

Using Advanced Call Router with SQL

Introduction AltiGen's Advanced Call Router (ACR) is an add on application that can be used to provide a higher level of granularity for inbound call routing than standard DNIS and Caller ID based routing allow by using an auto-attendant to gather information from a caller. This Data Directed Routing (DDR) can be used to ensure that the caller gets routed to the RIGHT agent, thus reducing the number of parties that a call may have to be routed through, decreasing the number of transfers, and increasing caller experience

AltiGen has provided other ACR configuration documents for static configuration of ACR, such as how to configure ACR to route calls based on collecting a caller's zip code. While these types of configuration work well for datasets that do not change over time, it often more useful in a real world scenario to provide this routing based on data that constantly changes, such as purchase order numbers, and customer ID numbers. Typically, this information is added to (or created by) a database as needed by a user, a web interface, or other similar mechanism. ACR can be configured to run a SQL query against such a database to retrieve a record, and route calls accordingly. In this manner the ACR rule set can automatically adjust to business needs.

Notes On This Document

Instructions for installing and configuring are outside of the scope of this document, and outside of the scope of AltiGen support. It is assumed that the technician performing these steps is already familiar with basic SQL syntax, such as SELECT queries. For purposes of this document MSSQL Server 2008 R2 Express is used as the database. An ODBC connector is configured to connect to this DB. Configuration of the ODBC connector is also outside the scope of this document

Scenario 1

Widgets, Inc. has a sales team consisting of 5 sales agents. Orders are placed by customers via a web page, where the order number is created. Occasionally, a customer will call in to follow up on the status of an order. Widgets, Inc. would like to route inbound callers to agent that processed their order. They plan to use their Auto Attendant to prompt the caller to enter their order number to accomplish this. Because order numbers are generated automatically, and continuously, it is not possible to manually keep the ACR route rules up to date.

Because the orders are generated on line, and allocated to a sales agent by the ordering system, the data is already stored in a database. A table called order2agent has been configured to provide only two columns of data, order_id and agent, that are used to match these. An ODBC should already be configured on the server running the Advanced Call Router to retrieve data from this database. For this example scenario, an ODBC with a DSN of "acr_sql" will be used. A sample dataset from the database looks like this:

Note that the table above can be thought of as only a small set of the actual number of orders. Every new order that comes in will populate a new row in this table.

The first step to configure will be setting up the auto-attendant to collect the order number. The order number

is a seven digit string, so the auto attendant will be structured as follows:

Item O1 in the AA is the digit collection for the order number.

Item O1& in the AA is where the collected digits get passed to the Advanced Call Router.

Item O1&& is the fail over in case the ACR cannot be reached.

Next, with ACR connected, configure the Default Routing Rule for the when a match cannot be found. In this case, we'll route these calls to the operator:

Phrase 0104 announces that "The order ID could not be found, please hold for the operator." Since there are no other rules created at this point, it's a good time to test. An inbound call is placed to the AA. Option 1 is selected to enter an order number. An order number of "1234567" is entered. Notice that at this point, even though the order number "1234567" does exist in the database, that without a rule to process it, we still reach the operator via the default routing rule:

The ACR log window shows this call, along with the collected data confirming that the data was correctly transmitted to ACR. At this point, rules can be entered to match the data collected from the auto attendant to the database. The important thing to remember here is that the call destination is not retrieved from the database (i.e. ACR does not retrieve actual extension numbers to be used as a calling target). Instead a rule is added for each agent. To create the rules click on Operation => Routing Rule in the ACR window, then click on Add to create a new rule.

The rule should be named for the Agent, and will need to be established based on the "IVR Data". The first rule will be called "Orders_Alice". Matches to this rule have Alice's extension number (1002) set as the target. Check the box next to "IVR Data" and then right click on IVR data and select "From User Database". Use the previously configured DSN with an account that has rights to run SELECT queries against the SQL database.

The "Tag Name" for the rule needs to match the "Text Tag" that is assigned in the auto attendant where these digits are gathered. This is an int value in the db, and the column name in the database is 'order_id' in the 'order2agent' table. Left like this, the example call above would route any call that has a valid order number to Alice. To make sure that only calls assigned to Alice match this rule, an additional condition is added to the WHERE clause of "AND agent = 'Alice'". Thus, the SQL Query to be run is:

Select order2agent.order_id From order2agent Where order2agent.order_id=@order_id AND agent = 'Alice'

This will ensure that only calls for the orders that are assigned to Alice will get directed to her. A test call at this point will confirm that this is working. The same call example that was used to test the default rule previously can now be used to confirm that the record is located, and routed appropriately:

Once the call routing for this order number is confirmed to work, similar rules can be populated for each user. The only item that needs to be changed is the additional WHERE clause, and the target extension. For example the rule for Bob, needs to have a target extension of 1003 (Bob's extension), and the WHERE clause needs to be "AND agent = 'Bob'".

Finally, make test calls for an order number assigned to each agent to confirm that the rules are all correct. Any further entries that are added to the database, will now be accessible as data for call routing.

<https://know.altigen.com/questions/1065/>