

Adobe® Flash® Media Server 4.5.1

Release Notes

Welcome to the Flash® Media Server 4.5.1 release! Flash Media Server 4.5.1 is the next stage in the evolution of scalable high quality interactive media delivery on the web. This release introduces new functionality to help increase quality of service and capacity while helping to reduce your total costs for interactivity and media delivery.

[Minimum system requirements](#)

[What's new in this release](#)

[Issues fixed in this release](#)

[What's changed in this release](#)

[Known Issues](#)

[Install your software](#)

[Uninstall your software](#)

[Other resources](#)

Minimum system requirements

The Adobe Flash Media Server 4.5 system requirements are:

Supported operating systems

- Microsoft® Windows Server® 2008 R2 64 bit
- Red Hat® Enterprise Linux® Server 5.5 64 bit
- Linux CentOS 5.5 64 bit
- Windows XP 32bit (for Flash Media Development Server only)
- Windows 7 64bit (for Flash Media Development Server only)

Hardware requirements

- 3.2GHz Intel® Pentium® 4 processor (dual Intel Xeon® or faster recommended)
- 64-bit operating systems: 4GB of RAM (8GB recommended)
- 1Gb Ethernet card recommended (multiple network cards and 10Gb also supported)

What's new in this release

Corrected issues found in 24x7 Streaming support for HDS and HLS

Issues preventing long-running linear or live HTTP stream output to HDS and HLS formats now support 24x7 live streams without disruption in the encoder or server.

SWF Hasher Tool update for AIR

AIR compatibility was added to the SWF Hasher tool to support SWF Verification functions of PHDS.

RTMPE update

Update to RTMPE for VOD streaming to improve protection of RTMP streams. This update requires the use of Flash Player 11.

Compatibility with AIR for TV and Multichannel Audio

Fix for an issue that prevented multichannel audio (Dolby/DTS for HDS) packaging for HDS to AIR for TV.

Issues fixed in this release:

2887377	Added two new configurations to httpd.conf: HLSJITConfAllowed (under /hls-vod Location) and HttpStreamingJITConfAllowed (under /hds-vod Location). They can be set to true or false. When set to false, jit.conf will be ignored.
2984567	Added the configuration PublishTimeout into the Application.xml of the livepkg application with the value of 0 to allow the publisher to take over the live stream.
2996834	FMS is not streaming certain mp3 audio files completely in Edge Origin configuration as the origin gives file size including id3v1 and id3v2 tags while playback limits the data excluding id3v1.
2902076	Provided the configuration support for setting the cache control headers on HDS/HLS related files (e.g. .bootstrap, .f4m, .f4f, .ts) with HttpStreamingF4MMaxAge, HttpStreamingBootstrapMaxAge and HttpStreamingFragMaxAge parameters.
2962374	When playing a corrupted mp4 file through an edge origin setup, FMS edge may go into livelock wasting CPU and memory on spurious requests for missing content.
2962352	When running ipv6 enabled (dual stack) hosts, the s-ip field in the access logs tends to be incorrect most of the time.
2987222	To disable the minimum password length policy, introduced a configuration in User.xml to use as below. <Root> <PasswordPolicy enable="false"></PasswordPolicy> </Root>
3016016	Sharing around many web cams in connect meeting, introduces delay in transmitting webcam stream and VOIP audio with audio loss/pauses inconsistently during the meeting.
2970089	Setting bandwidth to a very low setting via the setBandwidth API and seeking may stall a stream permanently requiring a reconnection and replay. While moving to 64 bit we made changes which resulted in to a state where negative value got assigned to uint and resulting value caused fms to halt streaming.

2968782	Fixed a Frozen frame present at start of serverside stream recordings when Application/StreamManager/FilterNonKeyFrames key is set to true.
2999679	First Publisher disconnects when another publisher starts streaming
2988750	Fixed a issue with unbuffered video stream where video packets are not dropped consistently when subscriber bandwidth is limited, and leading to latency build up , sometimes as high as 2 minutes
2623244	There is a known issue where HTTP streaming hangs when proxying through the FMS server. So, recommended to configure FMS to NOT to use port 80 for HTTP streaming to work properly. In 4.5.1 this has been taken care by letting Apache serving HTTP streaming through port 80 and is not proxying through FMS.

What's changed in this release

By default, the Apache server installed with FMS will listen on port 80 when opted to install Apache during installation. This will allow both HDS and RTMPT traffic to use port 80. RTMPT will be proxied from Apache to FMS. But, letting RTMPT traffic proxying through Apache, will reduce the quality of service for the RTMPT connections, but resolves the known issue with HTTP Streaming.

