Streaming on idle extension

OPERATIONAL DIRECTIONS

Mitel®

NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

TRADEMARKS

1 GENERAL

The feature Streaming On Idle Extension is specific to MITEL terminals 6800 and later models. It allows the user to connect to a music (or other) media stream without any line being occupied. The phone will be opened to inward and outward dialing, and when returning to idle state after a call, the music can be retrieved automatically or with a single button push.

1.1 GLOSSARY AND ACRONYMS

For a complete list of abbreviations and glossary, see the description for ACRONYMS, ABBREVIATIONS AND GLOSSARY.

2 PREREQUISITES

To run this feature a media server and program unit AUXMSP needs to be present. The existence of a media server is checked with command *media_gateway_info.* The existence of program unit AUXMSP is checked with command *pu_info*.

A media server can be added with commands *media_gateway_config*, *media_gateway_interface* and *media_server,* which shall have been done before executing these operational directions.

3 AIDS

I/O terminal

4 REFERENCES

These Operational Directions refer to the following documents:

Command description:	*(media_gateway_config)* *(media_gateway_info)* *(media_gateway_interface)* *(media_server)* ***streaming_data*** ***extension_key*** ***start,*** in the Technical Reference Guide, unix commands
Operational Directions:	--

5 PROCEDURE

-

6 EXECUTION

6.1 GENERAL

The Streaming-on-idle-extension function is realized by ordering SIP terminals to listen to an existing stream or to set up a stream, and then order terminals to listen to it. The stream is set up in a "Media Server".

The stream is either dynamic, set up at terminal button push, or static, terminal is ordered to listen to an already existing stream (normally multicast).

The streamed media connection will be as following, from a media source to a media server, and further to a terminal:

---- server input --->| **Media Server** |----server output ----> **(Terminal)**

6.1.1 DYNAMIC CHANNELS

A dynamic channel is a channel that is set up on the Media Server when ordered from a terminal, and torn down either when the message is finished, or when ordered from the same terminal.

The following parameters are used for dynamic channels.

--channel
The channel identity needs to be unique.

--execution-uri
This parameter is sent to the terminal. If the IP address is 0.0.0.0 it is replaced with the terminals actual IP address. It tell the terminal what address to listen to.

--media-server-address
The address of the media server to handle the request. This address will be matched as close as possible.

--media-server-input-uri
This parameter tells the media server where to get the file or stream from.

--media-server-output-uri
This parameter tells the media server where to send the output to. The port number in this parameter shall match the port in the execution-uri.

--description
The data in this parameter will show up on the terminal’s menu for selection. The displayed “name” of the channel shown in the phone menu.

--port-limit
This parameter is used to vary the port used for channels. A number between the port number in the execution-uri and the port number + --port-limit will be selected.

--volume
This parameter states the relative volume the stream will be connected with.

--synchronous
If this parameter is set to false the connection is established as a sunfan in the media server. This also means that the connection is done to the point where the message is currently playing.

If set to true a one to one connection is established and in case of a file it will be played from the beginning.

6.1.2 STATIC CHANNEL

A static channel is a channel which is intended to be active all the time, and serve many users.

To do this it will preferably be a multicast channel. This channel is then matched with dynamic channel that only contains the order to the terminal.

The following parameters are used for static channels.

--conference-id
The conference id needs to be unique.

--lim
This is the LIM or server (Service Node) to handle the communication with the media server.

--media-server-address
The address of the media server to handle the request. This address will be matched as close as possible.

--media-server-input-uri
This parameter tells the media server where to get the file or stream from.

--media-server-output-uri
This parameter tells the media server where to send the output to. Preferably this should be a multi-cast address.

6.2 CONFIGURING A DYNAMIC (UNICAST) CHANNEL

6.2.1 DYNAMIC CHANNEL FROM A FILE

In a dynamic channel there is one part for the terminal and one part for the media server.

Note: The file must be in wav mono format.

Set up a channel *dynamic_1* to send data from server at IP-address: 192.168.26.60/32.

Receive data from file: http://10.105.3.23/messages/test/message004.wav

Use port 60000 + random number (from 1 to 1000)

1. Set up uri to terminal, Note that since this is used by multiple terminals we use 0's instead of IP-address:

   streaming_data -i --channel dynamic_1 --description "Monty Python sketch" --execution-uri "RTPRx:0.0.0.0:60000" --port-limit 1000

2. Setup uri from where the media server is retrieving data:

streaming_data -c --channel dynamic_1 --media-server-input-uri http://10.105.3.23/messages/test/message004.wav

3. Setup where the output is sent from:

   streaming_data -c --channel dynamic_1 --media-server-address 192.168.26.60/32

4. Setup volume and synchronous mode. Since this file shall be played from beginning to end, it shall be played synchronously.

   streaming_data -c --channel dynamic_1 --volume -6 --synchronous true

If we now look at the channel, we now have the following parameter settings:

Streaming channel = dynamic_1

Description: Monty Python sketch

Execution URI: RTPRx:0.0.0.0:60000

Server Input URI: http://10.105.3.23/messages/test/message004.wav

Server address: 192.168.26.60

Codec: G.711 mu-law

Packet Size: 20 ms

Port limit: 1000

Volume: -6 dB

Synchronous: true

6.2.2 DYNAMIC CHANNEL FROM A MEDIA STREAM

In a dynamic channel there is one part for the terminal and one part for the media server. Note that the file must be in .wav mono format. For the cross coding capabilities see documentation for *MX-ONE Media Server*.

