

ADOBE® FLASH® MEDIA GATEWAY 2.0

INSTALLATION AND CONFIGURATION GUIDE

© 2010 Adobe Systems Incorporated. All rights reserved.

Adobe® Flash® Media Gateway Installation and Configuration Guide for Windows®.

This user guide is protected under copyright law, furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

This user guide is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the user guide for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the user guide; and (2) any reuse or distribution of the user guide contains a notice that use of the user guide is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Adobe, the Adobe logo, and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Intel and Pentium 4 are registered trademarks of Intel Corporation in the U.S. and other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

Updated Information/Additional Third Party Code Information available at <http://www.adobe.com/go/thirdparty>.

Portions include software under the following terms:

This product contains either BSAFE and/or TIPEM software by RSA Security Inc.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by the JDOM Project (<http://www.jdom.org/>).

This product includes software developed by The Werken Company.

This product includes software developed by the IronSmith Project (<http://www.ironsmith.org/>).

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Acronyms	1
Chapter 1: Installing the server	
System specifications	2
Checking the system requirements	2
Installing FMG	2
Chapter 2: Configuring FMG Files	
Modifying the configuration file	4
	5
Configuring SIP for FMG	5
Setting codecs in SIP	7
Setting user profiles in SIP	7
Setting the default profile in SIP	8
Setting an account on SIP clients	10
Configuring RTMP for FMG	10
Configuring speex	14
Configuring the workflow	15
Configuring the HTTP interface	17

Acronyms

FMG	Flash Media Gateway
SIP	Session Initiation Protocol
RTP	Real Time Protocol
RTMP	Real Time Messaging Protocol
DTMF	Dual-tone Multi-Frequency
FMS	Flash Media Server
UAC	User Agent Client
UAS	User Agent Server

Chapter 1: Installing the server

Adobe® Flash® Media Gateway (FMG) is an application server that communicates with Adobe® Flash® Media Server (FMS) on one side and SIP on the other, to provide seamless telephony service to Flash users and SIP users. To install FMG, review the system requirements and run the installer.

System specifications

Supported operating systems

Microsoft® Windows® Server® 2003 32-bit Enterprise Edition, SP2

Hardware requirements

- 2 GB of RAM (4 GB recommended)
- Intel® Core 2 Duo 2.2 GHz processor (Intel® Xeon 3 GHz recommended)

Checking the system requirements

If you're not sure if FMG can run on your computer, review the complete system requirements and recommendations for your Adobe software. See the *ReadMe* file included with the installation file.

The additional software dependencies are as follows.

- FMG requires FMS 3.0 or higher to provide telephony services to Flash clients. You can obtain a free, development version of FMS from <http://www.adobe.com/products/flashmediaserver>.
- (Optional) FMG requires a SIP-compliant client to make calls to/from SIP phones within the local network.

Note: FMG has been tested with X-lite® and Zoiper® SIP clients.

- (Optional) You require a user account at an external SIP gateway service to make/receive calls to/from third-party SIP accounts, including calls to mobile and PSTN networks.

Note: FMG 1.1 has been tested with Asterisk® 1.4 and 1.6 releases.

Installing FMG

- 1 Double-click the installation file, FMGSetup.exe, and follow the prompts in the installation wizard.
- 2 Read and accept the End User License Agreement to continue the installation process.
- 3 Enter a serial number.

Note: If you don't enter a serial number, the development version of FMG, which allows a maximum of 10 calls, is installed.

- 4 Enter the desired username and password to be used by the administrator for the HTTP interface.

5 Accept the default installation location or click Change to enter a new location.

6 Select/deselect the following options:

- Start FMG immediately after installation
- Start FMG after each system startup

By default, these options are selected.

7 Review your installation choices. Click Back to make any changes, or Next to continue the installation.

The wizard displays the installation status. When the installation is complete, it displays the message, “InstallShield Wizard Completed”.

8 Click Finish to exit the wizard.

Important: Do not delete or modify the files in the `_defaultRoot_` folder in the configuration directory. These files are required for the working of the HTTP interface.

Chapter 2: Configuring FMG Files

To configure FMG to work with FMS and SIP, modify the settings in the following files.

Configuration file name	Description
fmsg.xml	Main configuration file for setting top level configurable elements such as debug level and max call legs.
sip.xml	SIP configuration file for setting up SIP stack parameters and SIP clients/gateways/proxies.
rtmp.xml	RTMP configuration file for setting up Leg Service and Control Service with FMS.
workflow.xml	Workflow definition file.
speex.xml	Configuration file to control the speex codec settings.
http.xml	HTTP interface settings file to modify elements such as admin username and hosts.
fmg.ini	File containing variables names to be used in any of the above XML configuration files.

