

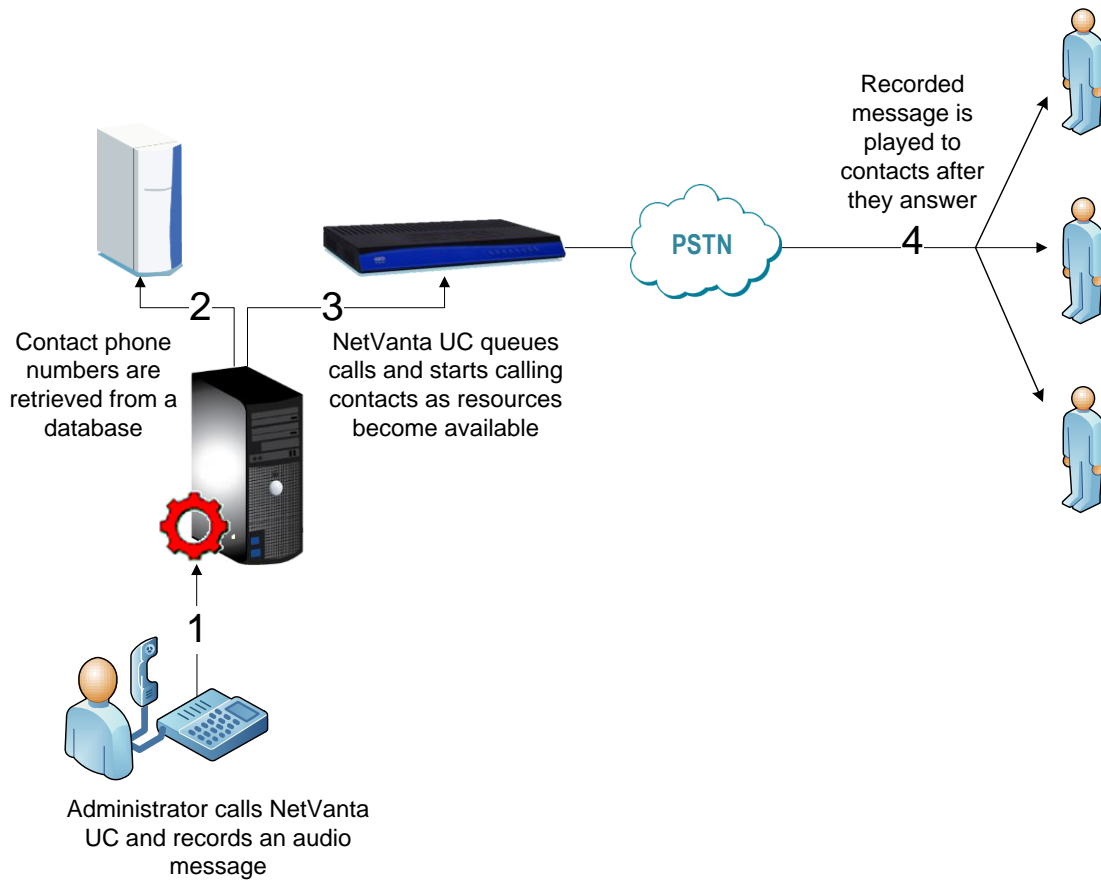


NetVanta Unified Communications Technical Note

Configuring Simple Outbound Call Campaigns with NetVanta UC server

Introduction

NetVanta UC server is capable of calling a group of contacts and delivering a common message to automate call campaigns typically conducted as a manual task. A simple outbound call campaign is one initiated by an administrator calling into an IVR service that first records an announcement (to play to the contacts being called in the campaign), then fetches the contacts to notify from a database, and then queues outbound calls to these contacts. When connected, the call to each contact will play the audio recorded by the administrator. The diagram below shows the high level call flow for a simple outbound call campaign.



