Global Knowledge®

Expert Reference Series of White Papers

Avaya Aura Contact Center Scripting Demystified

Avaya Aura Contact Center Scripting Demystified

Brett Hanson, Principal Technical Instructor - Global Knowledge

Introduction

It is time to de-mystify the procedure of crafting Call Processing instructions and make it a skill that you can comfortably use. By possessing the skill of scripting, you can manage a call center more completely. The ability to redesign Call Flow and trouble-shoot call routing expands your Contact Center repertoire. Why is scripting daunting? For most people it is for a lack of opportunity to practice. The few who specialize in scripting will design and commission a system. Then it "simply works." If modest changes need to be made, it is fairly simple to change a RAN route or skillset name and re-activate the script. Many skilled Contact Center support personnel have not graduated beyond "stare and compare" script editing.

How will this white paper help you? What kind of information can help you increase your skills? By introducing you to an identifiable pattern that virtually all Contact Centers follow and associating key elements of that pattern with the corresponding scripting code.

Many people can interpret a script that has already been created, but feel unconfident in their ability to create a script from scratch. The argument here is that if you know the outline and the basic formulas, you can create the code. If you already have a background with scripting, this paper may be all you need to take a turn at writing new scripts. If you are new to scripting, you will gain insight into some fundamental, key concepts. The ambition of this document is to elevate your ability, regardless of experience and to inspire you to work with scripts instead of avoiding them. If we begin with a blank page, the process is overwhelming. If we have a series of questions to answer – a kind of plug and play approach, script writing is manageable and satisfying. It all begins with "IF!"

Note that with CC7, Nortel (since purchased by Avaya) introduced the Service Creation Environment. SCE is a program that allows the writing of text-based scripts and graphical flows. In this document we focus on text-based scripts for voice contacts that can be applied to any release of Contact Center. Look for separate white papers that explore the mysteries of graphical flows and call processing instructions for multimedia contacts.

Four Primary Components of Scripting

Scripting design can be classified into four primary components:

1. Call Sorting/Filtering

This process takes place in the master script of identifying the purpose of the call and distributing it to the appropriate primary script. (The Master Script and its evolution to the Contact Router will be covered in the exploration of graphical flows in a separate white paper.)

2. Closed Condition Check

a. Scheduled Closures
   i. Holidays
   ii. Weekends and after hours
b. Unscheduled Closures
   i. Meeting/Emergency/issue that arises during business hours
   ii. Weather conditions or circumstances preventing agents from staffing the contact center
   iii. Skillset closed during regular business hours

3. Basic Treatment

a. Queuing
b. Initial Announcements

4. Loop Function

a. Repeat announcements
b. Check for changes to the call's status

Almost all well-written scripts have code that meets the above criteria. You will certainly encounter scripts that have more than the above – especially when integrating the Contact Center Manager Server with other systems, but we shall focus on the basics. It is astounding how much scripting logic you can extrapolate/create/discover once you understand the fundamentals.

1. Call Sorting/Filtering

This process takes place in the master script of identifying the purpose of the call and distributing it to the appropriate primary script. (The Master Script and its evolution to the Contact Router will be covered in the exploration of graphical flows in a separate white paper.)

2. Closed Condition Check

We are going to start with the second point in the outline above. The key to understanding scripting logic - the essence of transforming the outline above to actual code - is the IF-THEN-END IF statement and its brother, the IF-THEN-ELSE-END IF statement. Fundamental to mastering the art of script writing is learning how to play the contingency game. Think like a boy/girl scout and be prepared. It's all well and good if: the contact center is open, the agents are logged in, there aren't too many calls waiting, and it is sunny outside. But what if the contact center is closed, or all the agents are logged out, or there are lots of calls waiting, or there is a storm? Conditions vary in a call center, and we must determine the current conditions for the caller in order to process his call appropriately.