Modifying the configuration file

The fmsg.xml is the top level configuration file for certain system-wide configurations.

Note: Any change to fmsg.xml requires a restart of the FMG service or reload through the control interface or the HTTP interface.

1 Locate the code similar to the following in the fmsg.xml file:

```
<FMSGConfig>
  <Config>
    <!-- maximum number of call legs allowed -->
    <maxCallLegs>400</maxCallLegs>
    <!--
      Log Levels are (in increasing order of severity)
      DATA (dumps send/recv timestamps for Audio/Video data, not recommended in
production env)
      DEBUG
      INFO
      WARNING
      ERROR
    -->
    <logLevel>8</logLevel>
    <MaxCutoffCPUUsage>80</MaxCutoffCPUUsage>
    <maxHoldTime>0</maxHoldTime>
  </Config>
</FMSGConfig>
```

2 Edit the following elements.

Element	Description
maxCallLegs	Specifies the maximum number of call legs that are allowed at FMG. The default is 1000. A value more than 250 isn't recommended.
logLevel	Identifies the severity for which logs are written to the log file. All logs of severity equal to or higher than this log level are written to the log file.
MaxCutoffCPUUsage	Specifies the maximum CPU % beyond which the status of the FMG changes to BUSY. Note: Valid CPU usage value is between 30% and 80%. Memory usage above 90% also places FMG in the BUSY state; this isn't configurable.
maxHoldTime	Specifies the maximum duration that a SIP call can be put on HOLD if a bridged RTMP call breaks due to an FMS crash. The call isn't to be put on HOLD in any of the following cases: <ul style="list-style-type: none">• if the element isn't present• when no value is set for it• when the value is "0"

Configuring SIP for FMG

The sip.xml file defines the settings related to the SIP stack in FMG.

Note: Any change to sip.xml requires a restart of the FMG service or reload through the control interface or the HTTP interface.

1 Locate the code similar to the following in the sip.xml file:

```
<FMSMGConfig>
  <portUpperLimit> 6000 </portUpperLimit>
  <portLowerLimit> 5000 </portLowerLimit>
  <SipPort> 5060 </SipPort>
  <registrationExpiration> 200 </registrationExpiration>
  <messageQueueSize> 2000 </messageQueueSize>
  <processingThreads> 2 </processingThreads>
  <debugSIP> false </debugSIP>
  <pingTime> 0 </pingTime>
  <dtmfTime> 80 </dtmfTime>
  <EnableAutoProfile > true </EnableAutoProfile >
  <DefaultContext > someContext </DefaultContext >
</FMSMGConfig>
```


2 Edit the following parameters.

Parameter	Description
portUpperLimit	The upper limit of the RTP port range. The default value is 11000.
portLowerLimit	The lower limit of the RTP port range. The default value is 10000.
SipPort	The port at which FMG is listening for SIP requests. The default port is 5060.
registrationExpiration	The refresh time when the FMG refreshes the registration with the remote server. The default value is 120 seconds.
processingThreads	The number of threads the SIP plugin invokes to process the SIP network events. The default value is 0.
messageQueueSize	The maximum length of the processing queue. The processing queue holds events such as the network events. The ideal value is around four times the maximum simultaneous calls expected. The default value is 100.
debugSIP	If the value is set to "true", extra debugging information is logged in SipLog.txt, including the packet capture. This isn't recommended for the production environment; it's only for debugging purpose.
pingTime	The time interval, in seconds, after which the OPTIONS message is sent to determine the availability of each remote SIP Profile. The default value is 60 seconds. 0 disables ping. If a remote SIP entity was available at some time (as determined by an OK reply to OPTIONS), and becomes unavailable for more than 300 seconds (5 mins) (as determined by an OK reply to the timeout waiting OPTIONS), all ongoing call legs for that SIP profile are disconnected. This works on the assumption that the remote SIP entity is no longer reachable.
dtmfTime	The duration for which the DTMF digits are to be sent. The default is 80 milliseconds.
EnableAutoProfile	This creates a default profile to accept any incoming calls on SIP. This can accept boolean values, true/false. The default value is false. If an actual default profile is configured in sip.xml, the default profile takes precedence over this tag and the <DefaultContext> tag.
DefaultContext	If <EnableAutoProfile> is true, then it specifies the context in the workflow in which the default calls will be handled. The default value is "default".