Overview

This guide has the following sections:

- Requirements
- Limitations
- Assumptions
- User Data Table Configuration
- Service Creation
 - Service Played for Contacts
 - Outbound Call Campaign Kick-Off Service
- Service Testing

Requirements

Software

- Any NetVanta UC server product (BCS, ECS, UCS, or BAS) with version 4.5 and above
- Data Source Connector license
- Sufficient number of PBX Port and/or Application Channel licenses to handle the required number of simultaneous outbound calls (in addition to any other requirements for Application Server interactions like voicemail, auto attendants, IVR, handset playback, etc...)

Hardware

- Sufficient number of physical PSTN trunks (analog, PRI, and/or SIP) to handle the required number of simultaneous outbound calls (in addition to requirements for regular inbound and outbound call activity from/to the PSTN)

Limitations

This implementation style of this simple outbound call campaign does not support the following features:

- Graphical user interface for a desktop PC for manually triggering outbound calls and viewing/managing results
- Real-time monitoring of active outbound call campaigns
- Interrupting active outbound call campaigns
- Automatically starting and/or regularly scheduled outbound call campaigns
- Extracting detailed reports about the success or failure of outbound call campaigns¹

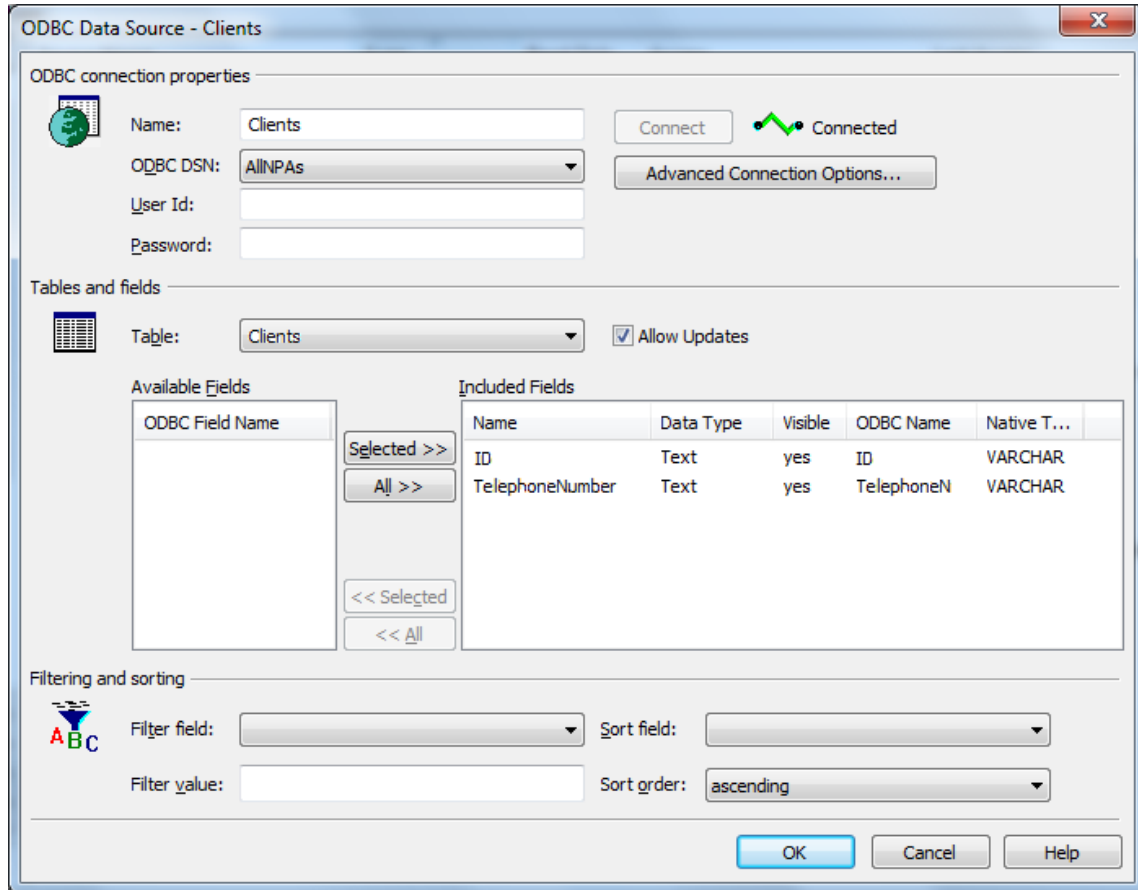
If any of the above features are required then you will need the UC Notification Server license (part number 1952NSGENG1). You will also need ADTRAN's Business Integration Services (BIS) to implement the service. Please contact Applications Engineering for a quote.