The IF-THEN-END IF is the scripting language method for asking a question and processing the answer. The Contact Center manages all kinds of conditional information in the form of intrinsics. Circumstances are continually changing and intrinsics tell us the current state of affairs. Our job as script writers is to know how to ask the right questions and when to ask them. The outline above identifies the "what" and the "when;" let's address the "how."

2a – Scheduled closures

Consider:

```
IF DATE = JAN 1 THEN
      GIVE RAN 30
      DISCONNECT
END IF
```

The question we are asking here is, "what is the date for the call being processed?" If the system answers that indeed the caller has reached our contact center on the first day of the year, the system will execute the THEN statements. In this instance, our caller will hear a recording from RAN 30 and then be disconnected. There is no need to examine any further script instructions because the call center is not open for business.

How about:

```
IF DAY OF WEEK = SATURDAY..SUNDAY
   OR TIME OF DAY <> 8:00..16:59 THEN
      GIVE RAN 31
      DISCONNECT
END IF
```

Now we are asking two questions in one statement: is it the weekend or outside of regular business hours? The "OR" on the second line indicates that either condition is acceptable for identifying a scheduled closure. The business will be closed on the weekend or when the time of day is not (the "<>" means "not equal to") during normal business hours.

2b – Unscheduled closures

```
      IF NOT LOGGED OUT AGENT 12345 THEN
            GIVE RAN 32
            DISCONNECT
      END IF
```

If an unplanned closure (e.g., meeting, network problem, building evacuation) needs to occur during regular business hours, there are a variety of ways to address the situation through scripting. The most common approach is to implement the "emergency agent." An agent profile is created in Contact Center Management that does not belong to an actual person. Instead, the supervisor or administrator is instructed to log in the emergency agent when the unscheduled event occurs. There is no intrinsic for "logged in agent;" therefore, we must employ a double negative. If the agent is not logged out, he is logged in.

```
      IF weather_boolean_gv = TRUE THEN
            GIVE RAN 33
            DISCONNECT
      END IF
```

The preceding scenario works very well when the unscheduled event takes place while the contact center is staffed. We must take a different approach when weather conditions prevent personnel from reaching the contact center in the first place. A variable that can be set to TRUE or FALSE (weather_boolean_gv) is configured and assigned a value of FALSE for regular call processing. When weather or some other event prevents agents from physically staffing the call center, an administrator makes a browser connection to the CCMA (Contact Center Manager Administration) interface (typically via Virtual Private Network) and changes the value of the variable to TRUE. Now callers will hear the announcement in RAN 33.

```
      IF NOT OUT OF SERVICE customer_service_sk THEN
            QUEUE TO SKILLSET customer_service_sk
            WAIT 2
      ELSE
            GIVE RAN 34
            DISCONNECT
      END IF
```

Whoa! What's happening here? Now the ELSE clause is making an appearance. Let's make sense of this. The condition we are evaluating is the status of the skillset. We have checked for scheduled closures, and we have checked for unscheduled closures. It is possible that the call arrives during regular business hours and no previous unscheduled closure check is in effect. However, we must determine if one or more agents is staffing the skillset. During a shift change, a faulty Agent to Skillset Assignment, or simply tardy agents could cause the skillset to be out of service.

The condition being evaluated in our IF statement is whether the skillset is in service (open). Again we must use a double-negative to determine something positive. If the skillset is staffed, we queue the call to the skillset (the THEN statements are executed). When the skillset is not staffed, our IF condition is FALSE and, as a result, the ELSE statements are executed. Consider how this example varies from the previous examples that lacked an ELSE clause. The issue is what happens when the condition (intrinsic) being evaluated by the IF statement is not met. When there is no ELSE clause, the system follows the next command – that which follows the END IF – provided, of course, that we did not disconnect the call. When there is an ELSE clause, the system follows the instructions that follow the ELSE when the IF condition is not met.