See also

[Setting codecs in SIP](#)

[Setting user profiles in SIP](#)

[Setting the default profile in SIP](#)

[Setting an account on SIP clients](#)

Setting codecs in SIP

The FMG can be configured to use only the specified audio and video codecs from the SIP-side. Codec IDs must be provided in `sip.xml` for each codec that is to be included.

The allowed codecs are G711u (G711 U-Law) and speex for audio and H264 for video. DTMF is always supported.

Example Setting G711u, speex and H264codec

```
<CodecList>
  <codecID> G711u </codecID>
  <codecID> speex </codecID>
  <codecID> H264 </codecID>
</CodecList>
```

H264 codec can be optionally qualified using attributes *deviceId* and *qualityLevelId*. The values of these attributes are defined in the configuration file *videoqualitylevels.xml*. This file contains a set of tested video quality profiles. The following example uses *high* quality-level of VTC device Tandberg 990 MXP -

```
<CodecList>
  <codecID deviceId="Tandberg990MXP" qualityLevelId="high"> H264 </codecID>
</CodecList>
```

It is recommended that *videoqualitylevels.xml* should not be modified. If there is no *deviceId* that matches your SIP device, then one of the *qualityLevels* of *deviceId="GenericVtcDevice"* can be used. The parameters, used in the value of *fntp* attribute of *qualityLevel* tag inside *videoqualitylevels.xml*, are defined in RFC 3984. In case you want to devise a new *qualityLevel* for your SIP device, then try experimenting by setting the *fntp*-parameters to different values, as per RFC 3984 and observing the quality of video generated from the device. It is possible that a SIP device does not implement the RFC completely and might not respond to some of these parameters.

Setting user profiles in SIP

Profiles are the accounts of users on FMG or the account of FMG on some other SIP-server (such as the FMG itself or Asterisk).

The XML parameters for profiles are the following.

Parameter	Description
profileID	Unique profile name, which isn't repeated in <code>sip.xml</code> . This is a mandatory parameter.
userName	Account name that is used to make the calls. This also serves as the username for REGISTER/INVITE authentication. Defaults to "FMG" or the username of the original calling party, in the case of a bridged call.
password	Password of the corresponding account; used for the authentication of REGISTER/INVITE requests. If the password is empty, then authentication is skipped.
displayName	Caller ID to be displayed for SIP requests. If this parameter isn't specified, <code><username></code> is used as the displayName. Username of the original calling party is used for bridged calls and "FMG" is used for the other calls.

Parameter	Description
authUserName	Authentication user name to be used while registering with a SIP server. If this field is empty, the value of field <username> is used by FMG to authenticate
doRegister	Set to 1 if FMG has to register on a different SIP server with the specified profile. Set to 0 if another SIP client is to register on FMG. The default value of this parameter is 0.
remoteSipHost	When the value of <doRegister> tag is 1, it is the address of the SIP server to which this profile has to register. Otherwise it is the IP address or hostname of the SIP client. It can be '*', which denotes that the SIP client address is dynamically determined when it registers on FMG. An empty value also defaults to '*'. It is recommended that FQDN (complete hostname qualified with domain) or IP address of the remote SIP machine is specified for remoteSipHost. Some PBX softwares are known to incorrectly reject the call when only machine's hostname is specified.
sipDomain	<sipDomain> is the domain name of the SIP service provider when FMG acts as a User Agent Client. <sipDomain> should be specified if it is different from the outbound proxy, as specified in <remoteSipHost> tag.
sipHeaders	A list of SIP headers (name: value pairs) that is added to any SIP INVITE message sent for this profile. Individual headers can be specified inside the <header> tag.
context	The workflow/dial-plan with which the profile is to be associated. All incoming calls for a profile are handled per the definition of this context in workflow.xml. The default value for this parameter is "default".
globalAddress	This tag is used if FMG is behind a firewall, and the public IP to reach the SIP Gateways is statically bound to the external side of the firewall. FMG uses this static IP/domain inside the SIP packets.

Setting the default profile in SIP

The SIP side can be configured to handle anonymous calls that aren't bound to any specified user profile. Such calls will be handled according to the "default" value of the <profileID> parameter. If the default profile isn't specified, FMG handles these calls in the "default" context, and supports all specified codecs.

