

# LiveStreamCast 1.0 Release Notes

Welcome to LiveStreamCast 1.0! LiveStreamCast is released as sample code so that end users can modify the source code as necessary to implement any desired functionality.

The package includes the LiveStreamCast (LSC) framework (Server Side Actionscript, xml and ini files), as well as a demonstration publisher & subscriber AS2 client application (FLA, AS and SWF).

[Minimum system requirements](#)

[Known Issues](#)

## Minimum system requirements

- Flash Media Interactive Server 3.5 configured in Origin-only mode.  
**Note:** Edge/Origin server configuration is not supported.
- Windows or Linux distributions.
- Flash Media Live Encoder 3.0 for live stream capture.
- Private VIP network between Nodes recommended.

## Known Issues

Following are the known issues with this framework. Bug numbers are included where applicable.

2357728	Dynamic Streaming (i.e., a Play2 transition) is not supported.
2336839	The FLVPlayback 2.5.0.15 component does not send the necessary FCSubscribe to the LSC framework.
2340556	The Sample Video Player that ships with Flash Media Server does not send the necessary FCSubscribe to the LSC framework.
2357774	Currently LiveStreamCast 1.0 only supports RTMP between the various nodes in the topology. All protocols are supported between subscribers and publishers.
2346072	In a publishing scenario with full publishing failover (i.e. two publisher entities connected to two OriginNodes), there is incomplete failover from the primary OriginNode to the secondary if the primary publishing client has its ethernet cable pulled (failover occurs fully when the connection is restored).
2349300	If a publisher stops publishing and closes the NetConnection, an erroneous PublishNotify event is sent to subscribers, after the correct UnPublishNotify event.
--	There is a dependency on a simple server side helper app called "monitor" that

	provides timing information via a stream to that monitor app. The monitor app itself simple consists of the application instance folder, there is no code associated with the app.
2349308	All application instances use the monitor app, so it is possible to have streamname collisions between different applications. In the case of multiple LSC apps, Adobe recommends using multiple monitor app, each dedicated to each application instance.
--	<p>Developers will need to modify two SSAS files to change the name of the hardcoded monitor app, and create the associated new monitor app folder as follows:</p> <p>#1 main.asc &gt; FCSubscribe = change rtmp://localhost/{yourNewMonitorAppName}</p> <p>code snippet:</p> <pre> newClient.FCSubscribe = function( name ) {     if (application.streams[name] == null)     {         application.streams[name] = new ExLiveStream( name, "rtmp://localhost/monitor", 1000, 1 );     }     application.streams[name].addSubscriber(this);     this.call("onFCSubscribe", null, {code:"NetStream.Play.Start", description:name}); } </pre> <p>#2 FCRemoteConnection.asc &gt; FCRemoteConnection &gt; addstream = change rtmp://localhost/{yourNewMonitorAppName}</p> <p>code snippet:</p> <pre> this.nc.addStream = function(name) {     if (application.streams[name] == null)     {         stream = new ExLiveStream( name, "rtmp://localhost/monitor", 1000, 1 );         application.streams[name] = stream;     }     ... } </pre>