¹ Simple logs can be exported from the NetVanta UC Client administrator by using the "Import-Export Wizard" from the File menu and using the "Outbound Call Outcome" filter.

Assumptions

This document assumes the following items:

- The contacts to be notified are stored and managed in an external ODBC-compliant database
- The Windows *System DSN* has already been configured to point to this ODBC database
- The ODBC Source has already been created in NetVanta UC server with the following (minimum) properties:
 - Name: *Clients*
 - Field 1: *ID* (Data Type: *Text*; Native Type: *VARCHAR*)
 - Field 2: *TelephoneNumber* (Data Type: *Text*; Native Type: *VARCHAR*)



For more information about connecting NetVanta UC server to external ODBC-compliant databases:

1. Open the built-in help from the NetVanta UC client administration application
2. Navigate to the "Index" tab
3. Type "ODBC" for the keyword search then press Enter
4. Select "Managing Data Sources"

User Data Table Configuration

A “User Data Table” is a database that is exclusively managed inside the NetVanta UC administration client. User Data tables are most useful for storing small amounts of data that doesn’t generally increase in size and that doesn’t require the ability to extract data for reporting purposes. A user data table will be used for this example to store recorded audio that is played to contacts during the outbound call campaign. To create a user data table to store recorded audio, follow these steps:

1. Log into NetVanta UC client as a user with administrative privileges.
2. From the selection pane on the left, navigate to the **Data Sources** section.
3. Right-click in the contents pane and select **New > User Data...**
4. Provide a meaningful name like *RecordedAudio* (optionally adding a description and instructions)
5. Create two fields:
 - ID (data type: Text)
 - Audio (data type: Audio)

For example:

User Data Source - RecordedAudio

Data source properties

Name: RecordedAudio

Description: Stores audio used for outbound call campaigns

Instructions: Do not manually add records to this table; used only by the outbound call kick-off service.

"Add button" label: Add

Database fields

Seq	Name	Data Type	Decimals	Default Value	Display Width
1	ID	Text			20
2	Audio	Audio			20