The codecs, listed under <supportedCodecs>, are used for any call made using that profile. If <supportedCodecs> doesn't exist inside a profile or it doesn't contain any <codecID>, then the global codec list, specified using <CodecList>, is used for that profile.

Example 1 To define a profile with the username 100 at address of 10.40.63.160 and port 12450, which acts as the SIP client for FMG:

```
<Profile>
  <profileID> abc </profileID>
  <userName> 100 </userName>
  <password> 100 </password>
  <displayName> TestProfile </displayName>
  <doRegister> 0 </doRegister>
  <remoteSipHost> 10.40.63.160:12450 </remoteSipHost>
  <supportedCodecs>
    <codecID> G711u </codecID>
    <codecID> speex </codecID>
    <codecID deviceId="TandbergEdge95MXP" qualityLevelId="medium"> H264
  </codecID>
```

```

        </supportedCodecs>
        <sipHeaders>
            <header> Min-SE:900 </header>
            <header> Supported:timer </header>
        </sipHeaders>
        <context> sipToSipCall </context>
    </Profile>

```

Example 2 To define a profile with the username 200 at address of 10.40.63.160 and port 12450, which acts as the SIP outbound proxy with the SIP domain as sipdomain.com:

```

<Profile>
    <profileID> abc </profileID>
    <userName> 200 </userName>
    <password> 200 </password>
    <displayName> TestProfile </displayName>
    <sipDomain> sipdomain.com </ sipDomain >
    <doRegister> 0 </doRegister>
    <remoteSipHost> 10.40.63.160:12450 </remoteSipHost>
    <supportedCodecs>
        <codecID> G711u </codecID>
        <codecID> speex </codecID>
    </supportedCodecs>
    <sipHeaders>
        <header> Min-SE:900 </header>
        <header> Supported:timer </header>
    </sipHeaders>
    <context> sipToSipCall </context>
</Profile>

```

Example 3 To define a profile with the username 200 at address of 10.40.62.22, in which FMG acts as the SIP-client to other SIP servers like Asterisk:

```

<Profile>
    <profileID> abd </profileID>
    <userName> 200 </userName>
    <password> 200 </password>
    <displayName> TestProfile </displayName>
    <remoteSipHost> 10.40.62.22 </remoteSipHost>
    <doRegister> 1 </doRegister>
    <supportedCodecs>
        <codecID> speex </codecID>
    </supportedCodecs>
    <context> sipToSipCall </context>
</Profile>

```

Example 4 To define a profile that accepts anonymous calls to FMG:

```

<Profile>
    <profileID> default </profileID>
    <supportedCodecs>
        <codecID> G711u </codecID>
        <codecID> speex </codecID>
        <codecID> H264 </codecID>
    </supportedCodecs>
    <context> sipToSipCall </context>
</Profile>

```

Setting an account on SIP clients

To set an account with SIP clients like X-Lite:

- 1 Right-click in X-lite and follow the “SIP Account Settings” option.
- 2 Click Add.
- 3 Enter the values for User Name, Display Name, Password, and Authorization Name. The Authorization Name and Password must match those specified in sip.xml.

Note: The values of the fields other than Authorization Name and Password needn't match those specified in sip.xml.

- 4 In Domain, specify the hostname or IP of FMG.
- 5 Per your settings in sip.xml, specify if X-lite must register.
- 6 Click OK.
- 7 Click Close.

Configuring RTMP for FMG

The rtmp.xml file is read when FMS starts. The RTMP configuration in FMG defines the way RTMP interacts with different FMSs. Hence, configuring RTMP correctly is essential before developing applications on FMS.

Note: Any change to rtmp.xml requires a restart of the FMG service.

RTMP supports the following tags.

Tag	Description
Registrations	To specify the FMS machines that will be able to use FMG provided services.
Profiles	To create a unique identifier <ProfileID> to address an app instance on FMS.
Protocol	To specify the communication protocol between FMG and FMS.
CallLegs	To configure parameters related to general audio quality management of RTMP call legs.
AuxConnectionPool	To enable and configure load balancing of calls over RTMP connections between FMG and FMS.

Define Registrations

To specify the server names and the corresponding service names, the format used is <Server host = "server IP/hostname">servicename</Server>, where:

- The hostname can be IPaddress/hostname.domain.

Note: Consistency in addressing FMSs in rtmp.xml is important. For example, if an FMS in <registrations> is addressed "ipaddress:port", the exact same string must be used throughout the file, including the <Profiles> tags.

- The hostname must be suffixed with the port number if the FMS isn't listening on the default port: 1935.
- Service names assigned must be unique for that FMS.

