


Oct 21, 2020 19:01

Delayed Paging

 This document explains how to set up Delayed Paging – allowing users to record the message and then have it delivered as a page.

Created: January 2019

Permalink: <https://confluence.wildix.com/x/GgAJAg>

Basic information on Paging feature: [Paging - Admin Guide](#)

- [Use Case](#)
- [Limitation](#)
- [Configuration](#)
 - [Step 1. Set up an entry in the "users" dialplan to be used for this new "Delayed Paging"](#)
 - [Step 2. Add "forwarding users" 444 and 555](#)
 - [Step 3. Build your paging group](#)
 - [Step 4. Add "users" dialplan entries for digits 445 and 556](#)
 - [Workaround: Using SIP invite for paging more than 15 users](#)
 - [Step 5. Create a separate dialplan procedure 'mypaging' with two entries](#)
- [How it works](#)
- [Additional Information](#)

Use Case

Customers would like a paging solution that allows them to record a message, then have this message delivered as a page. If they do not like the recording that they have recorded, they would like an opportunity to record the message over again before it is sent out.

Paging on the Wildix system is by default real time, meaning that when a person is paging a page group the paged parties hear what the paging party is saying in real time. There are cases where it would be preferred to record a message and have this recorded message delivered as a page. Specifically:

1. If the latency in the environment is enough to be noticeable on an overhead paging system. Example, the paging party can hear the page being delivered slightly after they are saying it. This can be distracting to the paging party.
2. If the paging party wants an opportunity to correct their paging message. Example, I am sending a page and I say "the work facility will close 2pm today due to weather conditions. Sorry, correction ... 3pm the facility will close". In such a case, the paging party would like to start the message over rather than correcting it in real time.

Limitation

This configuration currently ONLY works for invite based paging. It does not work for multicast paging. This means that it is useful for cloud PBX installations that do not utilize any local multicast paging solution such as a local HW PBX with WMS Network configured.

⚠ Note: Work around for this limitation is described in Step 5 "Create a separate Dialplan procedure 'mypaging' with two entries".

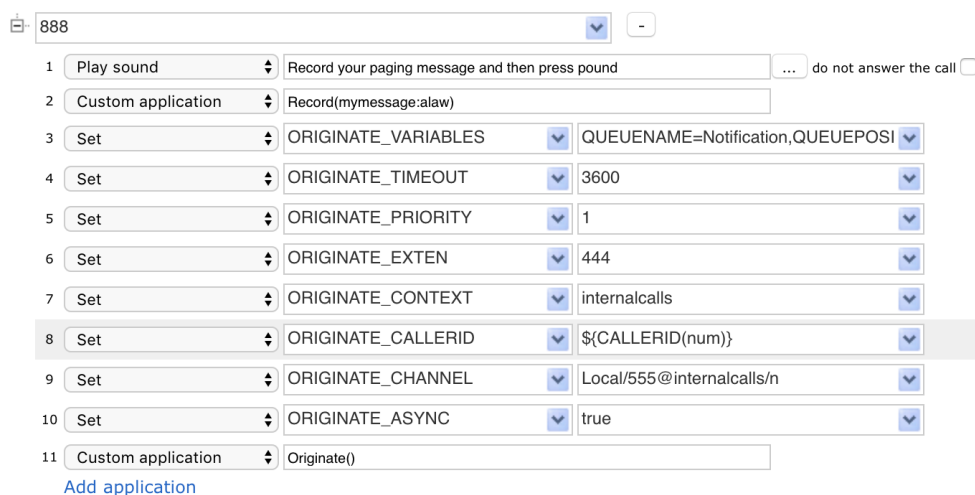
Configuration

Step 1. Set up an entry in the "users" dialplan to be used for this new "Delayed Paging"

Notice in the screenshot below that there we are using 888 for the number to be dialed by the user to perform the delayed paging. The applications in the dialplan essentially prompt the caller using TTS to "record your page message and then press pound", then set variables required to originate the page when done and ultimately originate a call to the paging system.

⚠ Note: Originate Extension is set to 444. This 444 is a "fake user" in the system exclusively used for call forwarding. This user does not need to be online. It is simply used to forward the call to a dialplan entry that will playback the recorded announcement to the paging group.

⚠ Note: Originate Channel is set with 555 as being the user part. This user 555 is another "fake user" which forwards the call to a dialplan entry that will route to the paging group. Again, it does not need to be an online user, it is simply used for call forwarding to the paging group.



Step	Application	Parameters
1	Play sound	Record your paging message and then press pound ... do not answer the call <input type="checkbox"/>
2	Custom application	Record(mymessage:alaw)
3	Set	ORIGINATE_VARIABLES: QUEUENAME=Notification, QUEUEPOSITION
4	Set	ORIGINATE_TIMEOUT: 3600
5	Set	ORIGINATE_PRIORITY: 1
6	Set	ORIGINATE_EXTEN: 444
7	Set	ORIGINATE_CONTEXT: internalcalls
8	Set	ORIGINATE_CALLERID: \${CALLERID(num)}
9	Set	ORIGINATE_CHANNEL: Local/555@internalcalls/n
10	Set	ORIGINATE_ASYNC: true
11	Custom application	Originate()

[Add application](#)

Below are the applications used in the corresponding order:

- Set -> ORIGINATE_VARIABLES -> QUEUENAME=Notification, QUEUEPOSITION=1
- Set -> ORIGINATE_TIMEOUT -> 3600
- Set -> ORIGINATE_PRIORITY -> 1
- Set -> ORIGINATE_EXTEN -> 444
- Set -> ORIGINATE_CONTEXT -> internalcalls
- Set -> ORIGINATE_CALLERID -> \${CALLERID(num)}
- Set -> ORIGINATE_CHANNEL -> Local/555@internalcalls/n