3. Basic Treatment

Queuing

After checking for scheduled and unscheduled closures, our scripting logic dictates that we must be open if the call has made it to this point in the script (in other words, we haven't disconnected or otherwise treated our call for a closed condition). Our previous action was to queue the call to the appropriate skillset, provided that the skillset was in service. If an agent is available, the Contact Center Manager Server will use **Skill Based Routing** to direct the call to an agent. If no agents are available, the call will continue through your script commands and now is the time to play an announcement indicating the name of the business and the status of the agents. (It is important to note that the announcement can be played before the call is queued, especially if the message must be heard by all callers – such as a legal disclaimer.)

The **GIVE RAN** or **GIVE IVR** command is used to play the message. Use **GIVE RAN** if you have RAN routes, use **GIVE IVR** if you CallPilot or a voice messaging system.

Skillset Intrinsics

An option at this point is to use skillset intrinsics to assess the status of the skillset. If the skillset is particularly busy, you can provide some additional option or treatment to the call. Instead of automatically playing a message about "the agents are busy and will be with you shortly," you can determine if the wait is likely to be longer than usual and provide the option to leave a message:

```
IF THE QUEUED COUNT customer_service_sk >
   3 * LOGGED AGENT COUNT customer_service_sk THEN
      GIVE IVR 8000 WITH TREATMENT 1000
ELSE
      GIVE IVR INTERRUPTIBLE 8000 WITH TREATMENT 2000
END IF
```

In this case, we are analyzing the number of callers in the skillset queue and comparing that value to the number of agents logged into the queue. If the number of waiting callers is more than three times the number of agents staffing the queue, the caller is directed to IVR (CallPilot) queue and presented with a menu service that informs the caller of situation and gives him the option to leave a message (1000 is an entry in the Service Directory Number table of CallPilot that points to the menu service). If the described criteria is not met, the call is pointed to the ELSE clause, and the caller hears the standard announcement naming the company and the intent to answer the call as soon as possible. The INTERRUPTIBLE syntax is required when using the GIVE IVR command while the call is queued so the CCMS can deliver the call to an agent if one becomes available during the playback of the announcement. As a reminder, the code we are discussing would immediately follow the END IF portion of the IF statement that checked for the status of the skillset and queued the call when the skillset was in service.

Now that caller has heard the initial message (provided he did not opt out by leaving message and hanging up), use the GIVE MUSIC command to play music until the next announcement is played.

4. Loop Function

Oh No! The dreaded loop! No one wants to be here because now the wait time is starting to grow. While we can't make the agents work faster through script commands, there are a few simple techniques we can use that will make the extended wait more interesting. How about this:

```
      (a)    ASSIGN 1 TO Loop_counter

      (b)    SECTION Loop_Around

      (c)    WAIT 45

      (d)    IF NOT QUEUED THEN
             GIVE RAN 35
             DISCONNECT
      END IF

      (e)    IF NOT LOGGED OUT AGENT 12345 THEN
             GIVE RAN 32
             DISCONNECT
      END IF

      (f)    WHERE Loop_counter EQUALS
             VALUE 1: GIVE IVR INTERRUPTIBLE 8000 WITH TREATMENT 3001
             VALUE 2: GIVE IVR INTERRUPTIBLE 8000 WITH TREATMENT 3002
             VALUE 3: GIVE IVR INTERRUPTIBLE 8000 WITH TREATMENT 3003
             VALUE 4: GIVE IVR INTERRUPTIBLE 8000 WITH TREATMENT 3004
             VALUE 5: GIVE IVR INTERRUPTIBLE 8000 WITH TREATMENT 3005
                                ASSIGN 0 TO Loop_counter
      END WHERE

      (g)    ASSIGN Loop_counter + 1 TO Loop_counter

      (h)    EXECUTE Loop_Around
```

Let's examine what is taking place in this loop.

(a) The code is declaring an initial value for the call variable **Loop_counter**. It is like saying, "X = 1"; Loop_counter is like X in a simple math formula. We will use this call variable to keep track of how many times the caller passes through the loop. Since the call is about to pass through the loop for the first time, we are setting the value to 1.