Note: Using same service names from multiple/single FMG to a single FMS leads to failure in registration on conflicting services.

Also, <ControlService> and <LegService> use a shared namespace on any specific FMS.

- <ControlService> can be used to specify the list of FMS and service names for registry connections to be used for providing ControlService.
- <LegService> can be used to specify the list of FMS and the corresponding names of LegService to be available.

Sample code

```
<Registrations>
  <LegService>
    <Server host = "machine-name1"> telephony </Server>
    <Server host = "machine-name2:8506"> telephony_messenger </Server>
    <Server host = " machine-name2:8506"> telephony </Server>
  </LegService>
  <ControlService>
    <Server host = "machine-name1"> telephony_control </Server>
  </ControlService>
</Registrations>
```

Create Profiles

FMG must have a unique identifier <ProfileID> to permit an FMS application to connect to a LegService. This ID is used to address all call legs originating from or ending at that FMS application. To request a LegService connection using the SSAS API, services.request(<serviceName>), FMG must provide LegService registry connection on the host FMS.

FMG provides LegService connection only to those applications that have their profile name listed in rtmp.xml under the <Profile> tag. If, however, the <EnableAutoProfile> is true, any application on the registered FMS can be granted a LegService. In such cases, calls can only be made from FMS to FMG and not from FMG to FMS. This is because call flow entries in workflow.xml are static and hence can't route calls to a dynamically created profile.

Note: When <EnableAutoProfile> is true, FMG automatically creates unique identifier strings for addressing any FMS application that requests LegService with no predefined ProfileID.

At any point, each specified instance of an FMS application can have only one LegService connection from FMG; further requests are ignored.

Important: All entries in <Profile> are mandatory.

The entries used to create Profiles are:

Tag	Description
Profiles	Contains a set of profile that is used by workflow configurations (workflow.xml) to address calls created over the associated FMS server-side application. This also contains automatic profile creation settings.
EnableAutoProfile	Enables FMG to create and assign an automatically created profileID to all Leg Services for FMS applications of which no entry was found in the list of <Profile> tags in rtmp.xml. The default value is false.
DefaultPassCode	Provides the tag value "value" to be used as <PassCode> value for all AutoProfiles created by FMG.
Profile	Denotes a node of XML codes that specify the set of properties used to define a complete profile of an FMS application at FMG.
Profile ID	Specifies a unique profile ID value across rtmp.xml. This profileID can be used in workflow.xml to specify call flows. This tag is mandatory within each listed <profile>.

Tag	Description
Server, Application, Instance	Provides the values for FMS-host, application, and instance name, which are mandatory to specify an application to which this profile ID is assigned. Note: An FMS must be addressed with a consistent value (IPAddress or Hostname) across applicable tags (including those under <registrations>).
Context	Specifies the name of the workflow context (in workflow.xml in FMG) that is to be used to decide the flow of calls originating from this profileID.
PassCode	When FMG connects to an FMS application, the FMS server-side script receives it in application.onConnect() method as the third argument.

Sample code

```

<Profiles>
  <EnableAutoProfile> true </EnableAutoProfile>
  <DefaultContext> rtmp </DefaultContext>
  <DefaultPassCode> passcode_default </DefaultPassCode>
  <Profile>
    <ProfileID> profile_3 </ProfileID>
    <Server> machine-name2:8506 </Server>
    <Application> telephony </Application>
    <Instance> name2 </Instance>
    <Context> rtmp </Context>
    <PassCode> passcodeString </PassCode>
  </Profile>
  ...
  ...
  ...
  <Profile>
    ...
    ...
    ...
  </Profile>
</Profiles>

```

Establish Protocol

<ConnectionProtocolType> specifies the RTMP protocol variant to be used for network connection between FMG and FMS.

The valid values are rtmp or rtmps only. The default value is rtmp.

Note: To configure FMG, in order to make an RTMPS connection with an FMS, you must install the client certificate on the FMG machine.

Sample code

```

<ConnectionProtocolType> rtmp </ConnectionProtocolType>

```

Specify call legs

SSAS LegService APIs, `<setSilenceLevel>` and `<setSilenceTimeout>`, can be used to configure specific values for SilenceLevel and SilenceTimeout at runtime. In a typical use-case, these tags help in catching up the delay in the SIP-to-Flash audio channel/stream. Such delay might be due to unreliable network connectivity between the FMS and the FlashPlayer. For more details on `<setSilenceLevel>` and `<setSilenceTimeout>`, see *Flash® Media® Gateway Leg Service API Reference*.