Known Issues

The following are known issues in this release:

2953561	When doing HDS and HLS VoD streaming, we recommend scheduled cycling of servers to maintain stable quality of service for end users. Because HTTP streaming is stateless, one can have one or more servers from which content is served, and bring any of them down without service interruption.
2945043	Non-English URL's are not currently supported
2944919	When doing HDS and HLS live streaming, if audio or video is received after the first fragment has been created can lead to a "Track not found" error that results in a recording error. To ensure this does not happen, audio and video should start at the same time.
2929657	Flash Media Server 4.0 inverted the order of delivering NetStream.Unpublish.Success and Connection.Closed messages where now Connection.Closed appears second. This can lead to compatibility issues with previous server script applications.
2913041	Encrypting of an HLS live stream may start to fail under high load, and heavy seeking.
2910985	When the disk load is high you may see "Failed to swap" warnings in the logs which appear to be a nuance of the Windows file system. This may result in a stale file being returned, but is otherwise harmless, and the HDS Live recording should be unaffected.

2908308	MP4/F4V files that have a negative cttts offset may not play properly via HLS and may exhibit artifacts during playback.
2905444	Changing encoder configurations and appending to an existing recording will result in playback problems. A recording is assumed to be a particular audio/video configuration, and changing it mid stream is unsupported.
2885338	Requests for HLS encryption key from iPad 1(3.2.2) over HTTPS will result in a client authentication failure during the SSL handshake.
2873583	Apple's MediaStreamValidator tool does not support validating encrypted media. So, validating an HLS encrypted ts file will report errors.
2873133	Calling NetStream.send() immediately after receiving NetStream.Publish.Start status message may intermittently fail due to a race condition when initializing the Stream object in server-side ActionScript on the receiving server.
2870589	When doing HDS Live recording, an "Inserting safety gap" warning is logged to indicate a drift between the ideal configured fragment duration and actual fragment duration. This can happen if the fragment duration is misconfigured and is not a multiple of the encoder's GOP size, or if the encoder is not able to keep a stable, constant GOP size. When this drift becomes large, a "gap" is inserted in the fragment to represent the drift. It should not result in any observable playback problem.
2859133	Playback issues are seen on iOS device versions 4.3.x when using the HTML5 video source tag and disabling the SlidingWindow configuration on the server (HLSSlidingWindowLength set to 0 - at httpd.conf or at Application.xml or at Event.xml level). Playback stops and clicking the Play button does not resume it. The workaround is to seek to somewhere in the middle of the stream, but seeking to the latest point in playback will also return it to a stopped state. If one tries to resume, playback starts from the beginning instead of the latest point in the timeline. These playback issues are not seen if you put the whole stream URL in the Safari address bar, or on iOS device versions below 4.3.
2426933	The File Plugin API getAttributes() may be called before the E_FILENAME_TRANSFORM event has been processed. As a result, the wrong file attributes may be returned.
2476494	When enabling stream logging via the admin commands logLiveStream and logNetStream, the log directory is always created in the default log directory under the FMS installation directory, regardless of what configured in fms.ini and Logger.xml.
2850151	FMS fails to startup if IPv6 is enabled, but OS doesn't support IPv6
2849944	The LiveCast application edge node does not reconnect to origin node application properly.
2674905	In an n-tier deployment where multiple live streams with different bitrates are being multi-point published from the ingest node to a downstream node, (and synchronized keyframes as required for seamless switching when dynamic streaming), if the streams are being recorded at the downstream node with the intention of utilizing them as DVR streams, the value of "AssumeAbsoluteTime" configuration in application.xml should be set to "true" for the application at the downstream node where the recording is taking place.
2633530	CGI not supported in the Flash Media Server Apache installation.

