# Amazon AppStream

## Developer Guide

# Amazon AppStream: Developer Guide

# Table of Contents


What is Amazon AppStream? ..... 1
- Advantages of Streaming Your Application ..... 1
- What Can You Do with Amazon AppStream? ..... 2
  - CAD, 3D, Simulation ..... 2
  - Games ..... 3
  - Life Sciences ..... 3
  - Enterprise and Productivity ..... 3
- How Does Amazon AppStream Work? ..... 4
  - Amazon AppStream Components ..... 4
  - Architectural Overview of Amazon AppStream ..... 5
  - Amazon AppStream Application Lifecycle ..... 7
- Can My Application Run on Amazon AppStream? ..... 7
  - Supported Operating Systems ..... 7
  - Hardware Specifications ..... 8
  - Video Input Specifications ..... 8
  - Audio Specifications ..... 8
  - Bandwidth Requirements ..... 8
  - Persistent Data ..... 9
  - User Input ..... 9
  - Regions ..... 9

Setting Up ..... 10
- Service Requirements ..... 10
- Sign Up for AWS ..... 10
- Downloads ..... 11
  - Client Applications ..... 11
  - Amazon AppStream SDK ..... 11
  - Amazon AppStream Chrome App SDK ..... 11
  - Amazon AppStream SDK for Java ..... 11
  - Sample Entitlement Service ..... 11

Getting Started ..... 13

Streaming Your Application ..... 16
- Stream in One Click ..... 16
- Chrome App ..... 17
- Android and Fire OS ..... 22
- iPad, iPhone, or iPod Touch ..... 28
- OS X ..... 34
- Windows ..... 35

Managing Your Application ..... 38
- Deploying Your Application ..... 38
  - Deploying with the Interactive Wizard ..... 39
  - Deploying with the Advanced Wizard ..... 42
- Displaying Mouse Pointers ..... 53
  - Remote mouse pointer ..... 54
  - Local mouse pointer ..... 54
- Generating a New Test Connect Link ..... 56
- Viewing All Applications ..... 57
- Viewing Application Summary ..... 58
- Editing an Application ..... 60
- Archiving an Application ..... 63
- Enable Logging on an Application ..... 66
  - AppStream Log Names ..... 66
  - Default Amazon AppStream Logs ..... 67
  - Custom Amazon AppStream Logs ..... 68
  - ..... 69
  - Enabling Amazon AppStream Logging Programatically ..... 71

Increasing Your Service Limits ..... 71
Managing Your Users ..... 73
Building a Client Application ..... 74
Design Considerations ..... 74
Building a Client for Android ..... 74
Lifecycle of a Client for Android ..... 75
Building the Example Client for Android ..... 75
Choosing a Color Subsampling Rate ..... 75
Getting Authorization to Connect to Your Application ..... 76
Sending Your Client Inputs to Your Application ..... 79
Building a Chrome App ..... 82
Lifecycle of a Chrome App ..... 83
Modifying the Example Chrome App ..... 83
Choosing a Color Subsampling Rate ..... 83
Building a Client for iOS ..... 83
Lifecycle of a Client for iOS ..... 84
Building the Example Client for iOS ..... 85
Creating Your Client ..... 85
Choosing a Color Subsampling Rate ..... 90
Sending Your Client Inputs to the Application ..... 90
Terminating Your Client ..... 95
Building a Client for OS X ..... 96
Lifecycle of a Client for OS X ..... 96
Building the Example Client for OS X ..... 97
Creating Your Client ..... 97
Choosing a Color Subsampling Rate ..... 100
Sending Your Client Inputs to the Application ..... 101
Terminating Your Client ..... 107
Building a Client for Windows ..... 107
Lifecycle of a Client for Windows ..... 108
Building the Example Client for Windows ..... 109
Creating Your Client ..... 109
Choosing a Color Subsampling Rate ..... 114
Sending Your Client Inputs to the Application ..... 114
Terminating Your Client ..... 116
Codec and Open Source Licensing ..... 116
What audio and video formats does Amazon AppStream use? ..... 116
How does my client decode video from Amazon AppStream? ..... 116
Does use of Amazon AppStream require proprietary licenses? ..... 117
Are there any open source considerations? ..... 117
Client Application License and Notices ..... 117
1. Use of the Application. ..... 117
2. Limitations. ..... 117
3. Reservation of Rights. ..... 118
4. Updates. ..... 118
5. Termination. ..... 118
6. Disclaimer of Warranties and Limitation of Liability. ..... 118
7. Indemnification. ..... 119
8. Export Regulations. ..... 119
9. U.S. Government End Users. ..... 119
10. Amendment. ..... 119
11. Conflicts. ..... 119
12. Contact Information. ..... 119
Advanced Topics ..... 166
Tools for Amazon AppStream ..... 166
Amazon AppStream SDK ..... 166
Amazon AppStream SDK for Java ..... 166
Amazon AppStream Console ..... 167

Building a Streaming Application ... 167
Design Considerations ... 167
Add Streaming to Your Application ... 168
Testing Your Streaming Application ... 181
Security considerations ... 189
Build an Installer ... 192
Building an Entitlement Service ... 193
Design Considerations ... 193
Building the Entitlement Web Service ... 194
Deploying Your Entitlement Service ... 199
Example Entitlement Request and Response ... 199
Sample Entitlement Service ... 200
Opaque Data Specification ... 205
Using Environment Variables ... 205
Troubleshooting ... 208
Deployment Problems ... 208
Cursor and Screen Elements Move Very Slowly in Windows ... 209
Error: Installer Cannot Find a GPU, Graphics Card, or Sound Card ... 209
Sample Entitlement Service Problems ... 209
Error: **Remote Server Not Responding** ... 209
Error: Web Page Not Available ... 209
Error: 503 ... 209
Streaming Problems ... 210
Error: Connection Timed Out ... 210
Error: No Instance Available ... 210
Error: **Session Closed** ... 211
Error: **Session Requested But No Available Instance** ... 211
Error: WebGL Not Found when using the Chrome App ... 212
Link to Streaming Application No Longer Works ... 212
Link Sent to User Does Not Work ... 212
Two mouse pointers are visible in the Chrome App, OS X client application, or Windows application ... 212
Error Codes ... 212
APPLICATION_DELETION_FAILED ... 212
APPLICATION_INSTALLATION_FAILED ... 213
APPLICATION_LAUNCH_FAILED ... 213
APPLICATION_INSTALLATION_TIMED_OUT ... 213
APPLICATION_LAUNCH_TIMED_OUT ... 213
APPLICATION_RUNTIME_FAILURE ... 214
INTERNAL_FAILURE ... 214
SDK_VERSION_DETECTION_FAILED ... 214
S3_URL_INVALID ... 214
API Reference ... 216
Amazon AppStream REST API ... 216
Hypertext Application Language ... 216
Making HTTP Requests ... 218
Signing Requests ... 221
Handling Errors ... 222
Resources ... 225
Link Relations ... 233
Product Updates ... 240
Release for May 14, 2015 (Latest) ... 240
New client application for Android and Fire OS devices ... 240
New client application for iOS devices ... 241
Connect to a streaming application in one click ... 241
Release for March 26, 2015 ... 241
Updates to the Amazon AppStream Chrome App SDK and Chrome App ... 241
Display the mouse pointers ... 241