`<RtmpBuffer>` contains delay catch-up and buffering policies that are specific to the Flash-to-SIP audio channel/stream.

The entries used to specify call legs are:

Tag	Description
CallLegs	Root tag for RTMP call leg specific configurations.
SilenceLevel	Audio will not be published to FMS when activity level stays below the <code><silenceLevel></code> value for <code><SilenceTimeout></code> milliseconds. The default value is 0; range is from 0 to 100. For "0", there will be no silence detection and all frames will be published to FMS.
SilenceTimeout	Audio will not be published to FMS when activity level stays below the <code><silenceLevel></code> value for <code><SilenceTimeout></code> milliseconds. Tag value is in milliseconds; default value is 0.
RtmpBuffer	Configuration tags to configure buffer that is specific for the media flowing from FMS to FMG.
MonitoringInterval	Buffer monitoring interval in seconds. The default value is 10. Valid range is from 1 to 100.
ActivityThreshold	In the buffer catch-up mode, packets with activity level less than <code><ActivityThreshold></code> value can be dropped to improve interactive playback experience. The default value is 0; that is, FMG will not drop any audio frame under the value "0". Valid input values are from 0 to 100.
frameResampleFactor	FMG can resample audio data up to a percentage value of the original sampling rate, to improve interactive experience. The default value is 100; resampling is switched off. Valid values are in the range 50 to 100.
DelayedFrameThreshold	Maximum size of accumulated delayed frames (in number of frames) in a buffer, beyond which frames should be dropped to keep the queue size in control. The default value is 100. Valid integer values can be from 1 to 1000.

Sample code

```
<CallLegs>
  <Audio>
    <SilenceLevel> 10 </SilenceLevel>
    <SilenceTimeout> 1000 </SilenceTimeout>
    <RtmpBuffer>
      <MonitoringInterval> 10 </MonitoringInterval>
      <ActivityThreshold> 5 </ActivityThreshold>
      <frameResampleFactor> 90 </frameResampleFactor>
      <DelayedFrameThreshold> 200 </DelayedFrameThreshold>
    </RtmpBuffer>
  </Audio>
</CallLegs>
```


Configure AuxConnectionPool

A properly configured Auxiliary Connection Pool significantly improves the CPU usage of FMG. When `<AuxConnectionPool>` is enabled, FMG tries to maintain a pool of auxiliary clients connected with the FMS application instance. All these clients have the `client.agent` property set to "FMG Aux Leg Service 1.0". This, along with other identifiers such as the IP address, can be used to define a server-side logic so that connect/disconnect from AuxLeg-service client doesn't interfere with the FMS server-side application logic.

Note: The value of `Auxconnections` must be set such that the product of `<MinFreeSlots>` and `<LoadPerConnection>` is greater than, or equal to, the maximum count of simultaneous new call requests.

The entries used to configure `AuxConnectionPool` are:

Tag	Description
<code>AuxConnectionPool</code>	The root tag for configuring auxiliary connection pool for <code>legService</code> . The default value is false.
<code>LoadPerConnection</code>	The maximum number of call legs whose data transmission load can be multiplexed over an auxiliary connection. A higher value results in more streaming load on each auxiliary connection, that is, higher CPU usage. On the other hand, a decrease in value causes an increase in the number of auxiliary connections required, that is, more system resources. Tag value range is [1, 20]; default value is 5. Note: Capacity of an auxiliary connection to carry call leg data is called slots. Slots are further divided into free and used categories. These categories measure the number RTMP call legs that can be created using the existing set of auxiliary connection.
<code>MinFreeSlots</code>	The value of this tag is used to trigger the creation of new auxiliary connections when the number of available free slots falls below this value. When a sudden burst of new call leg requests arrives and FMG doesn't have any free slots or auxiliary connections, the creation of the requested call leg is forced to fail. Therefore, setups expecting large bursts of new call requests should increase the value suitably to keep sufficient free slots for sudden burst of new call leg requests. Valid Tag value range is from 1 to 1000; default value is 10. Note: One RTMP call leg consumes one free slot on an auxiliary connection; when a slot can't be made available, the call leg creation fails.
<code>MaxFreeSlots</code>	When the number of free slots exceeds this value, FMG triggers the closing of some auxiliary connections. The precondition is that $(\text{MaxFreeSlots} - \text{MinFreeSlots} > 2 * \text{LoadPerConnection})$ must be true. Valid value range is from 1 to 1000; default value is 100.