Sorting and retrieval

Sort field: -- no sort --

Sort order: ascending

Maximum rows: 200

OK Cancel Help

Service Creation

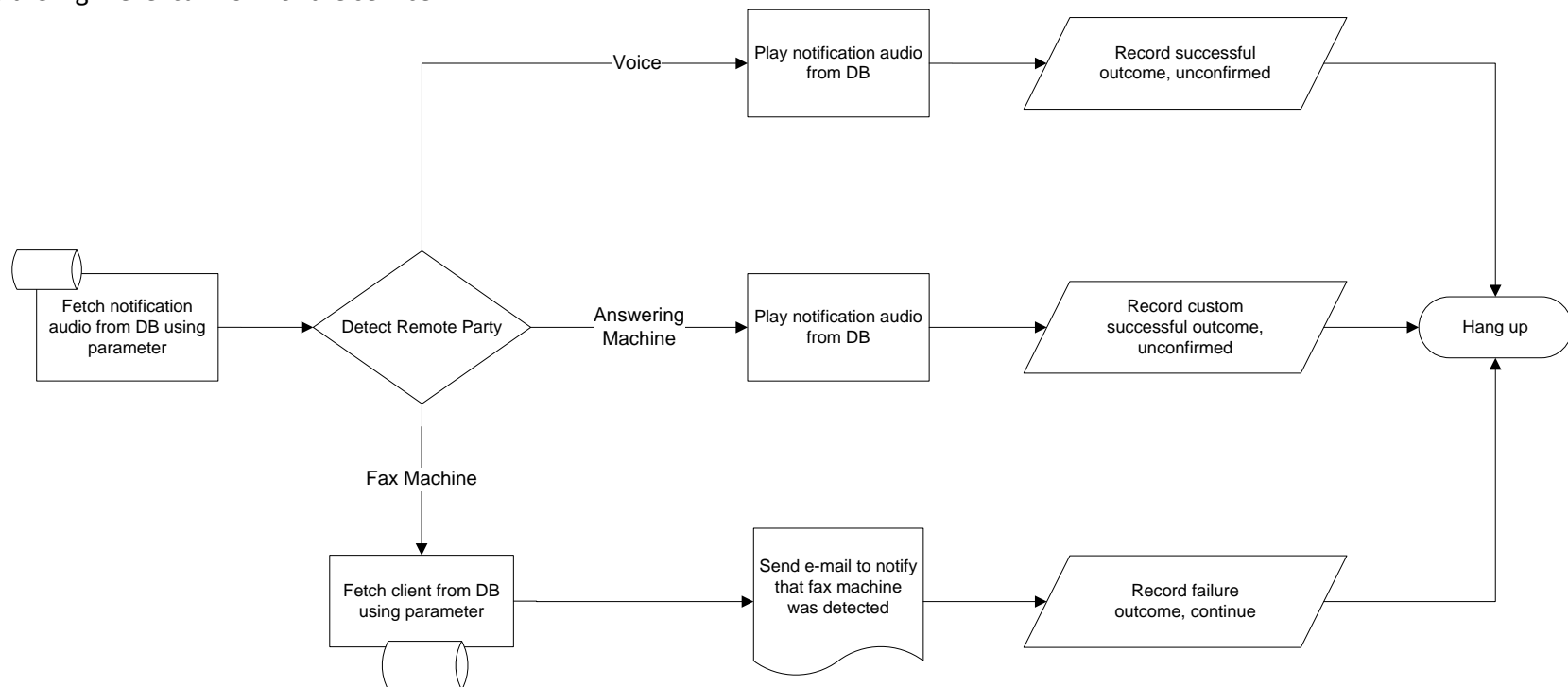
Two services must be created for an outbound call campaign: one that the administrator calls to start the outbound call campaign and another that is used to play the recorded audio message to the contacts that are called as part of the campaign. We will start with the service that is executed once a call from the outbound call campaign is answered.

Service Played for Contacts

This fundamental purpose of this service is to play an audio message recorded by the administrator that started the outbound call campaign. The service first attempts to detect what the remote party is – i.e. human voice, answering machine, or fax machine – then does something different based on the outcome. If a human voice or answering machine is detected it plays the audio and records the outcome citing the result for each. If it detects a fax machine then an e-mail is sent to the system administrator to indicate that a fax machine was detected and records the outcome accordingly.

The “Service Played for Contacts” service is somewhat unique as it is not associated with an identity as a typical service would be: instead, it is referenced in the *Queue Outdial* element in the “Kick-off” service (detailed in the next section).

Here is the high-level call flow for the service:



Before covering how to create the service we must first take a moment to explain the concept of a *parameter*.