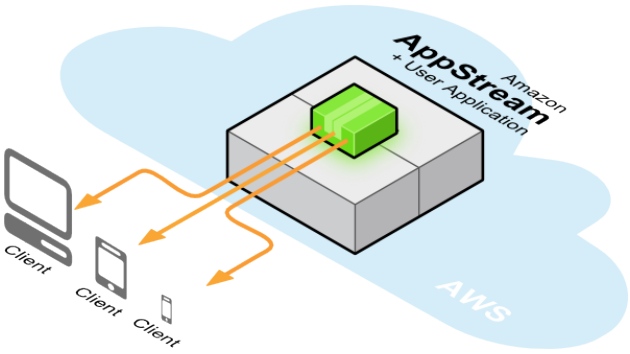
Release for February 2, 2015 ..... 241
Connect to an entitlement service using a Chrome browser ..... 241
New deployment wizards ..... 242
Release for November 20, 2014 ..... 242
Revised deployment from the console ..... 242
Quick link to test your streaming application ..... 242
Release for September 30, 2014 ..... 242
Amazon AppStream now available in Asia Pacific (Tokyo) region ..... 242
Sample entitlement service template updated for use in multiple regions ..... 242
Release for September 16, 2014 ..... 243
Support for Chromebooks and Chrome browsers ..... 243
Release for August 22, 2014 ..... 243
Simplified sample entitlement service deployment ..... 243
Password requirements for standalone mode ..... 243
Release for August 5, 2014 ..... 243
Android client application updates ..... 243
iOS client application updates ..... 243
OS X client application updates ..... 244
Windows client application updates ..... 244
Other Updates ..... 244
Release for June 20, 2014 ..... 244
Up to 20% performance improvement on 64-bit iOS devices ..... 244
Simplified sample streaming application code ..... 244
Remote debugging in standalone mode ..... 244
Other Updates ..... 245
Release for May 30, 2014 ..... 245
Utilization Log is in Coordinated Universal Time (UTC) ..... 245
Utilization Log Contains GPU Metrics ..... 245
Simplified the Application Resource Properties ..... 245
Other Updates ..... 245
Release for May 9, 2014 ..... 245
Logging for Utilization Metrics ..... 245
YUV444 Color Subsampling Support ..... 246
Other Updates ..... 246
Release for April 22, 2014 ..... 246
Standard and User-defined Logging ..... 246
Updated OpenSSL Version ..... 246
Other Updates ..... 246
Release for March 28, 2014 ..... 246
New Billing Practice ..... 246
SDK Version Detected from the Files instead of the Registry ..... 247
Other Updates ..... 247
Release for March 7, 2014 ..... 247
Amazon AppStream Is Available to Anyone with an AWS Account ..... 247
Improved Deployment Error Messages ..... 247
Example Client Application for Mac OS X ..... 247
Increase Your Service Limits ..... 247
Other Updates ..... 247
Release for February 14, 2014 ..... 248
Improved Deployment Experience ..... 248
Silent Installer Requirement ..... 248
Other Updates ..... 248
Release for January 24, 2014 ..... 248
Single Download for the Amazon AppStream SDK ..... 248
Improved Console Load Time ..... 248
Support for Android Hardware Decoding ..... 249
Other Updates ..... 249
Document History ..... 250

AWS Glossary ..... 252

# What is Amazon AppStream?

Amazon AppStream is an application streaming service that lets you stream your existing resource-intensive applications from the cloud without code modifications.

Amazon AppStream enables you to stream your applications from the cloud, reaching more users on more devices. With Amazon AppStream, your application will be deployed and rendered on AWS infrastructure and streams the output to mass-market devices, such as personal computers, tablets, and mobile phones. Because your application is running in the cloud, it can scale to handle vast computational and storage needs, regardless of the devices your customers are using. Amazon AppStream also provides a SDK, giving your development teams the flexibility to build experiences that are unique for your customers, streaming either all or parts of your application from the cloud. Amazon AppStream enables use cases for applications that wouldn't be possible running natively on mass-market devices. Using Amazon AppStream, your applications are no longer constrained by the hardware in your customer's hands.




Amazon AppStream currently supports streaming applications from Microsoft Windows Server 2008 R2 and includes a SDK to build client applications for devices running FireOS, Android, Chrome, iOS, Mac OS X, and Microsoft Windows.

## Advantages of Streaming Your Application

Interactively streaming your application from the cloud provides several benefits:

- **Remove Device Constraints** – You can leverage the compute power of AWS to deliver experiences that wouldn't normally be possible due to the GPU, CPU, memory or physical storage constraints of local devices.
- **Support Multiple Platforms** – You can write your application once and stream it to multiple device platforms. To support a new device, just write a small client to connect to your streaming application.
- **Fast and Easy Updates** – Because your streaming application is centrally managed by Amazon AppStream, updating your application is as simple as providing a new version of your streaming application to Amazon AppStream. You can immediately upgrade all of your customers without any action on their part.
- **Instant On** – Streaming your application with Amazon AppStream lets your customers start using your application or game immediately, without the delays associated with large file downloads and time-consuming installations.
- **Improve Security** – Unlike traditional boxed software and digital downloads, where your application is available for theft or reverse engineering, Amazon AppStream stores your streaming application binary securely in AWS datacenters.
- **Automatic Scaling** – You can use Amazon AppStream to specify capacity needs, and then the service automatically scales your streamed application and connects customers' devices to it.

# What Can You Do with Amazon AppStream?

## CAD, 3D, Simulation

- **Increase trials and improve conversion by eliminating large install file downloads.** – Installation files for graphics-intensive design and modelling applications can be very large, requiring hours for customers to download. Additionally, these applications may require third-party dependencies and plug-ins, complicating the install process. These long download and installation processes can lead to customer frustration and abandonment. With Amazon AppStream, your large installation file can be replaced with a significantly smaller streaming client (as little as 5 MB) since the high-resolution graphics are streamed from the cloud. Your customers can be hands-on with your software in minutes, not hours.
- **Reduce your piracy risk.** – Since the application binary is streamed not downloaded with Amazon AppStream, your exposure to intellectual property theft is reduced. You can increase the distribution of demos or full applications online while at the same time limiting the potential for piracy. For more information on security at AWS, visit our Security Center.
- **Reduce the operational costs of training labs.** – Training lab maintenance can be costly, since you must ensure that all computers are up-to-date with the latest software releases and compatible hardware. With an AppStream Chrome App, for example, new versions of your software suite can be made available instantly to every Chromebook and Chrome browser on Mac, Windows, and Linux desktop in your training lab, easing support and maintenance. With fewer constraints on local hardware, your training lab can be mobilized and made accessible to more customers, partners, professional services, and sales engineers.
- **Help your customers increase their productivity on job sites.** – Empower contractors and designers in the field to make on-location changes to their designs. Today, your customers may be constrained by the graphics-rendering capability of the hardware in their hands, which often means that project updates are put on hold until your customer returns to their office. Build simple client applications for FireOS, Android, Chrome, iOS, Mac OS X, and Microsoft Windows devices with the Amazon AppStream SDK, while all of the heavy computation and storage of the application is done in the cloud. As a result, your customers are more mobile and more efficient on-the-go.