- Set -> ORIGINATE_ASYNC -> true
- Custom application -> Originate()

Step 2. Add “forwarding users” 444 and 555

Preferences for user 444:

☰ **Features**

Classes	internal ▾
Reject all calls	<input type="checkbox"/>
Call Forward Busy	<input checked="" type="checkbox"/> VOICEMAIL
Call Forward No Answer	<input checked="" type="checkbox"/> VOICEMAIL
Call Forward All	<input checked="" type="checkbox"/> 445
Call waiting	<input checked="" type="checkbox"/>
Call timeout	<input type="checkbox"/>
Mobility extension management	<input type="checkbox"/>
Activate mobility extension for Mobile [X]	10
Mobility confirmation	<input type="checkbox"/>
Notify missed calls via email via sms	<input checked="" type="checkbox"/>
Custom Ring	<input checked="" type="checkbox"/> Ring1 ...

Note: Call forward all is selected and sends all calls destined to this user record to 445.

Preferences for user 555:

☰ **Features**

Classes	internal ▾
Reject all calls	<input type="checkbox"/>
Call Forward Busy	<input checked="" type="checkbox"/> VOICEMAIL
Call Forward No Answer	<input checked="" type="checkbox"/> VOICEMAIL
Call Forward All	<input checked="" type="checkbox"/> 556
Call waiting	<input checked="" type="checkbox"/>
Call timeout	<input type="checkbox"/>
Mobility extension management	<input type="checkbox"/>
Activate mobility extension for Mobile [X]	10
Mobility confirmation	<input type="checkbox"/>
Notify missed calls via email via sms	<input checked="" type="checkbox"/>
Custom Ring	<input checked="" type="checkbox"/> Ring1 ...

Note: Call forward all is selected and sends all calls destined to this user record to 556.

Step 3. Build your paging group

Build a standard paging group that you want to send the page to. In this example, we built a paging group called “TestPage”. See 556 entry Step 4 below. More information on how to create a Paging group: [Paging - Admin Guide](#).

Step 4. Add "users" dialplan entries for digits 445 and 556

445

1 Play sound mymessage ... do not answer the call

[Add application](#)

556

1 Paging group TestPage

[Add application](#)

Workaround: Using SIP invite for paging more than 15 users

Replace Step 4.

556

1 Custom application Page(Local/502@mypaging&Local/503@mypaging)

[Add application](#)

- Custom application -> Page(Local/<user>@mypaging&Local/<user>@mypaging)

Eg. Page(Local/502@mypaging&Local/503@mypaging)

Note: In brackets you need to specify all users to be paged using '&' as a delimiter.

Step 5. Create a separate dialplan procedure 'mypaging' with two entries

Edit mypaging

Description: MulticastRTP work around

X.

1 Jump to if Condition "\${DEVICE_STATE(SIP/\${EXTEN})}"="INUSE" Procedure mypaging (Hacked) Number hangup Set

2 Custom application SipAddHeader("Call-Info: sip:wildixgw;answer-after=0;force")

3 Custom application Page(SIP/\${EXTEN})

[Add application](#)

hangup

1 Hangup Reason -

[Add application](#)

[Add number](#)

Included procedures (analyzed after the content of this procedure)

Applications used for X.

- Jump to if -> "\${DEVICE_STATE(SIP/\${EXTEN})}"="INUSE" -> Procedure mypaging -> Number hangup
- Custom application -> SipAddHeader("Call-Info: sip:wildixgw;answer-after=0;force")
- Custom application -> Page(SIP/\${EXTEN})

Applications for hangup

- Hangup -> Reason

Side note: if you want paging to interrupt current call, disregard this custom app

- Jump to if -> "\${DEVICE_STATE(SIP/\${EXTEN})}"="INUSE" -> Procedure mypaging -> Number hangup

How it works

A user dials 888. They are prompted to record a message and press pound. If they do not like the message they recorded, they can simply hang up. In this case, the recording will not be delivered as a page. If they are satisfied with the recorded message they press #. The system then hangs up on this party and originates a new call. This call is originated TO the paging group and is connected to the application that plays back the recorded message.

Additional Information

In this example we used:

- Users 444 and 555
- Dialplan entries for call forwards of 445 and 556

These are arbitrary values and can be changed to anything that you would like. However, if you do use different values, be sure to change ALL references for it to the value of your choice.

Example: If you want to have the user as 321 instead of 444, you must of course build a user 321 instead of 444, and you must change the reference to 444 in the 888 dialplan entry to 321 as well. This one! Set -> ORIGINATE_EXTEN -> 444

Also note that in this example we have added everything to the user dial-plan since the originating call is expected to come from an internal user dialing 888. However, this example could be tweaked to use for some other purpose that perhaps has a triggering event of a call arriving on a trunk. If you reuse this example in such a way you would want to be sure to add the originate application steps into the dialplan associated with the trunk that the call will be arriving on. For that matter, if a user that is to dial 888 is NOT assigned the user's dialplan but rather some other dialplan, the configuration must be added to the dialplan that this user is assigned.

In our example we have created a recording called mymessage. It is possible that if multiple people are using delayed paging at the same time that one user's message would overwrite another user's recording. This is of course highly unlikely in a normal environment but in such a case you could setup different paging codes using different filenames to avoid such an edge case.