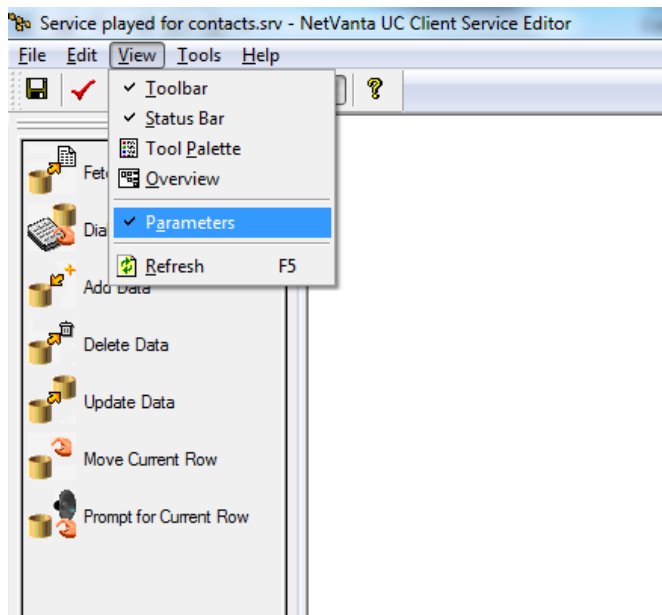
Parameters

The *Queue Outdial* element is used to place a call to one or more numbers and execute a specific service when that call is answered. This is a key element in the “Kick-off” service (detailed in the next section) since this element is what causes the system to place a call and execute a service once that call is answered. The *Queue Outdial* element includes an option to specify “parameters”: these are values that are passed to the service that is executed when that call is answered. Parameters can be phone numbers, a unique ID, the caller ID, etc... This allows the called service to have contextual information about the call without having to collect it from the called party. In this example, a parameter is used to pass the unique ID of the administrator-recorded audio message stored in a database – without this parameter the called service would not know which audio file to fetch from the database for that particular campaign. With the above in mind we will assume the following when designing this service:

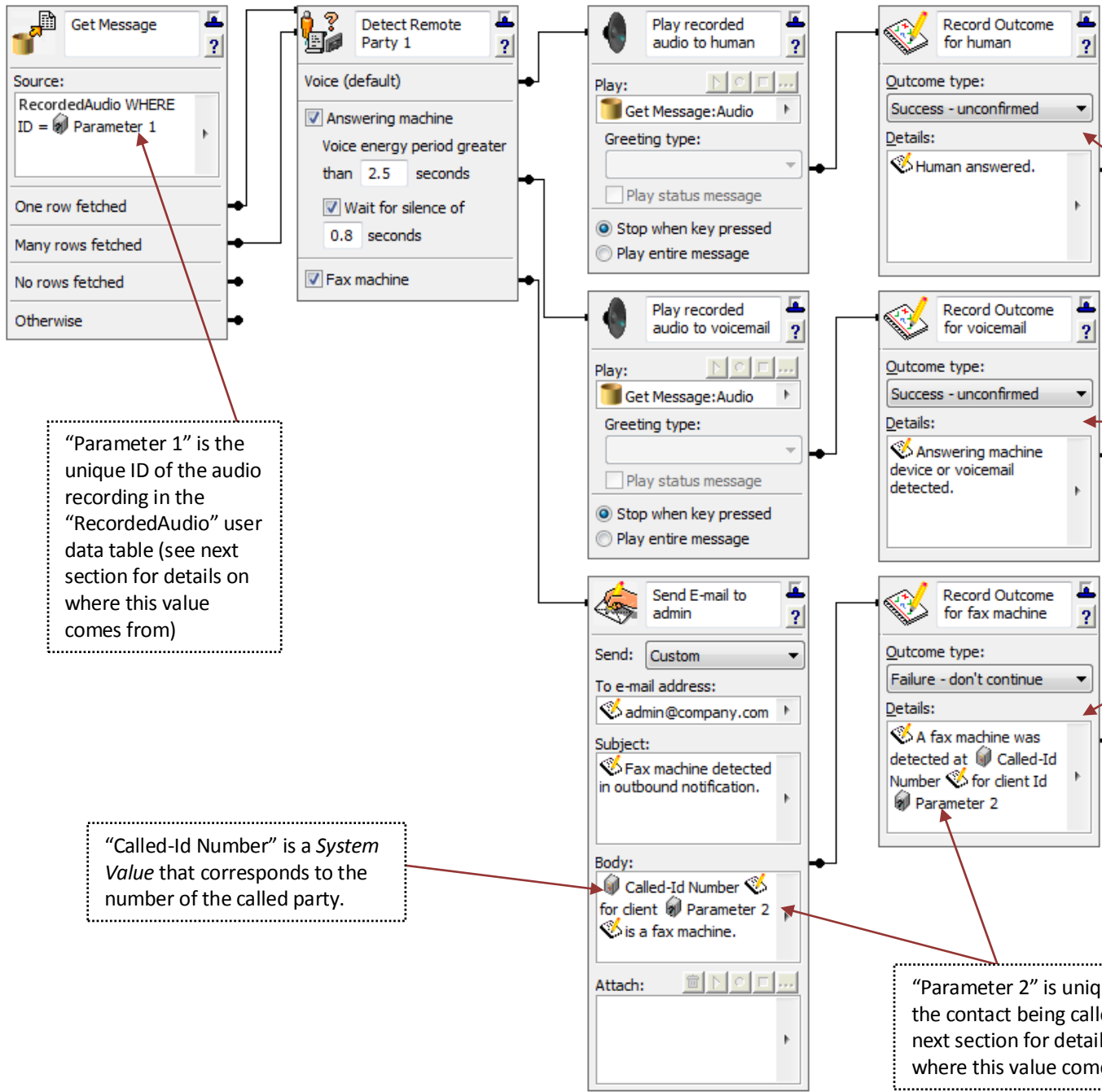
- Parameter 1 = unique identifier for the record in the *RecordedAudio* user data table
- Parameter 2 = unique identifier for the client/contact in the ODBC database