# Games

- **Get players engaged with instant-on demos.** – High-end gaming pushes the boundaries of life-like graphics, which means that game install files can be very large, usually several GBs. Your players may wait several hours for the game to download, or, worse, will grow impatient and abandon the download before it completes and never play your game. With Amazon AppStream, you can create a streaming experience at the start of your game – such as a tutorial or an avatar creator – while the full game downloads in the background. This helps you get gamers' attention right away with an instant-on experience.
- **Reduce your piracy risk.** – Since the game binary is streamed not downloaded with Amazon AppStream, your exposure to intellectual property theft is reduced. You can increase the distribution of demos or full applications online while at the same time limiting the potential for piracy. For more information on security at AWS, visit our Security Center.
- **Deliver stunning visuals, free from hardware constraints.** – With Amazon AppStream, your game design is no longer constrained by the hardware that your players are using. You can develop innovative, high-resolution graphics that can be enjoyed by every user. Because all of the rendering is done in the cloud, you can take advantage of the processing power of the GPU inside Amazon EC2 G2 instances to deliver rich experiences to users across a range of FireOS, Android, Chrome, iOS, Mac OS X, and Microsoft Windows devices.
- **Stream all or parts of your game with the flexible Amazon AppStream SDK.** – Amazon AppStream enables you to combine local experiences and game controls with interactive game elements – such as the background elements and images - that are streamed from the cloud. This "hybrid streaming" scenario will enable quick-twitch game responsiveness and a small installation file size while still providing face-melting graphics.

# Life Sciences

- **Reduce time and capital costs associated with hardware upgrades.** – Medical imaging software is constantly being upgraded to enable medical professionals to diagnose patients with more accuracy, but this software can require hardware upgrades costing hundreds of thousands of dollars. With AppStream, the graphics processing required by medical scanning, image analysis, or surgery simulation solutions is offloaded to AppStream. Offloading the graphics processing allows medical and life sciences software developers to seamlessly deliver their most up-to-date imaging software on a broader range of devices to hospitals, medical centers, and medical schools.
- **Enable faster diagnoses and treatments with real-time, global collaboration between medical professionals.** – Twenty-first century medicine is increasingly a team effort, with a single patient's treatment sometimes involving professionals and specialists in multiple locations across the country or around the world. Amazon AppStream can help medical centers become more efficient by enabling real-time viewing of patient scans, live video, and online collaboration by specialists across the globe on FireOS, Android, Chrome, iOS, Mac OS X, and Microsoft Windows devices of their choice. By providing specialists flexibility and mobility in viewing their patients' records, they can reduce their time to treatment.

# Enterprise and Productivity

- **Deliver your resource-hungry Windows application to mass-market devices.** – Deliver a single Windows application to your employees on lower-resource workstations and devices without compromising quality. Unlike legacy remote application solutions, AppStream provides low latency and broad cross-platform support for an application, so your employees can use high-end software on mass-market devices.
- **Eliminate employee frustrations associated with slow application responsiveness.** – Upgrade the user experience on your existing cloud-hosted software application. The Amazon AppStream STX

protocol provides better latency than legacy application streaming solutions, which means that your users' experience is more fluid and responsive.

- **Increase your support reach while reducing your maintenance overhead.** – Amazon AppStream enables you to stream a software application from the cloud, enabling you to manage upgrades from a central point and ensure all users are always using the latest version.
- **Reduce large, up-front IT operating expenses.** – Since Amazon AppStream runs on AWS infrastructure, there are no large, up-front data center investments required for you to stream an application. You will only pay for the streaming hours you use. This model enables your company, large or small, to furnish your work force with the software tools they need to grow your business.
- **Make your Mac users happy by seamlessly supporting Windows applications on their preferred OSX or iOS device.** – Users across your enterprise have different personal device preferences that complicate IT application support needs. Providing support for an application across all devices is difficult to develop and maintain. Amazon AppStream enables you to build simple clients with targeted user experiences for FireOS, Android, Chrome, iOS, Mac OS X, and Microsoft Windows devices that allow your users to access the applications that you support on their preferred device. Amazon AppStream mobilizes your workforce, and makes them more productive.

# How Does Amazon AppStream Work?

Amazon AppStream provides a framework for you to host an application on AWS infrastructure and stream the input and output of the application to clients running on consumer devices such as PCs, mobile phones, and tablets.

In building your product, you provide the logic for the application, the client, and user authentication and authorization. This gives you the flexibility to create an end-to-end solution specifically tailored to your business and customer requirements.

The following topics describe the components that make up Amazon AppStream and explains how they work together to provide a high-definition, responsive experience for your users.

**Topics**

- Amazon AppStream Components (p. 4)
- Architectural Overview of Amazon AppStream (p. 5)
- Amazon AppStream Application Lifecycle (p. 7)

## Amazon AppStream Components

Streaming an application from Amazon AppStream involves several components working together, some are AWS products, and others you supply.

**Topics**

- Amazon AppStream Host (p. 4)
- Streaming Application (p. 5)
- Client Applications (p. 5)
- Entitlement Service (p. 5)

### Amazon AppStream Host

Amazon AppStream hosts your application on EC2 instances. Each host runs on a very large instance type called a GPU instance. Each GPU instance provide large amounts of parallel processing power. For more information, see GPU Instances in the *Amazon Elastic Compute Cloud User Guide*.

## Streaming Application

The *streaming application* is the code that you plan to host on Amazon AppStream. This can be a current application that you deploy without modification or a new application that you design specifically to work with Amazon AppStream.

## Client Applications

*Client applications* are lightweight applications that run on consumer devices. They decode the audio and video output of your application and display it on the device. They also encode user input from the device and return it to the application. Thus, they provide a fully interactive experience to your users.

Each device type requires a client application for that platform. For example, if you want your customers to be able to access your application from both iPhone mobile phones and Android tablets, you would provide two client applications, one for iOS and one for Android. Both client applications access the same streaming application. To support new device types, simply create a new client application for that device. You do not have to change the streaming application to support a new client application.

Example client applications are available as a Chrome app for the Chrome browser and for devices and computers running the Android, iOS, OS X, and Windows operating systems. To download a client application, see Streaming Your Amazon AppStream Application (p. 16).

## Entitlement Service

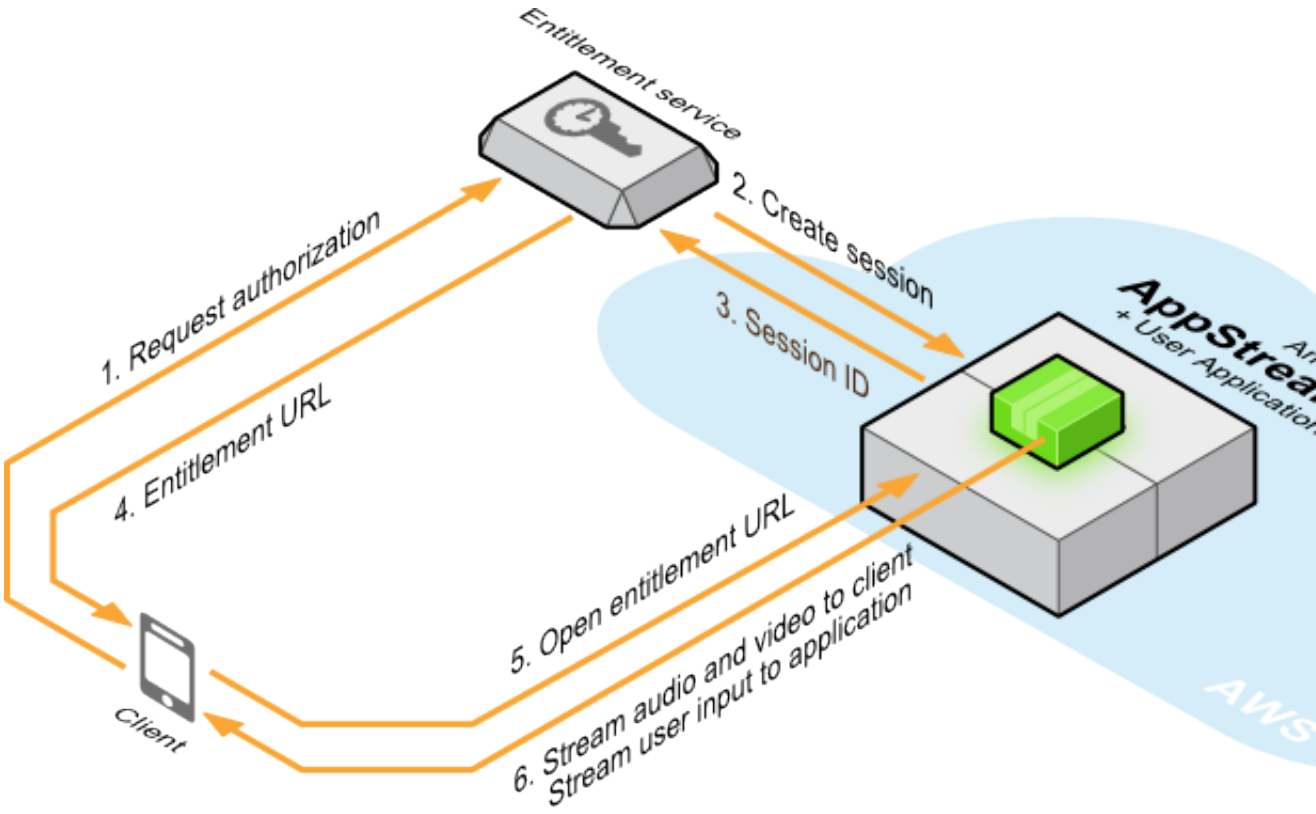
An *entitlement service* authenticates and authorizes users. It is the gatekeeper between clients and your application, ensuring that only those clients entitled to access your application can do so. Your entitlement service may authenticate users in a variety of ways: by comparing user login credentials to a list of subscribers in a database, by using an external login service such as Login with Amazon, or by simply authenticating all clients.