Sample code

```
<AuxConnectionPool enabled = "true">
  <LoadPerConnection> 5 </LoadPerConnection>
  <MinFreeSlots> 10 </MinFreeSlots>
  <MaxFreeSlots> 100 </MaxFreeSlots>
</AuxConnectionPool>
```

Configuring speex

The `speex.xml` file controls the encoding/decoding performance and quality of the speex codec.

The tags defined in `speex.xml` are as follows.

Tag	Description
Quality	This is an integer value between 1 and 10. Using a higher quality setting means less noise/aliasing, higher complexity, and higher latency. Usually, a quality of 3-6 is acceptable for most desktop uses and a quality of 10 is mostly recommended for high-end audio processing.
Complexity	This is an integer value between 1 and 10. It represents a compromise between CPU-usage and audio quality. Complexity directly affects the CPU resources allocated for the encoder.
Filtering	This is used to turn on/off the input/output for high-pass filtering. A value of 0 means Off and 1 means On.

The default `speex.xml` is as below.

```
<?xml version='1.0' encoding='UTF-8'?>
<Speex>
  <Configuration>
    <Quality> 3 </Quality>
    <Complexity> 1 </Complexity>
    <Filtering> 0 </Filtering>
  </Configuration>
</Speex>
```

Configuring the workflow

The workflow document is divided into a set of contexts, with each context having a set of conditions. Each condition defines a set of AppNodes sequences. Workflow is a sequence of AppNodes, which can be controlled using conditional jumps that are based on the success/failure of each node. Each AppNode is specified by a sequence number, the AppNode name, and the AppNode arguments. For example,

```
< AppNode sequence="1" app="AppNodeName" args="AppNodeArgs"/>
```

Each incoming and outgoing call leg is related to a configured profile. For example, calls to and from different SIP gateways are on different profiles. Each profile has an associated context name. For each incoming call, its context is located in the workflow and the defined conditions are matched to execute the AppNodes.

Note: Any change to `workflow.xml` requires a restart of the FMG service or reload through the control interface.

A list of all the AppNodes and their arguments is as follows:

AppNode	Function	Arguments
wait	Wait (do nothing) on the call leg.	<time in seconds>
answer	Explicitly answer the call leg. All AppNodes, except "wait", answer the call leg at a pre-defined stage (for example, "bridge" answers the call leg when the remote party answers in a SIP-to-RTMP call, "playfile" answers before playing the file). Using the AppNode "answer" in the workflow specifies when to answer the call.	
playfile	Play a file on a call leg.	<filename>, with the file extension. If no extension is specified, it searches for filename.raw. Allowed file extensions are RAW and NM. The files are stored in <Installation Path>/audio.

AppNode	Function	Arguments
record	Record the audio of a call leg.<filename>, with the extension of the file format of the recording. For example, if the file is to be recorded as NM, the filename must be name.nm; else, the recording is done in RAW format.	Allowed file extensions are RAW and NM.
getDTMF	Get dtmf on a call leg.	<p><minDigits> <maxDigits> <maxTries> <timeout> <initialFile> <retryFile></p> <p>All the arguments are optional; however, "I" is mandatory even for each omitted argument. The default values are:</p> <ul style="list-style-type: none"> • minDigits = 1 • maxDigits = 5 • maxTries = 5 • timeout = 10000 (10 seconds) (inter-digit timeout is 1/3 of this value) • initialFile = <filename> (default desired-exten.raw) • retryFile = <filename> (default valid-exten.raw) <p>If a number is negative, the positive value for that number is considered. If <initialFile> or <retryFile> doesn't exist, error.raw is played and failure is returned.</p> <p>The file name must either be the absolute path or the files must be placed in the InstallDir/audio directory.</p>
bridge	Bridge two call legs.	<p><call-leg uuid OR destination to call></p> <p>Destination is in the format "<proto> <called number>@<profile name>", where proto can be sip or rtmp.</p>
hangup	Hang up the call leg.	No argument (null).
statusCheck	Check the status of the last AppNode and jump accordingly.	<true jump sequence id false jump sequence id>
Goto	Jump to the sequence number.	<Jump to sequence id>

The flow of the call can be defined using a combination of the above AppNodes. If an AppNode in the workflow fails, the statusCheck node can be used to determine the next execution node. If statusCheck isn't used, the call is hung up.