Note: you will see how these parameters are passed to this service in the following section.

To make parameter selection visible in the service editor you must first enable the option from the menu bar. To enable parameters in the service editor select **View > Parameters**.



The following page shows a screen shot of the “Service Played for Contacts”. Design your service to match the screen shot.



“Parameter 1” is the unique ID of the audio recording in the “RecordedAudio” user data table (see next section for details on where this value comes from)

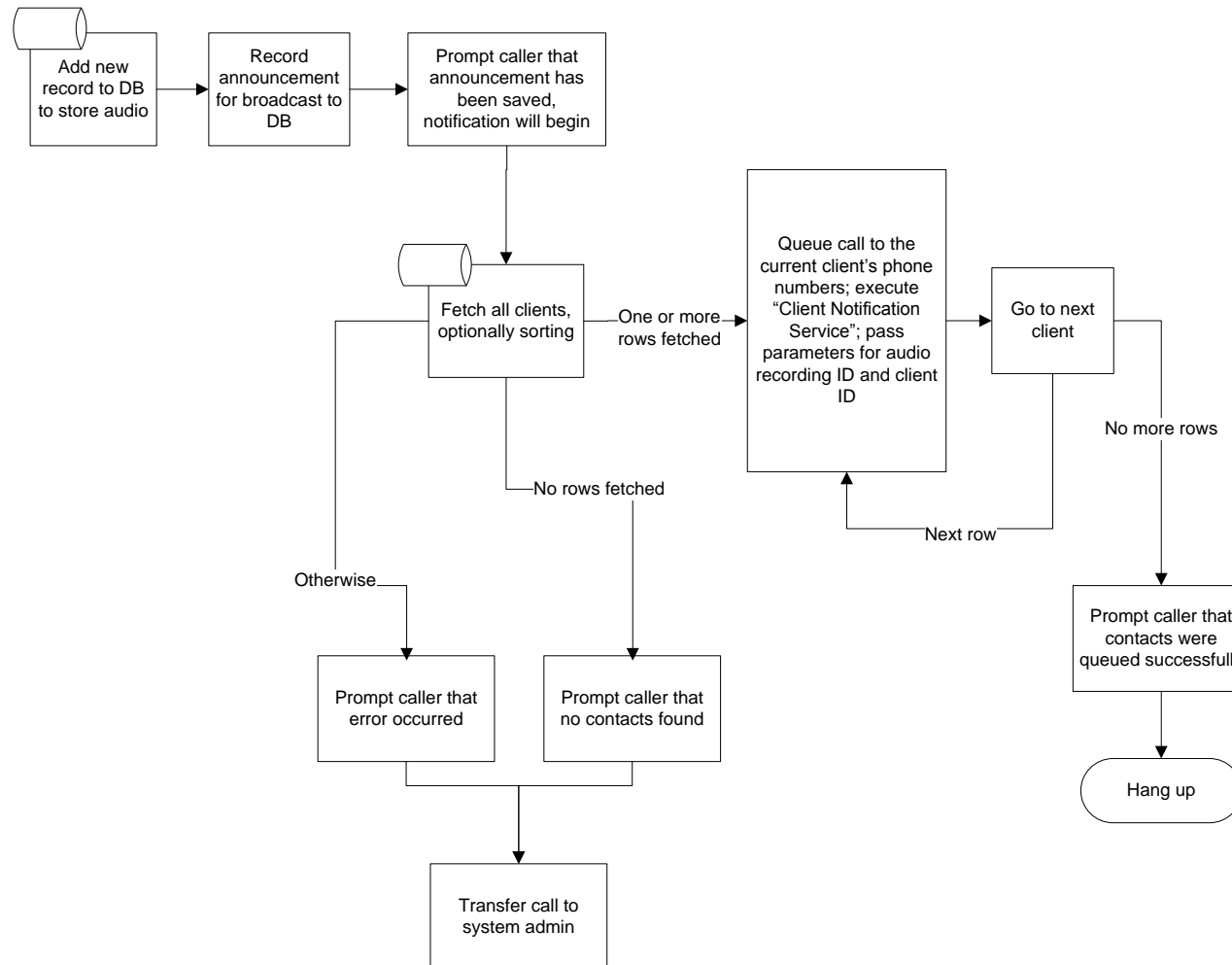
“Called-Id Number” is a *System Value* that corresponds to the number of the called party.