After the entitlement service creates a new session, it returns the session identifier to the authorized client as an entitlement URL. The client then uses the entitlement URL to connect to the application.

When you deploy a streaming application to Amazon AppStream, you also deploy an entitlement service that you can modify to authenticate and authorize users.

# Architectural Overview of Amazon AppStream

The following diagram illustrates how the various components of Amazon AppStream and your product work together. For information about the individual components, see Amazon AppStream Components (p. 4).




## How a Client Application Connects to the Streaming Application

In order to connect to a streaming application hosted on Amazon AppStream, the client application needs the service to create a new streaming session, and the client application needs credentials to access that session. The authentication and authorization of clients to access applications is handled by an entitlement service, which handles the negotiation between the client application and Amazon AppStream.

The steps of authorizing a client application to access a streaming application are as follows:

1. **Request authorization** – The client application calls the entitlement service and requests authorization for a set of user credentials.
2. **Create a session** – If the entitlement service successfully authenticates the user credentials and verifies they are authorized to access the application, the entitlement service calls the Amazon AppStream service to create a new client session.
3. **Return session ID** – The Amazon AppStream service creates a new client session and returns the session identifier to the entitlement service.
4. **Return entitlement URL** – The entitlement service uses the session identifier returned by the Amazon AppStream service to create an entitlement URL, which it returns to the client.
5. **Open entitlement URL** – The client redeems its entitlement to access the application by opening the entitlement URL. When it does, the Amazon AppStream service redirects the client to the IP address of the Amazon AppStream host hosting the streaming application.
6. **Stream output and receive user input** – The streaming application streams audio and video to the client application, and the client sends user input to the streaming application. The Amazon AppStream service manages the connection for best performance given the current network conditions.

# Amazon AppStream Application Lifecycle

When you deploy and manage your application on Amazon AppStream it goes through a series of states. For more information, see Deploying Your Streaming Application to Amazon AppStream (p. 38) and Managing Your Streaming Application (p. 38).

## Building

During the Building state, Amazon AppStream deploys your application. This has several substates, which are displayed in the console.

- **Preparing environment** – Amazon AppStream allocates the IT infrastructure required to host your application.
- **Copying application** – Amazon AppStream copies the installer for your application from Amazon S3 to the host. A presigned URL that you provide gives the installer access to the content stored in your AWS account. The installer installs your application and its dependency files.
- **Installing application** – Amazon AppStream calls the installer, including any command-line parameters you provided, to install your application on the host.
- **Creating AMI** – Amazon AppStream creates an Amazon Machine Image (AMI) of the host with your application installed. For more information, see Amazon Machine Images (AMI) in the *Amazon Elastic Compute Cloud User Guide*.

## Active

During the Active state, your application is ready to accept user connections.

## Archiving

The Archiving state begins after you archive your application but before it is fully archived. In this state, the application continues to stream to existing client connections, but no longer accepts new client connections. When all client connections have concluded, Amazon AppStream moves the application into the **Archived** state.

## Archived

Once it is fully archived, your application no longer accepts client connections. This state corresponds to **Archived** in the console.

## Error

A problem during deployment puts your application into the Error state.

# Can My Application Run on Amazon AppStream?

Any application that can run on the Supported Operating Systems (p. 7) can run on Amazon AppStream.

# Supported Operating Systems

The application must be able to run on the Microsoft Windows Server 2008 or later operating system. Windows Server 2008 is a 64-bit operating system; 32-bit applications are supported through the WoW64

extensions. If your application has other dependencies, such as the .NET Framework, you can include them as part of your application installer.

# Hardware Specifications

The Amazon AppStream servers that Amazon AppStream uses to host applications are GPU instances provided by Amazon Elastic Compute Cloud (Amazon EC2). The GPU instances have the following virtualized hardware. For more information, see GPU Instances in the *Amazon EC2 User Guide for Linux Instances*.

- CPU: High Frequency Intel Xeon E5-2670 (Sandy Bridge) Processors (8 virtual cores at 2.5 GHz each)
- RAM: 15 GiB
- Instance storage: 60 GB
- GPU: 1 * NVIDIA GK104 GPU with NVIDIA GRID K520
- GPU memory: 4 GB
- I/O performance: High (we recommend 2 Gbps/instance)
- EBS-optimized: Yes (500 Mbps)
- 64-bit platform: Windows or Amazon Linux
- HVM only

# Video Input Specifications

Amazon AppStream accepts YUV 444 video input from the application and outputs YUV 444 to the client. Amazon AppStream does not perform any color conversion internally.

# Audio Specifications

Your application can push audio to the Amazon AppStream library or it can make use of the automatic audio capture feature that Amazon AppStream provides. If your application uses automatic audio capture, the application simply writes audio as a normal Microsoft Windows application would. If the application pushes audio to the Amazon AppStream library, observe the following audio specifications:

- 48000Hz sampling rate
- 2 interleaved channels
- 16 bit signed

# Bandwidth Requirements

When accessing an application hosted on Amazon AppStream, the client must be continuously connected to the Internet with a minimum bandwidth of 3 Mbps.

Amazon AppStream recommends at least 3 Mbps for streaming video at 720 pixels at 30 frames per second (720p30). When more bandwidth is available, Amazon AppStream allows the encoding rate to go as high as .2 bits per pixel, which at 720p30 is about 5.53 Mbps. When less bandwidth is available, Amazon AppStream allows the encoding rate to go as low as .02 bits per pixel, which at 720p30 is about 553 Kbps. Amazon AppStream adapts the video bit rate based on available bandwidth. If Amazon AppStream measures the available bandwidth at 3 Mbps, for example, Amazon AppStream sets the encoding bit rate to meet that constraint.

# Persistent Data

Because Amazon AppStream hosts your application on Amazon AppStream hosts, any data stored on the server is lost when the client session ends. If your application needs to persist data between client sessions your application should record the data in a persistent store such as Amazon S3, Amazon RDS, or DynamoDB.

