Asterisk and Recording Calls

Often times we want to record a telephone conversation for certain purposes such as providing a record of a call to customer service to document what they agreed to order.

Check for the legality of this in your jurisdiction. Arizona only requires that one party be aware that the call is being recorded. It can be either the caller or the called party.

In our example, this might be a call to customer service and we want to record the calls to see how our customer service representatives are dealing with customers for training purposes.

Asterisk provides several ways of starting a recording. The customer service agent might dial a number to start the recording but in this example, we alert the caller that the call is being recorded and start it automatically when the called number answers.

This uses the **Monitor** feature.

Here's how it works:
The call to an extension is answered then a message is played to them informing them that the call is being recorded.

The Monitor application is started and it creates a wav file in the default folder which is /var/spool/asterisk/monitor.

Note that by default, the call is recorded in two files, one for the incoming speech and one for the outgoing speech. We can combine the two files by using the sox (Sound eXchange) application which is commonly installed on Linux.

The **m** in the monitor line tells sox to combine the two files into one and delete the original two. This makes it convenient for us to hear both sides of the conversation at once.

```
; Records calls to extension and creates a single output file The file
; would get overwritten each time the number receives a call.
exten => 6027,1,Answer()
exten => 6027,n,Playback(is-now-being-recorded)
exten => 6027,n,Monitor(wav,6027_rec,m)
exten => 6027,n,Dial(SIP/6027,20,rt);
exten => 6027,n,Hangup()

; Same as above but creates output file based on time. Since each call
; arrives at a different time, we have a recording for every call made
; to this number.
exten => 6027,1,Answer()
exten => 6027,n,Playback(is-now-being-recorded)
exten => 6027,n,Monitor(wav,${DATETIME},m)
exten => 6027,n,Dial(SIP/6027,20,rt);
exten => 6027,n,Hangup()
```

```
${DATETIME} Current date time in the format: DDMMYYYY-HH:MM:SS is now deprecated
but still works in Asterisk 1.8.

You can use ${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)}) instead to have it format the
time stamped name of the recording in a more user friendly format. Check out how
to format this string to report the time in local time.

This example should work to replace the second example above.
```

exten => 6027,1,Answer()
exten => 6027,n,Playback(is-now-being-recorded)
exten => 6027,n,Monitor(wav,${ STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)},m)
exten => 6027,n,Dial(SIP/6027,20,rt);
exten => 6027,n,Hangup()

Description

Formats the time specified by <epoch>, localized to <timezone>. The format comes directly from the underlying C function strftime(3). If <epoch> is not specified, defaults to the current time.

<timezone> likewise defaults to the timezone on the host computer. A list of possible timezones may be obtained from the directory listing in /usr/share/zoneinfo. The default format is %c. This example sets the variable CallTime to the correct local time string in the format Unix utilities like `touch` want, i.e. YYYYMMDDhhmm:

This might be an example of an S extension to format the time by using the epoch and specifying the GMT offset.
exten => s,n,set(CallTime=${STRFTIME(${EPOCH},GMT+8,%C%y%m%d%H%M