Multi-protocol, high performance and small resources footprint software platform for streaming live and on demand audio video content over IP networks. Integrates into existing solution infrastructure and delivers high quality streaming experience.

Unreal Media Server specs:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>Any computer starting with 1MHz CPU, 512MB RAM. Recommended: quad core 3+MHz CPU, 4+GB RAM</td>
</tr>
<tr>
<td>Required system</td>
<td>DirectX 8.0 or higher (comes with OS or service packs)</td>
</tr>
<tr>
<td>software</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Runs as a windows service</td>
</tr>
<tr>
<td>Configuration</td>
<td>Windows GUI application, API for remote or web-based configuration</td>
</tr>
<tr>
<td>Supported file</td>
<td>MP4, ASF*, WMV*, AVI, MKV, MPEG, FLV, OGG, MP3, 3GP, MOV, any other</td>
</tr>
<tr>
<td>container formats</td>
<td>*Windows Media Format runtime v9 or higher is required on the server computer</td>
</tr>
<tr>
<td>Playlist</td>
<td>VOD: Alphabetical and random order file playlists are supported.</td>
</tr>
<tr>
<td></td>
<td>LIVE: Live playlist allows switching between a/v sources streaming to the same player, server-side ad insertion and streaming whole file folders in live mode.</td>
</tr>
</tbody>
</table>

Delivery protocols:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Reach limits</th>
<th>Players</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSocket - HTTP</td>
<td>Penetrates firewalls / proxy servers</td>
<td>Web browsers, HTML5 &lt;video&gt;</td>
</tr>
<tr>
<td>WebSocket - HTTPS</td>
<td>Penetrates firewalls / proxy servers</td>
<td>Web browsers, HTML5 &lt;video&gt;</td>
</tr>
<tr>
<td>RTMP unicast</td>
<td>May be limited by some firewalls</td>
<td>Flash Player</td>
</tr>
<tr>
<td>RTMPT unicast</td>
<td>Penetrates firewalls / proxy servers</td>
<td>Flash Player</td>
</tr>
<tr>
<td>MS-WMSP unicast</td>
<td>Penetrates firewalls / proxy servers</td>
<td>Silverlight, Windows Media Player</td>
</tr>
<tr>
<td>MS Smooth streaming</td>
<td>Penetrates firewalls / proxy servers</td>
<td>Silverlight</td>
</tr>
<tr>
<td>Apple HLS</td>
<td>Penetrates firewalls / proxy servers</td>
<td>iOS, web HLS players</td>
</tr>
<tr>
<td>MPEG2-TS over UDP</td>
<td>May be limited by some firewalls</td>
<td>Set-Top boxes</td>
</tr>
<tr>
<td>RTPT; unicast, multicast</td>
<td></td>
<td>Windows OS: Unreal Streaming Media Player or browser plug-in</td>
</tr>
<tr>
<td>UMS over TCP and RTP; unicast, multicast</td>
<td>May be limited by some firewalls</td>
<td>iOS &amp; Android; mPlayer App</td>
</tr>
<tr>
<td>UMS over HTTP and HTTPS; unicast</td>
<td>Penetrates firewalls / proxy servers</td>
<td>Unreal Streaming Media Player or browser plug-in</td>
</tr>
<tr>
<td>Supported live encoders</td>
<td>RTMP Flash encoders: FMLE, Wirecast, xSplit, OS Broadcaster and other mobile apps; RTSP IP cameras, encoders and software; MPEG2-TS and MS-WMSP encoders: VLC, WME; Unreal Live Server and UM series IP cameras and encoders</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Time-shift and DVR</td>
<td>Live playback with time-shift is supported with HTML5 &lt;video&gt; player, Flash player and Unreal Streaming Media Player. Full DVR is performed by separately running Unreal Archival Server</td>
<td></td>
</tr>
<tr>
<td>User authentication</td>
<td>Live and recorded resources can be configured to use Internal or Session-based authentication</td>
<td></td>
</tr>
<tr>
<td>Internal authentication</td>
<td>Unreal Streaming Media Player asks for Username/Password; users need to be created on the server side</td>
<td></td>
</tr>
<tr>
<td>Session-based authentication</td>
<td>Web portals/applications authorize users; only those authorized users are given access to media resources</td>
<td></td>
</tr>
<tr>
<td>User logging</td>
<td>Full user activity logging including media resources used, amount of data transfer and other information</td>
<td></td>
</tr>
<tr>
<td>User control</td>
<td>Live console allows real-time user monitoring and management</td>
<td></td>
</tr>
<tr>
<td>Resources control</td>
<td>Concurrent connections limit and throughput limit are supported. Live broadcasts can be configured to limit per-user playback time</td>
<td></td>
</tr>
<tr>
<td>Live statistics</td>
<td>Live console displays current server state: current throughput for each delivery protocol, active users and media resources being used</td>
<td></td>
</tr>
<tr>
<td>SDK</td>
<td>API for programmatic user administration and session-based authentication. API for programmatic addition/removal of virtual folders and live broadcasts to/from Media Server configuration metabase. API for programmatic start/stop of Apple HTTP Live streaming, MS Smooth streaming, MPEG2-TS broadcasting.</td>
<td></td>
</tr>
</tbody>
</table>