# User Input

You can stream a variety of user inputs from the client to your application:

- **Keyboard** – transmits keyboard data from the client application to the streaming application.
- **Mouse** – transmits mouse move and mouse click data from the client application to the streaming application.
- **Touch** – transmits multi-touch and gesture data from the client application to the streaming application.
- **Raw input** – transmits a raw stream of bytes from the client application to the streaming application. You can use this to transmit user data that does not fit the keyboard, mouse, or touch models. For example, accelerometer data.

# Regions

Amazon Web Services run on servers in data centers around the world. These are organized by geographical region. When you launch an application on Amazon AppStream, you must specify which region to launch it into. You might choose a region to reduce latency, minimize costs, or address regulatory requirements. For the list of regions and endpoints supported by Amazon AppStream, go to Regions and Endpoints in the *Amazon Web Services General Reference*.

# Setting Up Amazon AppStream

Before you begin using Amazon AppStream, you need to understand the system requirements, sign up for an AWS account, and download the necessary development tools.

**Topics**


- Service Requirements (p. 10)
- Sign Up for AWS (p. 10)
- Downloads (p. 11)


## Service Requirements

The streaming application must run in the 64-bit version of Microsoft Server 2008 R2 or earlier.

Client applications must run in one of the following operating systems:

- Android 4.0 (Ice Cream Sandwich) or later
- Apple iOS 8.0 or later
- Chrome browser or a Chromebook with the latest updates installed
- Mac OS X Mountain Lion (10.8.5) or later
- Microsoft Windows 7 or later

## Sign Up for AWS

You need an AWS account in order to use Amazon AppStream.

If you do not have an AWS account, use the following procedure to create one.

**To sign up for AWS**

1. Open http://aws.amazon.com/ and click **Sign Up**.
2. Follow the on-screen instructions.

AWS notifies you by email when your account is active and available for you to use.

# Downloads

To help you get started, several downloads are available, including client applications, software development kits, and a sample entitlement service.

## Client Applications

Client applications are available for the following operating systems or devices:

- **Android and Fire OS.** – http://amzn.com/axc
- **Chrome App** – https://chrome.google.com/webstore/detail/amazon-appstream-example/jooibckpjiaaoaffacacomklfdodlmah
- **iPad, iPhone, or iPod Touch** – https://itunes.apple.com/us/app/amazon-appstream-basic-client/id974559711?ls=1&mt=8
- **Mac OS X** – https://s3.amazonaws.com/appstream-clients/latest/appstreamclient-osx.zip
- **Windows** – https://s3.amazonaws.com/appstream-clients/latest/appstreamclient-windows.zip

## Amazon AppStream SDK

The Amazon AppStream SDK provides tools for developing Amazon AppStream-compatible Windows streaming applications as well as client applications for Fire OS, Android, iOS, Mac OS X, and Windows.

To download the Amazon AppStream SDK, go to https://s3.amazonaws.com/appstream-sdk/AppStreamSDK_1.6.0.135.zip.

## Amazon AppStream Chrome App SDK

The Amazon AppStream Chrome App SDK provides the example Chrome app and native client files to create a Chrome app.

To download the Amazon AppStream Chrome App SDK, go to https://s3.amazonaws.com/appstream-sdk/AppStream_Chrome_SDK-1.2.1.189.zip.

## Amazon AppStream SDK for Java

The Amazon AppStream SDK for Java includes functions you can call to interact with Amazon AppStream. These functions wrap the Amazon AppStream REST API (p. 216) and handle details like signing the requests that are sent to Amazon AppStream. The most common uses of the Amazon AppStream SDK for Java are to write an entitlement service that authorizes user access to your streaming applications and to automate Amazon AppStream.

To download the Amazon AppStream SDK for Java, go to https://github.com/awslabs/aws-appstream-sdk-java/.

## Sample Entitlement Service

You can download the following files to use and deploy a sample entitlement service:

- For an Amazon AppStream entitlement service template, go to https://s3.amazonaws.com/appstream-sdk/appstreamEntitlementService.template. You can use this AWS CloudFormation template to deploy the sample entitlement service.

- For the Amazon AppStream sample entitlement service, go to https://s3.amazonaws.com/appstream-sdk/sample-entitlement-service.jar.
- For the source code for the sample entitlement service, go to https://s3.amazonaws.com/appstream-sdk/sample-entitlement-service-src.zip.

To use the sample entitlement service, see Amazon AppStream Sample Entitlement Service (p. 200).

For more information about creating your own entitlement service, see Building the Entitlement Web Service (p. 194).

# Getting Started with Amazon AppStream

Amazon AppStream deploys interactive applications on AWS infrastructure and streams input and output between the application and clients running on end-user devices such as personal computers, tablets, and mobile phones.

To deploy your application, you will need an application and device that meets the Service Requirements (p. 10). You can deploy your application by using the interactive deployment wizard in a Google Chrome or Mozilla Firefox browser. If you have an Apple Safari or Internet Explorer browser, go to Deploying with the Advanced Deployment Wizard (p. 42)

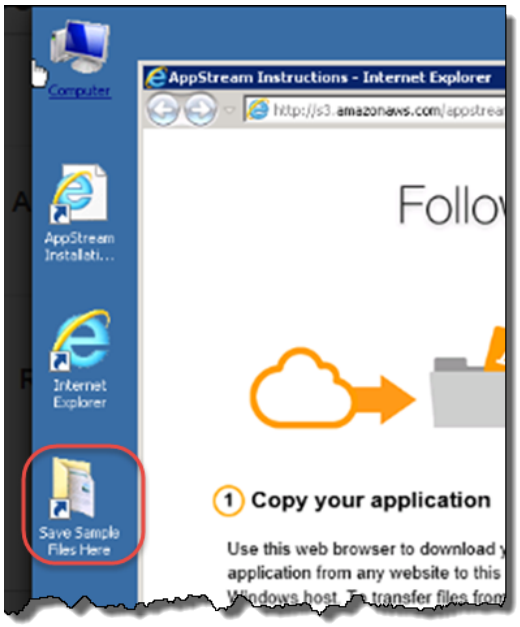
If you do not have an application, you can try deploying one of the following applications. Please note that these are open-source programs made available by third parties. Amazon cannot provide any warranties for these applications or accept liability for your use of them. You are responsible for reviewing and complying with the applicable license terms.

- **Blender** – A free and open-source 3D animation suite available at http://www.blender.org
- **GNU Image Manipulation Program (GIMP)** – A free and open-source program for photo-retouching, image-composition, and image authoring available at http://www.gimp.org
- **FlightGear Flight Simulator** – A free and open-source flight simulation program available at http://www.flightgear.org

### To deploy an application on Amazon AppStream

1. Open a window in a Chrome or Firefox browser.
2. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
3. Click **Deploy an application**.
4. Follow the on-screen instructions.

If your application requires additional files, such as data files, copy those files to the `My Documents` folder, which you can open from the **Save Sample Files Here** shortcut on the Windows live host desktop.