“Parameter 2” is unique ID for the contact being called (see next section for details on where this value comes from)

The *Record Outcome* element instructs UC server on how to proceed with this contact. A result with “continue” indicates that it should try the next phone number for the same contact if supplied (see “Kick-off” service *Queue Outdial* element). In this example, each contact only has one phone number and these outcomes instruct UC server to stop calling this contact and move to the next contact in the queue.

Outbound Call Campaign Kick-off Service

This service first records an audio message that's stored to a user data table then retrieves a set of contacts from the database and queues calls to the contacts to be notified. The "Kick-off" service is associated with an attendant identity that the administrator calls to start the outbound calls. This identity should be configured to be accessible from inside and outside the system (i.e. from an internal phone and/or from an external number), but this preference is ultimately up to the customer requirements. Here is the high-level call flow for this simple outbound call campaign:



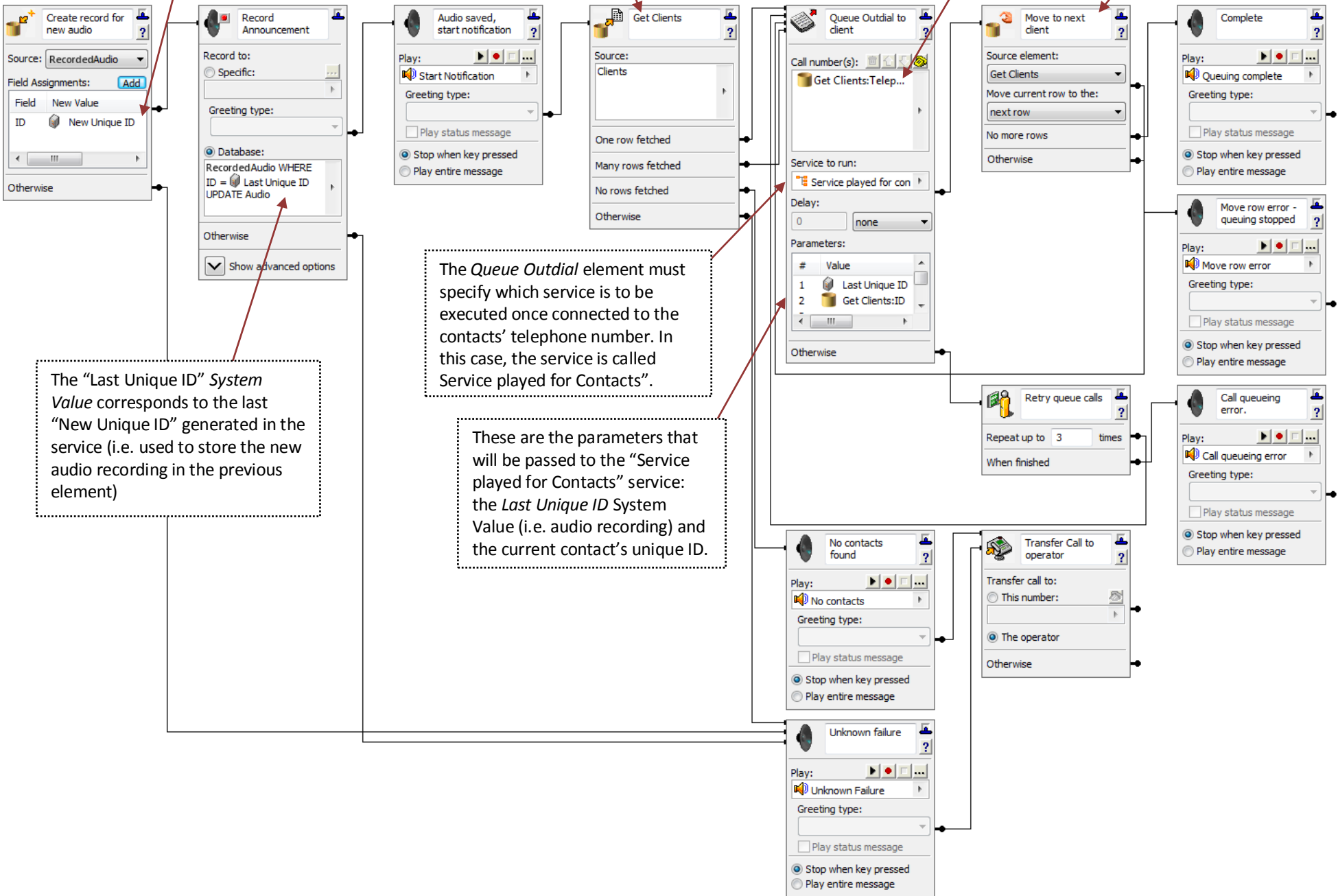
A screen shot of the corresponding UC service is shown on the following page.

The "New Unique ID" System Value is a globally unique ID used to identify the specific audio recording throughout this particular outbound call campaign.

This Fetch Data element references the Clients ODBC data table. No filter is specified here because all contacts are being fetched from the database.