2501913	Authorization and Access plugins have compilation warnings that are harmless. Removing the warnings now would require an interface change breaking compatibility with pre-existing plugins that were built using the current interface. To maintain compatibility and not require users to have to rebuild and retest their existing plugins, we have decided not to fix these warnings at this time.
2275665	sc-stream-bytes measures the bytes per stream sent from server to client for a particular client. Note that when compared with sc-bytes, which measures all bytes sent from server to client for a particular client, users may notice a discrepancy. The sum total of all sc-stream-bytes for a particular client will usually be. There are a number of situations that could trigger this discrepancy, but users are advised to take note of this if trying to use sc-stream-bytes for billing purposes. Billing should be done based on sc-bytes when possible.
1931450	The MP4/F4V format delivers XMP text under the wrong tag. Specifically MP4/F4Vs parsed by Flash Media Server or Flash Player (via Progressive Download Playback) will produce onXMPData messages containing their XMP text under the tag onXMPData/data. The correct tag is onXMPData/liveXML as specified by XMP definitions. FLV container format renders this data correctly. Users should be prepared for this difference in XMP reporting according to file format change. Note: customers should test the new XMP ActionScript libraries at: http://www.adobe.com/devnet/xmp
1630621	Very low frame rate H.264 videos may take a long time to start if the buffer is set too short.
1611644	The OnImageData event within H264 streams is not supported.
1525751	While using AMF0 mode with AS3, If Date (or XML) is being passed as an argument in a Remote Procedure Call (NetConnection.call, NetStream.send or SharedObject.send), it should be passed as the last argument. No argument should be sent as reference after a Date or XML has been added to the argument list of an RPC. Using AMF3 with AS3 is recommended.
2751666	x-suri-stem is missing x-sname in publish and unpublish event entries
3056883	Playing an pre-encrypted DRM content over Protected RTMP is not supported and it will record an error : Assert failed in .\core\prdsobject.cpp
3022151	The server records an error "Error from libf4f.dll: Unable to create sample descriptor for message " in the server logs when doing a multi-point publish from a f4v file to a livepkg applications. This has to be addressed by detaching a source stream from the server NetStream and re-attach a source stream as in the example below: ns.attach(false); // detach from the source stream ns.attach(application.sourceStream); // re-attach and resubscribe to the source stream ns.publish(streamURI); // republish
3052723	As of 4.5.1, RTMPT can be tunneled through Apache listening on port 80. This is to allow both RTMPT as well as HDS to work over port 80. As a result, it means that RTMP[e] connections on port 80 would fail to connect. This may affect Edge/Origin

	<p>environments where clients connect to the Edge server over RTMPT on port 80. The Edge server would try to connect to the Origin on the same port. However, the protocol between Edge and Origin is RTMP (or RTMPS), and hence would fail to connect.</p> <p>To avoid this, you could do one of the following:</p> <ol style="list-style-type: none"> 1. configure the Origin to have FMS listen on 80; this means that you cannot run both RTMPT and HDS over port 80. 2. configure <code><RouteEntry>*:80;*:1935</RouteEntry></code> in the Edge server's Vhost.xml to remap connections to 80, to 1935 instead. 3. Rewrite the Edge client connection url to replace port 80 with 1935 so that the Edge will attempt to use 1935 instead of the port the client connected on.
3008011	<p>By default, the Apache server installed with FMS will listen on port 80. This will allow both HDS and RTMPT traffic to use port 80 (RTMPT will be proxied from Apache to FMS). This is not possible if FMS is listening on 80. This will result in maximum performance for HDS, but reduced performance for RTMPT. If you wish to optimize for RTMPT, and HDS over 80 is not a concern or issue, you can manually edit <code>Apache2.2./conf/httpd.conf</code> to disable 80 for Apache by removing the line "Listen 80", and enable 80 for FMS by editing <code>conf/fms.ini</code> to add 80 to the list of ports specified in <code>ADAPTOR.HOSTPORT</code>.</p>
3021894	<p>If the client certificate presented by the iOS client has expired, the client authentication, as configured by the default configuration, will fail while serving the encryption keys.</p>
3056535	<p>Recording of PRTMP protected content is not supported as DRM metadata expires and cannot be persisted in a file. Therefore, a server side stream object cannot be used for recording if it is playing protected content from a VoD file.</p>
3015244	<p>When performing HDS Live, appending HDS content to an existing stream/file that is over two days old will result in an error, "Restarting the recording after 2.000000 days is not allowed and it's Stopping the recording. Please change the config parameter <code><MaxRestartGap></code> under <code><Recording></code> in <code>event.xml</code> if you would like to increase this." And the recording will fail. This is an expected behavior as it is not the intent to append to a content that is not of the same timeline. Clear such contents from the streams folder and restart the HDS stream.</p>
3020294	<p>NetStream object cannot be used to play a different stream when PRTMP is enabled. If a NetStream is used to play a single stream, and is reused to play a different stream later, the second play will fail. To workaround this problem, a new NetStream object can be created for each stream play instead of reusing the same NetStream for multiple streams. We have also confirmed that playing multiple streams on the same NetStream with client side playlist still works as expected.</p>
3004381	<p>When HLS encryption is enabled, it is observed that the HLS client on MAC OS X 10.6.x skips nearly one segment length of media during playback. This is not observed on iDevices.</p>
3060434	<p>Letting RTMPT traffic proxying through Apache, reduces the quality of service for the RTMPT connections.</p>

Install your software

To install Flash Media Server on Windows:

1. Double-click the installer application and follow the on-screen instructions.
2. Enter a serial number. If you don't enter a serial number, Flash Media Development Server installs.
3. During the installation process you will be asked to enter a user name and password for the first valid server administrator. The user name is written to fms.ini file and the password is written to Users.xml those are located under the conf folder after installation (You can use the Administration Console to add other administrators later).
4. It is optional to install Apache 2.2 with Flash Media Server. If you have chosen to install Apache, you will be asked if you would like for Apache to listen on port 80. If not, FMS will be using port 80 instead. (If Apache is listening on port 80, all the RTMPT traffic will be proxying through Apache by default. And it is highly recommended user to enter the interface (IP address) FMS is listening on. This is to ensure an IP address is included in HTTPIdent2 tag returning to the player, so that player can use the specified IP address instead of the original hostname when making the RTMPE connection through Apache. And the Ident2 tag can be found in Adaptor.xml). Also, note that letting the RTMPT traffic proxying through Apache, will reduce the quality of service for the RTMPT connections.
5. Accept the default server ports or enter new port numbers.
6. The final installation step gives you the opportunity to view the Readme.htm file, start Flash Media Server, and select the option to start Flash Media Server manually when you reboot your computer. Select any options you'd like, and click Finish.

The installation is complete. If you configured it to start automatically, the Flash Media Server service starts. To start the server manually, select Start > Programs > Adobe > Flash Media Server 4.5 > Start Adobe Flash Media Server 4 and Start Flash Media Administration Server 4.5

To open the Flash Media Administration Console, select Start > Programs > Adobe > Flash Media Server 4.5 > Flash Media Administration Console.

To install Flash Media Server on Linux:

1. Log in as a root user (required to install Flash Media Server).
2. Open a shell window and switch to the directory with the installation file, FlashMediaServer4.5.tar.gz
3. Unzip and untar the installation file. A directory with the installation program is created.
4. Switch to the directory that was just unzipped.

5. Start the installation program with the following command: `./installFMS`
6. The installation program starts and displays a welcome message.
7. Press Enter to start the installation. Follow the installation instructions on your screen. During the process you will be asked to
8. Enter a serial number. If you don't enter a serial number, Flash Media Development Server installs.
9. Enter the installation directory and ports which the server will use
10. It is optional to install Apache 2.2 with Flash Media Server. If you have chosen to install Apache, you will be asked if you would like for Apache to listen on port 80. If not, FMS will be using port 80 instead. (If Apache is listening on port 80, all the RTMPT traffic will be proxying through Apache by default. And it is highly recommended user to enter the interface (IP address) FMS is listening on. This is to ensure an IP address is included in HTTPIdent2 tag returning to the player, so that player can use the specified IP address instead of the original hostname when making the RTMPE connection through Apache. And the Ident2 tag can be found in Adaptor.xml). Also, note that letting the RTMPT traffic proxying through Apache, will reduce the quality of service for the RTMPT connections.
11. Enter an administrative user name and password for the first valid server administrator. These values are written to the `fms.ini` file which is located inside of the `conf` folder after the installation.
12. Enter a user for Flash Media Server processes to run as. The default is the "nobody" user. (The user you select is also the owner of the Flash Media Server files.) Your choices are written to the `fms.ini` file. You can edit the `fms.ini` file to modify this and other security properties later, if needed.
13. Review the summary of the installation options you have chosen, which are displayed in the installer.
14. To start the server manually, go to the installation directory (default is `/opt/adobe/fms/`). Use the command `./fmsmgr server fms start` to start Flash Media Server and `./fmsmgr adminserver start` to start Flash Media Administration Server.

The installation is complete. If you configured it to start automatically, the Flash Media Server service starts.

Uninstall your software

To uninstall Flash Media Server on Windows:

1. Select Start > Programs > Adobe > Flash Media Server 4.5 > Uninstall Adobe Flash Media Server 4.5
2. In the confirmation dialog box, click Yes to uninstall Flash Media Server.
3. Flash Media Server is removed from your computer. A second confirmation dialog box appears when the process is complete.

To uninstall Flash Media Server on Linux:

1. Log on to the server where Flash Media Server was installed.
2. Switch to the root user, or a user with root permissions. Normally you would use su - root to switch to the root user.
3. At the UNIX shell prompt, enter `cd /opt/adobe/fms`.
4. By default, `/opt/adobe/fms` is the directory where Flash Media Server is installed. If you installed the server in a different directory, replace `/opt/adobe/fms` with the actual installation location.
5. Execute the uninstall script `./uninstallFMS`.
6. Follow the displayed uninstall instructions.

Other resources

[Flash Media Server Documentation Center](#)
[Flash Media Server Product Home](#)
[Flash Media Server Developer Center](#)
[Flash Media Server End User License Agreement](#)
[Flash Media Server Online Forums](#)
[Flash Media Server Hands-on Training](#)
[Flash Media Server User Group](#)
[Flash Video Streaming Service](#)
[Flash Media Solutions Providers](#)
[Flash Player Product Home](#)

Copyright © 2011 Adobe Systems Incorporated.