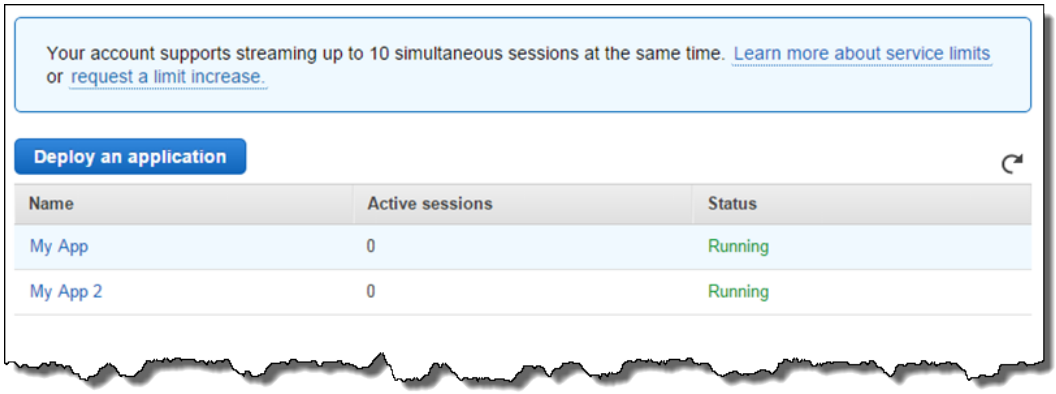

During the streaming application deployment, you may experience poor performance. This is caused by the Windows Remote Desktop Protocol (RDP) connection to the instance and does not reflect the actual streaming performance. For more information, see Cursor and Screen Elements Move Very Slowly in Windows (p. 209).

After you successfully deploy and stream your application, you can allow other users to use your streaming application by sending them a new quick link. The quick link is valid for 72 hours after creation and can only be used once.

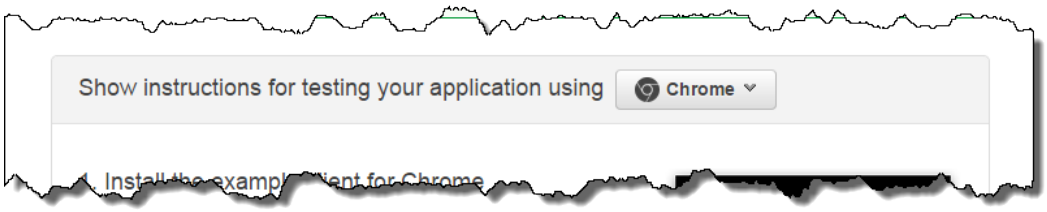
### To generate a new test connect link

1. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.

2. In the list of applications, click the name of the streaming application to view.




3. In **Show instructions for testing your application using**, select the operating system of your device or computer.




4. Do one of the following:

   - For Chrome, click **Generate a new quick link** in Step 2.
   - For other operating systems, click **Generate a new test URL** in Step 3.

# Streaming Your Amazon AppStream Application

To stream your streaming application, you can use the example client applications. Client applications are available as a Chrome app and for the Android and Fire OS, iOS, OS X, and Windows operating systems. Before connecting, ensure that your streaming application has available sessions.

**Active Sessions** in **Application Summary** will show if your streaming application has available sessions. To see if your streaming application has available sessions, see Viewing Application Summary (p. 58).

> **Note**
> If you cannot connect to your streaming application, make sure that your firewall allows traffic through TCP port 80 and 8080 and UDP ports 9070 through 9097.

Follow the appropriate instructions for your version:


- Streaming your Application in One Click (p. 16)
- Chrome App (p. 17)
- Android and Fire OS (p. 22)
- iPad, iPhone, or iPod Touch (p. 28)
- OS X (p. 34)
- Windows (p. 35)


## Streaming your Application in One Click

Your users can connect to your streaming application in one click through a browser in devices that use Android, Fire OS, and iOS as well as the Chrome App. You can connect by using a URL in a deep link format. This topic shows you how to construct this URL.

### To create a deep link to an entitlement service

1. Start with the following deep link:

   http://start.appstream.amazonaws.com/v3?applicationId=*applicationId*&desServer=*server*&identityToken=*username*

2. Replace the *italicized text* with the following parameters:

- For *applicationId*, insert the application ID of your streaming application, which is available from the summary page in the Amazon AppStream console. For more information about getting the application ID, see Viewing Application Summary (p. 58).
- For *server*, insert the URL of your entitlement service.
- For *username*, insert a user name that can connect to your streaming application.

For example, if your streaming application uses the following:

- Application ID – **5a3d0f51-5fa8-47d7-8d37-14c656ab90cd**
- Entitlement service – **des.example.com**
- User name – **marymajor**

Then you would use the following deep link in a browser to connect to your streaming application:

https://appstream.amazonaws.com/?applicationId=5a3d0f51-5fa8-47d7-8d37-14c656ab90cd&desServer=des.example.com&clientToken=marymajor

# Chrome App

The Amazon AppStream Example Client for Chrome is an app that you add to the Chrome browser that can connect to a streaming application.

This section contains the following procedures:

Installing the Chrome App (p. 17) – shows how to add the Chrome App to the Chrome browser.

Streaming an application through an entitlement service (p. 17) – use this method if you have the entitlement service URL, the application ID of the streaming application, and a user name that can connect to the streaming application.

Streaming an application from the Amazon AppStream console (p. 18) – use this method to test your streaming application by using a link in the application summary page.

Stream an application in standalone mode (p. 21) – use this method for a streaming application in standalone mode.

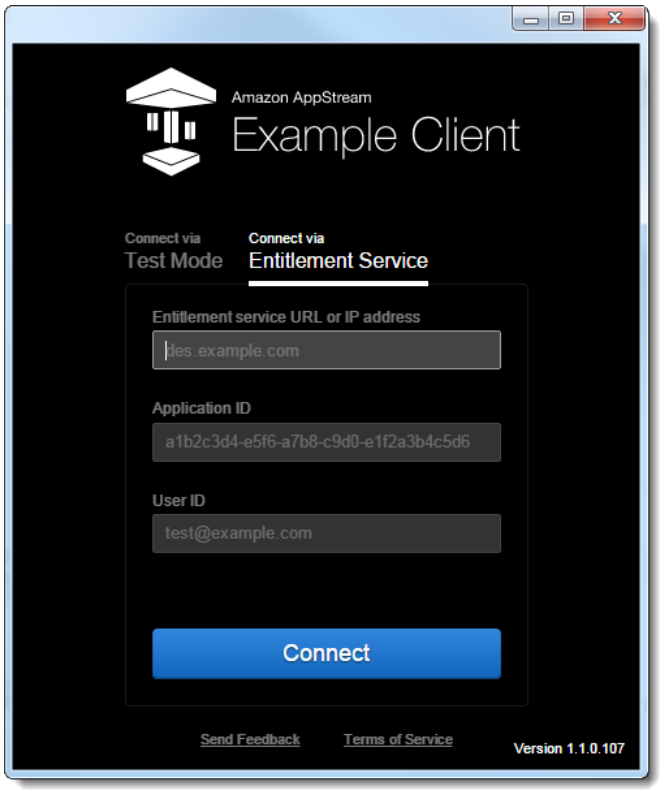
Changing the settings of the app (p. 21) – change the settings to show the mouse pointer in the Chrome app, the streaming application or both, show verbose debugging information to help you solve a problem, and change the inactivity time.

### To install the Chrome App

1. In a Chrome browser, go to https://chrome.google.com/webstore/detail/amazon-appstream-example/jooibckpjiaaoaffacacomklfdodlmah.
2. Click **FREE** and then click **LAUNCH APP**.