**Unreal Live Server specs:**

<table>
<thead>
<tr>
<th>OS support</th>
<th>Any 32-bit and 64-bit Windows OS, starting with Windows 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>Dual or quad core 2+MHz CPU, 2+GB RAM</td>
</tr>
<tr>
<td>Required system software</td>
<td>DirectX 8.0 or higher (comes with OS or service packs)</td>
</tr>
<tr>
<td>Process</td>
<td>Runs as a windows service</td>
</tr>
<tr>
<td>Configuration</td>
<td>Windows GUI application, API for remote or web-based configuration</td>
</tr>
<tr>
<td>Live sources</td>
<td>Video: USB, Firewire cameras. DV sources such as camcorders. Analog sources via capture cards; graphics card input or TV-tuner card. Hardware encoding appliances with DirectShow support. Audio: Sound card: microphone, line in. DV audio. USB microphones. TV-tuner audio. Network Streams: Ability to receive live audio / video via RTSP, RTMP, MMS (MS-WMSP), MPEG2-TS, HLS streaming protocols.</td>
</tr>
<tr>
<td>Codecs used for encoding and transcoding</td>
<td>Video: H.264, VC1 (WMV), Microsoft MPEG-4 Video, any other codec Audio: AAC-LC, WMA, MP3, GSM 6.10, any other codec</td>
</tr>
<tr>
<td>Encoding bitrates</td>
<td>Video: Multiple profiles from 40 kbps to 24 mbps. Audio: Multiple profiles from 5 kbps to 320 kbps.</td>
</tr>
</tbody>
</table>
Live streaming latency in Near Real Time mode

<table>
<thead>
<tr>
<th></th>
<th>Video only</th>
<th>Audio + Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency in seconds</td>
<td>0.05-0.3 sec</td>
<td>0.3-1 sec</td>
</tr>
</tbody>
</table>

Latency may grow if network bandwidth is not sufficient for particular stream bitrate.

Streaming delivery modes

Near Real Time mode: minimal latency on the client side. Refer to the table above. Suitable for conferencing or surveillance applications.
Buffered mode: Media Server, Live Server and Player buffer live content to compensate on network congestions. Suitable for live event webcasting; live radio/TV.

Connection to Media Server

Connections can be initiated by Media Server or by Live Server. Multiple Media Servers can connect to the same Live Server. Live Server can send data over TCP and RTP (UDP) transports.

Access restrictions

IP-based restrictions can be set to allow or prohibit Media Servers to use Live Server sources.

Transformations

Built-in logo/watermark, text, timestamp overlays. Ability to insert custom transformation plugin to get access to raw video frames / audio samples.

Recording

Live sources can be recorded based on scheduler or video motion / audio beat detection, independently of streaming. Recording format is ASF or MP4 containing WMV, H.264, MPEG4/WMA, AAC, MP3 media.

Resources control

Live console displays resources currently being streamed and recorded. Live console allows connecting to Media Server and starting/stopping recording of live sources.

SDK

API for programmatic control over recording of live sources. API for connecting live broadcasts to Media Server programmatically. SDK for creating custom Audio/Video transform components.

Client playback applications:

<table>
<thead>
<tr>
<th>OS</th>
<th>Player</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>HTML5 &lt;video&gt;, Flash Player, Unreal Streaming Media Player, Windows Media Player, HLS players, VLC</td>
</tr>
<tr>
<td>MAC</td>
<td>HTML5 &lt;video&gt;, Flash Player, HLS players, QuickTime Player, VLC</td>
</tr>
<tr>
<td>Linux</td>
<td>HTML5 &lt;video&gt;, Flash Player, HLS players, VLC, built-in players in Set-Top boxes</td>
</tr>
<tr>
<td>Android, Symbian, Blackberry, WindowsPhone</td>
<td>HTML5 &lt;video&gt;, HLS players, Flash Player with Flash-enabled browser</td>
</tr>
<tr>
<td>iOS</td>
<td>HLS players, mPlayer App, Flash Player with Flash-enabled browser</td>
</tr>
</tbody>
</table>

Unreal Streaming Media Player specs:

<table>
<thead>
<tr>
<th>OS support</th>
<th>All Windows versions starting with Windows 98</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin for web browsers</td>
<td>Internet Explorer, Firefox, Netscape, Mozilla, Safari, Opera, Chrome.</td>
</tr>
<tr>
<td>Player features</td>
<td>Pause/Resume/Seek controls. Resizable frame - custom size; Full screen. Contrast/brightness enhancements, playlist browsing, volume control. Uses hardware video acceleration. Any number of players can run on a single desktop at the same time (CPU bound).</td>
</tr>
<tr>
<td>DRM</td>
<td>Incoming content is not stored on client computer's hard disk and user is not allowed to save media locally. Streams cannot be ripped.</td>
</tr>
<tr>
<td>SDK</td>
<td>API for ActiveX control: complete automation control for customizing player behavior.</td>
</tr>
</tbody>
</table>
Performance figures:
Tests conducted with Unreal Media Server v10.0 on Intel I7 3MHz CPU, 8Gb RAM, Windows Server 2008.

Live stream parameters:
Video: H264, 30 fps, 720p, 1.5 Mbps
Audio: AAC, stereo, 128 Kbps

Live stream parameters:
Video: H264, 30 fps, 640x480, 512 Kbps
Audio: AAC, stereo, 128 Kbps