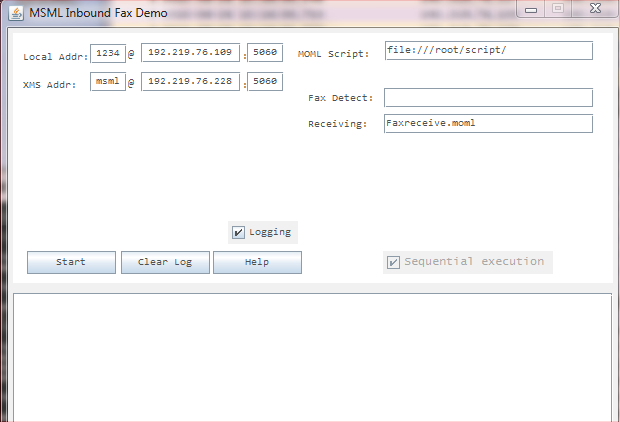
Introduction:

To simplify implementation and troubleshooting, this sample only handles MSML inbound fax. This demo is initially designed for interactive application as a quick test tool for voice application. With this design, the MSML script is read from an http server or from XMS server. The application remains unchanged while the MOML script file can be modified via a text editor for a quick test.

How to use:

To use this app, run “java -jar msml\_GUI\_fax.jar”. The windows app will appear. Enter the local machine IP address and XMS server IP address. MOML file location can be in an HTTP server as in the screen or in the XMS server <file:///root/>, if the MOML file is in /root/ folder. Click Start and it is ready to answer inbound faxes. MOML files can’t be changed once Start is clicked. 

Script to receive a fax:

This script is to save the fax in XMS <file://rxFax.tif> which is /var/lib/xms/media/en-US/rxFax.tif, if the XMS locale is set to “en-US”. The remaining elements <faxstart> … <faxopcomplete> are the events this application wants to receive from XMS. When the initial invite to XMS is answered, this app sends this script to XMS to start receiving fax. XMS sends re-invite for T38 media if T38 is enabled.

The initial invite to XMS needs to have Accept-Contact header set to \*;+sip.fax=”t38”. This is to make sure XMS reserves the fax UDP port (SR140), otherwise the voice UDP (HMP) will be reserved.

Because there is no interaction needed, this demo only sends this script to start fax receiving. If user prefers to wait for CNG tone before start receiving fax, user can place a faxdetect script in “Fax Detect” field detect CNG tone first. Fax tone detection is optional.

<?xml version="1.0" encoding="UTF-8"?>

<moml version="1.0" id="Faxrcv" name="FaxReceive">

<faxrcv id="fax\_id" lclid="local\_id">

<rcvobj objuri="file://rxFax.tif" maxpages="10"/>

<faxstart/>

<faxnegotiate/>

<faxpagedone/>

<faxobjectdone/>

<faxopcomplete/>

</faxrcv>

</moml>

Script to detect CNG:

Place this script in “Fax Detect” field. The demo will send this script to XMS as soon as XMS answers the initial invite. Once XMS detects CNG tone and reports to the app, the app will send next script to start receiving fax.

<?xml version="1.0" encoding="UTF-8"?>

<moml version="1.0" id="VoiceFaxdetect" name="CngDetect">

<faxdetect>

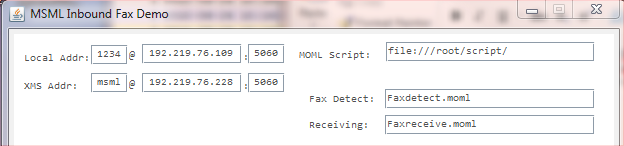
<faxdetectexit>

<exit namelist="faxdetect.tone faxdetect.end"/>

</faxdetectexit>

</faxdetect>

</moml>



Supported features:

1. T.38 or G711 fax passthru
2. With or without CNG tone present
3. Session timer refreshed by XMS

SIP call flow:

\* To simplify the implementation and troubleshooting, this app only handles inbound fax

\* Call flow:

\* Sr140 (fax over IP T38 device) -> this AS -> XMS

\* SIP flow:

\* Sr140 AS XMS

\* ->invite(sdp)

\* ->invite(sdp)

\* <-OK(sdp)

\* <-OK(sdp)

\* ->ACK

\* ->ACK

\* ->INFO(CNG detect, optinal)

\* <-INFO(CNG detected)

\* ->INFO(faxreceive)

\* <-invite(t38)

\* <-invite(t38)

\* ->OK(t38) or 488(for g711 fax passthru)

\* ->OK(t38) or 488 (g711)

\* <-ack

\* <-ack

\* <---T38 start between Sr140 and XMS--->

\* <-INFO(fax.negotiate event)

\* <-INFO(fax.pagedone, etc)

\* <---T38 end--------------------------->

\* <-INFO(fax.opcomplete)

\* <-INFO(msml.dialog.exit)

\* ->BYE

\* <-BYE

Implementation:

Controller.java – This handles the SIP messages.

GUI.java – This implements the graphical user interface.

MediaState.java – This enum class defines the call state and its moml file.

SipCall.java – This class behaves as book keeping for the pair of the call legs.