Sample code

```

<Workflow>
<ContextList>
<Context name="default">
  <Condition variable="destNum" value="^89.*$">
    < AppNode sequence="1" app="playfile" args="welcome.raw"/>
    < AppNode ?sequence="2" app="getDTMF" args="4|4|3|20|desired-exten.raw|valid-
exten.raw"/>
    < AppNode sequence="3" app="bridge" args="rtmp|$\\{dtmfDigits\\}@profile1 "/>
    < AppNode sequence="4" app="hangup" args="null"/>
  </Condition>
  <Condition variable="destNum" value="^88.*$">
    < AppNode sequence="1" app="playfile" args="welcome.raw"/>
    < AppNode sequence="2" app="record" args="recorded.raw"/>
  </Condition>
</ContextList>
</Workflow>

```

```

        < AppNode sequence="3" app="hangup" args="null"/>
    </Condition>
    ...
    ...
    ...
</Context>
<Context name="rtmp">
    <Condition variable="destNum" value="^89.*$">
        ....
        ....
    </Condition>
    <Condition variable="destNum" value="^88.*$">
        ....
        ...
    </Condition>
    ...
    ...
</Context>
</ContextList>
</ Workflow>

```

A sample condition, using conditional jumps, is as below:

```

<Condition variable="destNum" value="^89.*$">
    <AppNode sequence="1" app="playfile" args="welcome.raw"/>
    < AppNode sequence="2" app="getDTMF" args="4|4|3|6|desired-exten.raw|valid-
exten.raw"/>
    < AppNode sequence="3" app="statusCheck" args="4|6"/>
    < AppNode sequence="4" app="bridge" args="rtmp|${dtmfDigits\}@profile1"/>
    < AppNode sequence="5" app="statusCheck" args="6|7"/>
    < AppNode sequence="6" app="hangup" args="null"/>
    < AppNode sequence="7" app="playfile" args="ss-noservice.raw"/>
    < AppNode sequence="8" app="Goto" args="1"/>
</Condition>

```

In this workflow, at the AppNode sequence number 4, an attempt to bridge the call is made. The next AppNode (sequence number 5) is a statusCheck AppNode, which acts as a decision node. Based on the success or failure of the previous action (sequence 4 bridge AppNode), the statusCheck AppNode can decide the sequence number to execute next. So in this case, if the bridge of the call leg fails in sequence no 4, then at sequence number 5, status-Check makes the workflow jump to sequence 7. At sequence number 7, the AppNode is a playfile node, which plays the ss-noservice.raw file, and then moves to sequence number 8. This AppNode is a GoTo node that makes the workflow jump to sequence number 1, which restarts the entire workflow.

Configuring the HTTP interface

The http.xml file configures the FMG HTTP interface.

The tags under <AdminServer> are as follows:

Tag	Description	Format	Example
HostPort	Specifies the IP address and the port to bind to	[<ip>][:<port>]	The default is to bind to "any" available IP on port 2222.
Allow	Specifies the admin connections to respond to	Comma delimited list of host names, domain names, and full or partial IP address, and the keyword "all" (no quotes required).	<Allow>x.foo.com, foo.com, 10.60.1.133, 10.60</Allow> or <Allow>all</Allow> The default value is <Allow>all</Allow>.
Deny	Specifies which admin connections not to respond to	Comma delimited list of hostnames, domain names, and full or partial IP address, and the keyword "all" (no quotes required).	<Deny>x.foo.com, foo.com, 10.60.1.133, 10.60</Deny> or <Deny>all</Deny> The default value is that none is denied.
Order	Specifies the order to evaluate the <Allow>/<Deny> tags	This can be one of the following: <ul style="list-style-type: none"> • <Order>Deny, Allow</Order>; the request will be processed if not in <Deny> or in <Allow>. • <Order>Allow, Deny</Order>; the request will be processed if in <Allow> and not in <Deny>. 	

The tags under <UserList> are as follows.

Tag	Description
<User name=" \${SERVER.ADMIN_USERNAME} ">	User name for login. The user can connect to the server only from the specified hosts
<Password encrypt="false"> "\${SERVER.ADMIN_PASSWORD}" </Password>	Password for this vhost administrator.
Allow	This is used to turn on/off the input/output for high-pass filtering. A value of 0 means Off and 1 means On.
Deny	
Order	Specifies the order in which to evaluate the <Allow> and <Deny> tags. This can be "Allow,Deny" or "Deny,Allow". The default is "Deny,Allow", which means the access is allowed unless specified in <Deny> and not specified in <Allow>.