(b) This is a sign post; a reference point. The system will return to this line as a result of the last line of instruction in the loop (h).

(c) This **WAIT** command is filled with music from the previous **GIVE MUSIC** command.

(d) Here we make sure that the call is still queued. If the agents log out during the call's wait in the loop, or if an assignment puts all of the agents staffing the skillset into standby, then we detect the fact that now the caller is not going to get answered and give him the sad news before we disconnect. Of course this should rarely occur, but it is an important fail-safe.

(e) The "emergency" agent might get logged in while the caller is waiting in the loop. We must identify this change of events and respond accordingly. It is exactly the same code as we discussed in the **Unscheduled Closures** section.

(f) This code can make a big difference in the caller's perception of the wait process. What we are doing here is looking at the value of the Loop_counter and playing a different announcement each cycle through the loop. This is possible because we are changing the value of the loop counter each time the call passes through the loop (see (g) below for the explanation of how the Loop_counter is changing). Company history, interesting facts, even jokes could serve as the content of these announcements. The **ASSIGN 0 TO Loop_counter** starts the series of recordings all over again after the fifth pass through the loop. You can create more recordings and corresponding VALUE statements if you expect longer wait times.

(g) The system will increase the value of the variable each time the call executes the loop via the instructions in this line. This code takes the current value of our call variable and increases it by 1. If Loop_counter is a value of 1 at the beginning of the statement, it becomes a value of 2 by the end.

(h) This line tells the system to continue the loop by going to the section with the same name (b).

Conclusion

Scripting doesn't seem quite so mysterious now, does it? The outline (four primary components) provides flow and order. The IF-THEN-END-IF lets us examine the current conditions for the call and respond accordingly.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge. Check out the following Global Knowledge courses:

Avaya Aura™ Contact Center - Communication Control Toolkit and Contact Center Multimedia Implementation (3607)

Avaya Aura™ Contact Center - Installation and Configuration (3608)

Avaya Aura™ Contact Center - Service Creation Environment (3610)

Avaya Aura™ Contact Center Administration (3609)

Avaya Aura™ Contact Center Engineering (3612)

Avaya Aura™ Contact Center Fast Track - Installation, Administration, and Scripting (3621)

Avaya Aura™ Contact Center Networking (3611)

Avaya Aura™ Contact Center Reports (3613)

What's New with Avaya Aura™ Contact Center - Administration and SCE Scripting (3619)

What's New with Avaya Aura™ Contact Center - Installation and Configuration (3618)

What's New with Avaya Aura™ Contact Center - Multimedia (3620)

What's New with Avaya Aura™ Contact Center for Sales Engineers (3617)

For more information or to register, visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a sales representative.

Our courses and enhanced, hands-on labs and exercises offer practical skills and tips that you can immediately put to use. Our expert instructors draw upon their experiences to help you understand key concepts and how to apply them to your specific work situation. Choose from our more than 1,200 courses, delivered through Classrooms, e-Learning, and On-site sessions, to meet your IT and business training needs.

About the Author

Brett Hanson is an instructor of Avaya Contact Center solutions. He traces his career in the Telecom industry from 1988, when he established himself as a voice processing engineer. He specialized in voice mail integration when voice mail systems were in their infancy, training, installing and maintaining voice mail systems throughout Southern California until establishing himself at the La Palma Training Center in 1994. Teaching Norstar and later the Meridian 1 (including Meridian Max), Brett moved to the ground floor of Call Center evolution when Symposium Call Center Server was introduced in the late 1990s. Brett became an employee of Global Knowledge in 2000 via the acquisition of Nortel training by Global Knowledge. As the call center product has evolved so has Brett's experience and his expertise. Brett is certified on the latest release of the call center product which has evolved to Avaya Aura Contact Center. He has been commended for his training excellence and praised by Avaya engineers for his depth of knowledge with the product.