### To stream an application through an entitlement service

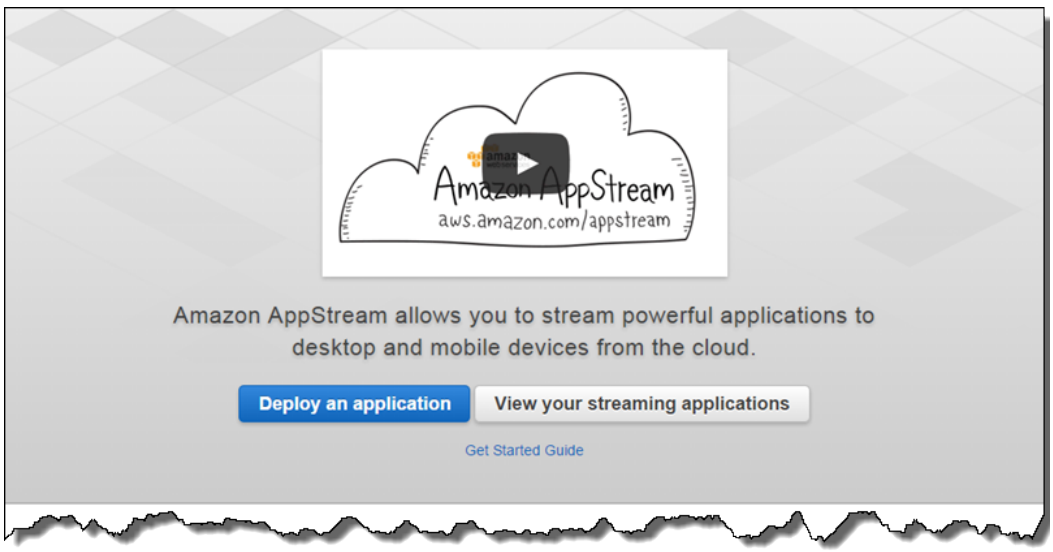
1. Start the Chrome App and choose **Connect via Entitlement Service**. Type the entitlement service URL or IP address, application ID, and user ID in the appropriate text boxes.



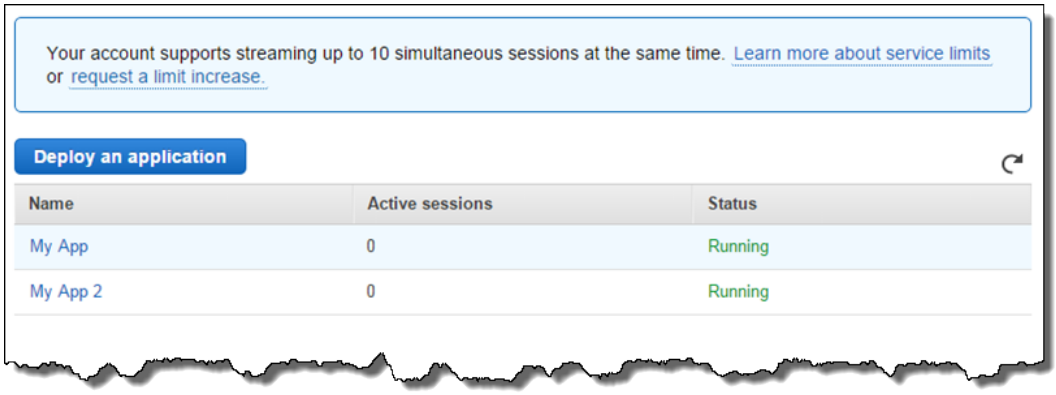

2. Click **Connect**.

### To stream an application from the Amazon AppStream console

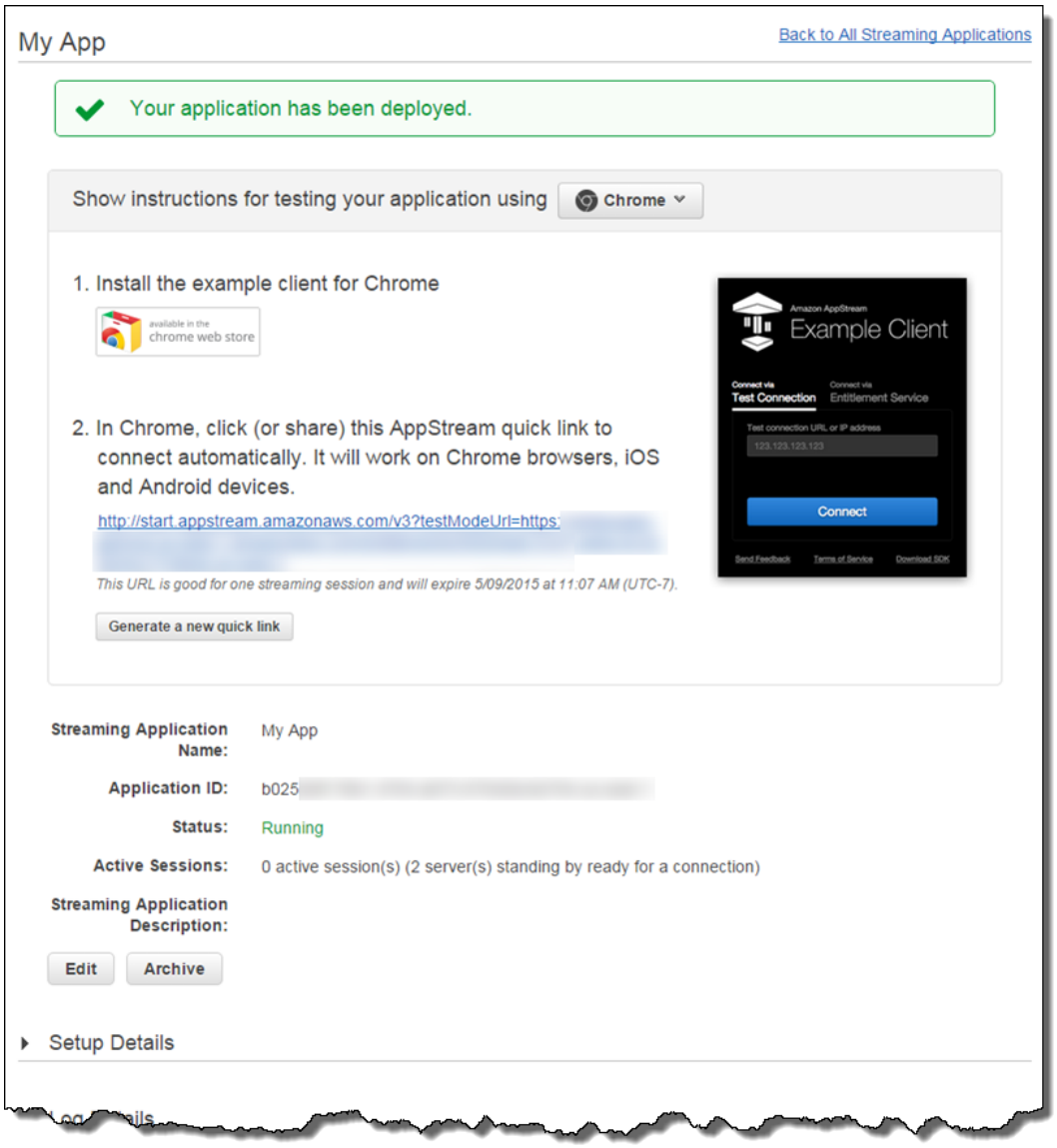
1. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.



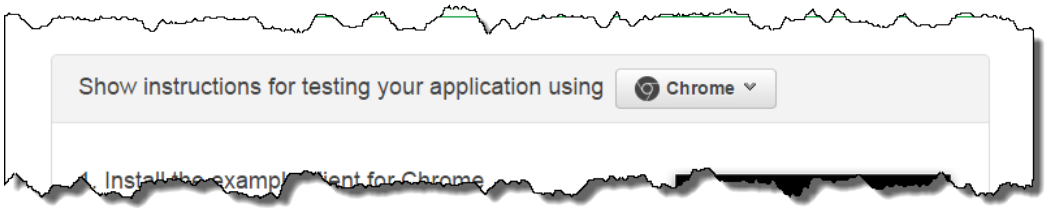
3. In the list of applications, click the name of the streaming application to view.