In this case, each contact has one phone number at which they will be called (TelephoneNumber field). If UC server is to try subsequent numbers in the event of a "continue" result in the "Service Played for Contacts" then they can be individually added here.

Moves the iterator to the next record retrieved in the Fetch Data (aka "Get Clients") element.



The "Last Unique ID" System Value corresponds to the last "New Unique ID" generated in the service (i.e. used to store the new audio recording in the previous element)

The Queue Outdial element must specify which service is to be executed once connected to the contacts' telephone number. In this case, the service is called Service played for Contacts".

These are the parameters that will be passed to the "Service played for Contacts" service: the Last Unique ID System Value (i.e. audio recording) and the current contact's unique ID.

Service Testing

The easiest way to test the services is to use test contact data for the list of contacts to notify. This can be done by either changing the ODBC Source in NetVanta UC Client to point to a different underlying data table in the ODBC database (while keeping the same Data Source name in UC Client), or by changing the underlying data in the ODBC database to use test contact data. The test contact data should be phone numbers that are answerable by you (e.g. one or more internal phones or cellular phones). Associate the “Kick-off” service with an attendant identity and then call the service to initiate the outbound call campaign. Allow the call to be placed to the test phone number(s) and answer the call(s), ensuring that the recorded audio is played properly. Next, examine the NetVanta UC logs in the administrator Logs pane and ensure that the outcome for the call is recorded properly.