Set up a channel *dynamic_2* to send data from server at IP-address: 192.168.26.60/32.

Receive data from the stream: "http://134.25.4.151/p3-ogg-64"

Use port 60000 + random number (from 1 to 500)

1. Setup uri to terminal. Note that since this is used by multiple terminals we use 0's instead of IP-address.

   streaming_data -i --channel dynamic_2 --description "KSMC Radio" --execution-uri "RTPRx:0.0.0.0:60000” --port-limit 500

2. Setup uri from where the media server is retrieving data:

   streaming_data -c --channel dynamic_2 --media-server-input-uri "http://134.25.4.151/p3-ogg-64"

3. Setup where the output is sent from. The port must match the port in the execution-uri:.

   streaming_data -c --channel dynamic_2 --media-server-address 192.168.26.60/32

4. Setup volume and synchronous mode. Since this file shall be played from beginning to end, it shall be played synchronously.

   streaming_data -c --channel dynamic_2 --volume -6 --synchronous false

If we now look the channel we have the following parameter settings:

Streaming channel = dynamic_2

Description: KSMC Radio

Execution URI: RTPRx:0.0.0.0:60000

Server Input URI: http://134.25.4.151/p3-ogg-64

Server Address: 192.168.26.60/32

Codec: G.711 mu-law

Packet Size: 20ms

Port limit: 500

Volume: -6 dB

Synchronous: true

6.2.3 DYNAMIC “OFF/STOP” CHANNEL

It is recommended to initiate an “Off” or “Stop” entry in the menu for music/streaming on idle. This entry is configured similar to the other dynamic channels, but with specific settings.

Set up a channel *stop* to stop wav streaming and to release RTP resources.

1. Setup uri to terminal.

   streaming_data -i --channel stop --description "Off/Stop" --execution-uri "RTPRx:Stop;Wav.Stop:”

If we now look the channel we have the following parameter settings:

Streaming channel = stop

Description: Off/Stop

Execution URI: RTPRx:Stop;Wav.Stop:

Server Input URI:

Server Address:

Codec:

6.3 CONFIGURING A STATIC (MULTICAST) CHANNEL

For a multicast initiation we need two channels.

Set up one channel static_1 to send multicast data from http://134.24.5.151/p3-ogg-64 to rtp://232.0.0.2:60000

The server to use has address 192.168.26.60.

Note that the terminal uses "RTPMRx...." to determine that this is multicast, and the media server that the ip address that the output uri is within the multicast range 224.0.0.0 to 239.255.255.255.

Note: This type of channel does not have an execution-uri.

Set up everything in one command per channel:

1. **streaming_data -i** --conference-id static_1 -l 1 --media-server-address 192.168.26.60/32 --media-server-input-uri "http://134.24.5.151/p3-ogg-64"\ --media-server-output-uri "rtp://232.0.0.2:60000"
2. Set up a channel connect_to_static_1 for the terminal to connect to rtp://232.0.0.2:60000

   streaming_data -i --channel connect_to_static_1 --description "KSMC Radio" --execution-uri "RTPMRx:232.0.0.2:60000" --synchronous false --volume -6

   Note: Since the channel (or conference) static_1 is already playing, the channel, connect_to_static_1 must have an exact port. Thus the parameter --port-limit is omitted.
3. Execute start-phase 1,5 to activate channel static_1.

   start --system

If we now look at the two channels, we have the following parameter settings:

Static conference ID = static_1

Lim : 1

Server Address : 192.168.26.60

Server Input URI : http://134.24.5.151/p3-ogg-64

Server Output URI: rtp://232.0.0.3:60000

Streaming channel = connect_to_static_1

Description: KSMC Radio

Execution URI: RTPMRx:232.0.0.2:60000

Server Input URI:

Server Output URI:

Codec: G711u

Packet Size: 20ms

Port limit:

Volume: -6 dB

Synchronous: false

6.3.1 STATIC “OFF/STOP” CHANNEL

It is recommended to initiate an “Off” or “Stop” entry in the menu for music/streaming on idle. This entry is configured similar to the other static channels, but with specific settings.

Set up a channel *stop* to stop wav streaming and to release RTP resources.

1. Setup uri to terminal.

   streaming_data -i --channel stop --description "Off/Stop" --execution-uri "RTPRx:Stop;Wav.Stop:”

If we now look the channel we have the following parameter settings:

Streaming channel = stop

Description: Off/Stop

Execution URI: RTPRx:Stop;Wav.Stop:

Server Input URI:

Server Address:

Codec:

6.4 SETTING UP A STREAMING KEY ON A MITEL 6800/6900 SIP TERMINAL

The SIP terminals must have a dedicated key for the function streaming on idle. For this we will use key function XML. The XML string shall be for the terminal to activate the streaming menu.

'http://$$PROXYURL$$:22222/StreamingMenu?user=$$SIPUSERNAME$$'

For a detailed explanation, see XML guide for terminals.

Example: The key shall be key number 2, and the display shall show "Radio"

Set this up with the following command.

1. **extension_key -i** --dir 1856811102 --key 2 --function XML --display-text Radio \
   --xml-on-demand-uri \
   'http://$$PROXYURL$$:22222/StreamingMenu?user=$$SIPUSERNAME$$'

7 TERMINATION

Dump to backup media is to be executed if exchange data have been changed and no further commands are to be entered.