should start ringing immediately and you will hear music on hold until they answer.

Congratulations! Your first ACD queue is functional. It's not very practical at the moment but we will fix that as we move on.

There are a number of problems with our queue as it stands right now. How does the caller know what department they have called? What happens at night when the office is closed and no one answers.

Let's start addressing these issues one by one then we will get on to adding a way for the agents to log in and out of the queues themselves.

Let's deal with the closed scenario first. We need to create a mailbox for the queue and have calls go to it whenever there are no agents logged in.

**vi /etc/asterisk/voicemail.conf** Go to the bottom of the file, hit I for insert and add this line:

7001 => 7001,Sales Department,sales@widget.com,,Tz=Phoenix Note the two commas after widget.com.

This creates a voice mailbox for sales and will even e-mail notifications of messages to sales@widget.com which would probably be a link on your company web site as well.

**Esc:wq** to save and now **vi /etc/asterisk/extensions.conf** again. Find the stanza where you defined the queues and add the line in bold.

exten => 7001,1,Verbose(2,${CALLERID(all)} entering the sales queue)
exten => 7001,n,queue(sales)
**exten => 7001,n,Voicemail(7001,u)**
exten => 7001 n,Hangup()

**Esc:wq** to save, **asterisk –rvvv** and **dialplan reload**.

Now, from the Asterisk CLI> **queue remove member SIP/MACADDRESS to sales** Again, use the mac address for your phone.

This will log your only agent out of the queue.

Now we need to record your voicemail message for the queue. Dial 8500 and when Comedian mail answers, dial 7001 for the mailbox and 7001 for the password. Then press 0 (zero). This gives you the mailbox options.

Press 1 and record an unavailable message, something like this: *Thank you for calling sales. Our offices are closed but you may leave a detailed message and we will contact you during the next business day*.

Once you have done that and followed the instructions to save it, try calling your queue now. It should go right to the voicemail and play back your message. You can leave a message if you wish. If we had mail configured and you had used a real e-mail address for sales, it would e-mail a message to that address with a wav file of the voicemail message attached.

Now we need to fix the problem of letting callers know what department they have reached.

Using your recording prompt context, record a message something like this: *You have reached the Acme sales department. Calls are answered in the order they are received. Please wait for the next available agent.* Rename that prompt something like acme-sales.wav and copy it to /usr/share/asterisk/sounds/en

Now, **vi /etc/asterisk/extensions.conf** again and change your queues stanza to look something like this:

Add the two lines in bold and be sure to change the 1 on the 7001,1,Verbose line to an n.

[queues]

**exten => 7001,1,Answer()**
**exten => 7001,n,Playback(acme-sales)**
exten => 7001,**n**,Verbose(2,${CALLERID(all)} entering the sales queue)
exten => 7001,n,Queue(sales)
exten => 7001,n,Voicemail(7001,u)
exten => 7001 n,Hangup()

**Esc:wq** to save, **asterisk –rvvv** and **dialplan reload**

Now when you call your queue, you should hear the introductory message and if you have no agents logged in, it should drop to voicemail again.

Now let's deal with allowing the agents to log in and out by themselves.

**vi /etc/asterisk/extensions.conf** and we will create another context to allow the agents to log in and out by dialing something on their phones.

Go to just under where you created the queue context and create this one:

[QueueMemberFunctions]

```
exten => *52,1,Verbose(2,Logging In Queue Member)
  same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
  same => n,AddQueueMember(sales,${MemberChannel})
  same => n,Playback(agent-loginok)
  same => n,Hangup()

exten => *54,1,Verbose(2,Logging Out Queue Member)
  same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
  same => n,RemoveQueueMember(sales,${MemberChannel})
```

```
  same => n,Playback(agent-loggedoff)
  same => n,Hangup()


exten => *56,1,Verbose(2,Logging In Queue Member)
  same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
  same => n,AddQueueMember(support,${MemberChannel})
  same => n,Playback(agent-loginok)
  same => n,Hangup()

exten => *58,1,Verbose(2,Logging Out Queue Member)
  same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
  same => n,RemoveQueueMember(ssupport,${MemberChannel})
  same => n,Playback(agent-loggedoff)
  same => n,Hangup()
```

The sound files agent-loginok and agent-loggedoff should already exist.

Again, **Esc:wq** to save, **asterisk –rvvv** and **dialplan reload**.

Now, without picking up the handset on your phone, dial *52 and lift the handset quickly to your ear. You should hear the “Agent-logged in ok” message and see it in the asterisk CLI>

Now have your neighbor call your queue and don’t answer immediately. They should hear the introductory message then drop to music while your phone is ringing. As soon as you answer, the music they are hearing will stop and you will be connected.

Hang up your phone, dial *54 then pick up the handset. You should hear “Agent logged out” and when someone calls your queue number, they will hear the introductory message them go to voicemail.

We’re on the home stretch now. We have a working queue, a way for agents to log in and out, and even a way to deal with callers after hours. Now we need to add a few “bells and whistles” to our queue to make it more informative to our caller.

Asterisk can tell the caller if they are next in line and even thank them for holding regularly if we wish. It can even tell them the approximate wait time based on the average length of the calls that the queue agents are taking. This is commonly used in ACD queues to keep the caller from hanging up.

One measure of efficiency of a queue is looking at the total number of calls handled vs the number abandoned where the caller hung up before getting answered. A high abandonment rate means you need to have more agents taking calls or else some way to faster process each caller.

**vi /etc/asterisk/queues.conf**, hit I for insert and add the following under the existing ringinuse=no line.

```
musicclass=default
; -------- Announcement Control --------
announce-frequency=30          ; announces caller's hold time and position every 30
                               ; seconds
min-announce-frequency=30   ; minimum amount of time that must pass before the
                               ; caller's position is announced
periodic-announce-frequency=45  ; defines how often to play a periodic announcement to
                                 ; caller
random-periodic-announce=no     ; defines whether to play periodic announcements in
                                 ; a random order, or serially
relative-periodic-announce=yes  ; defines whether the timer starts at the end of
                                 ; file playback (yes) or the beginning (no)
announce-holdtime=once         ; defines whether the estimated hold time should be
                               ; played along with the periodic announcement
announce-position=limit        ; defines if we should announce the caller's position
                               ; in the queue
```

Again, the semicolons and text following them are comments for documentation. You don't need to type them if you wish.

**Esc:wg** to save, **asterisk –rvvv** and **queue reload**

Now when your queue is called, the caller should hear the introduction then music. If you ignore the ringing agent phone for a while, the caller should hear some additional messages such as next in queue, etc.

Another interesting test is to have an agent log out while there is a call in the queue? What do you think will happen to those calls?

---

This is but a sampling of the many possibilities of creating queues and managing them in Asterisk. We did not touch on logging or printing reports based on the calls taken by the queues.

Also, many times you get the familiar "*Calls may be recorded or monitored for quality assurance and training*" message when you enter a queue. How much code would you have to add to the [queues] context to accomplish that?

The chapter in the book gives you an example of how the agent can initiate the recording by dialing a few digits.

Here is another scenario to explore. You receive a harassing or threatening call such as a bomb threat, how can you bridge another person on the line without the caller knowing it so as to have corroboration?