The application summary page displays information about the streaming application.




4. In **Show instructions for testing your application using**, select **Chrome**.

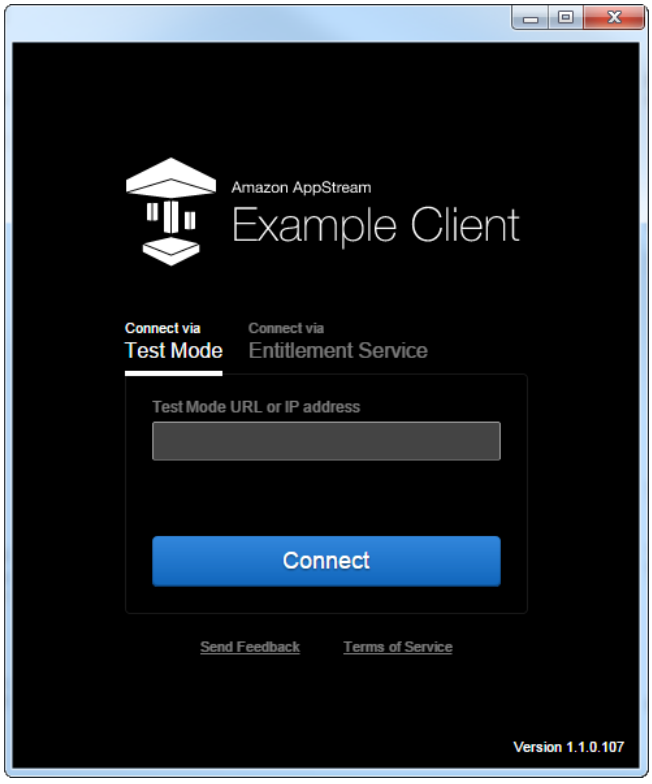



5. Click the link in step 2 to start the Chrome app and connect to your streaming application. The URL is valid for 72 hours and can only be used once. You can share this URL within the 72 hour period, but only one user can use this URL.

If you have already used this link, you can generate a new test connect link. For more information, see Generating a New Test Connect Link to Stream Your Application (p. 56).

### To stream an application in standalone mode

1. Start the Chrome App and choose **Connect via Test Mode**. In **Test Mode URL or IP address**, type the IP address of your EC2 instance.



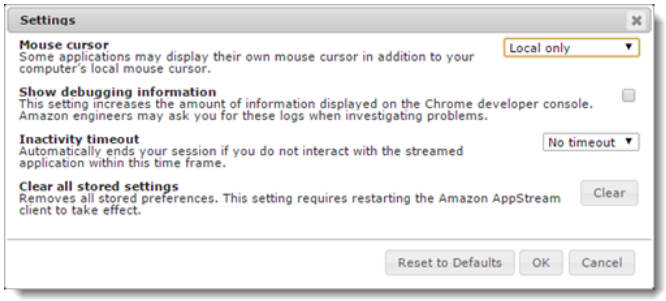

2. Click **Connect**.

### To change the settings of the Chrome App

1. Start the Chrome App and choose **Settings**.
2. In **Settings**, do the following:

   - In **Mouse cursor**, choose **Local** to display the mouse pointer of the Chrome App. Choose **Remote only** to display the mouse pointer of the streaming application. Choose **Local and Remote** to display the mouse pointers in both the client and streaming applications.
   - Select **Show debugging information** to display the Chrome developer console with verbose log information to help you find the cause of a problem. To display the Chrome developer console with the verbose log information, select this check box and then click **OK** to close the dialog box. In the client application, right-click the text box and select **Inspect Element**. If you are using a Chromebook, point to the text box and then click the touchpad with two fingers or press the **Alt** key while clicking with one finger and then select **Inspect Element**.
   - Choose **Inactivity timeout** and select a time after inactivity when the client application will disconnect from the streaming application. You are still incurring charges if a client application is connected to a streaming application but the client application is not sending any user inputs. You can reduce your cost by choosing an appropriate inactivity time.

- Choose **Clear** in **Clear all stored settings** to remove the stored preferences. Restart the Chrome App to see the effect.




3. Click **OK** to save the settings.

# Android and Fire OS

The Amazon AppStream Example Client is an app that your can run on a device running Android or Fire OS.

This section contains the following procedures:

Installing the application (p. 22) – shows how to install the client application on your device.

Streaming an application through an entitlement service (p. 22) – use this method if you have the entitlement service URL, the application ID of the streaming application, and a user ID that can connect to the streaming application.

Streaming an application from the Amazon AppStream console (p. 23) – use this method to test your streaming application by using a link in the application summary page.

Stream an application in standalone mode (p. 26) – use this method for a streaming application in standalone mode.

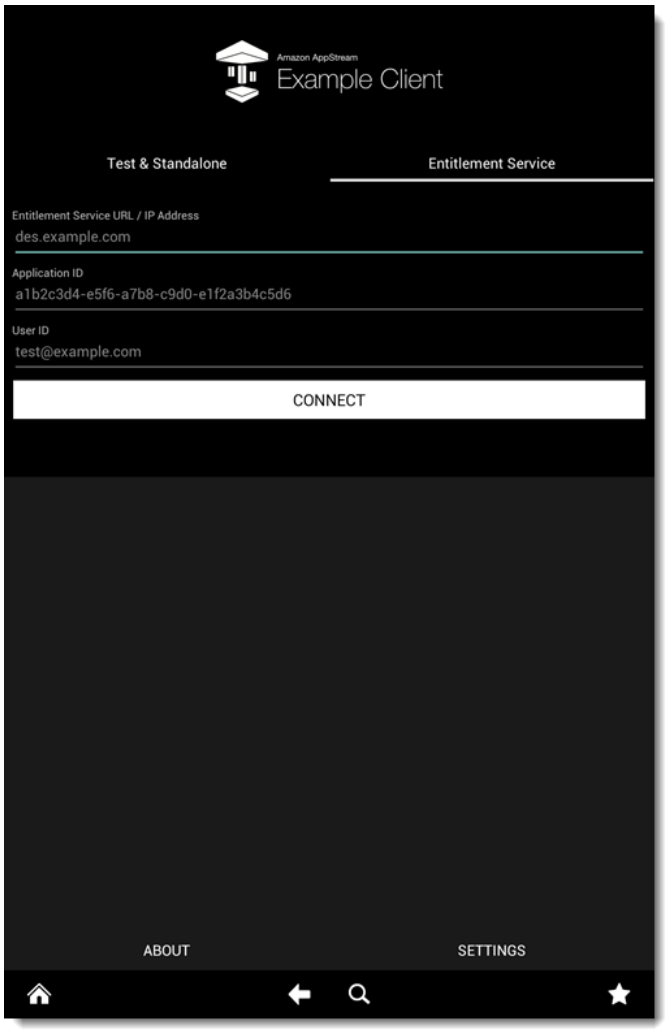
Changing the settings of the client application (p. 27) – change the settings to show the local mouse pointer, enable or disable hardware decoding, show the frame rate, and set the inactivity time.

### To install the Amazon AppStream Example Client

- Download and install the client application from the following links:
  - **Android device –** download the client application from Google play and install it.
  - **Fire OS device –** from your device, download the client application from http://amzn.com/axc. To launch the client application, tap **Install** from **Notifications** or **Downloads**.

### To stream an application through an entitlement service

1. Start the client application and choose **Entitlement Service**. Type the entitlement service URL or IP address, application ID, and user ID in the appropriate text boxes.



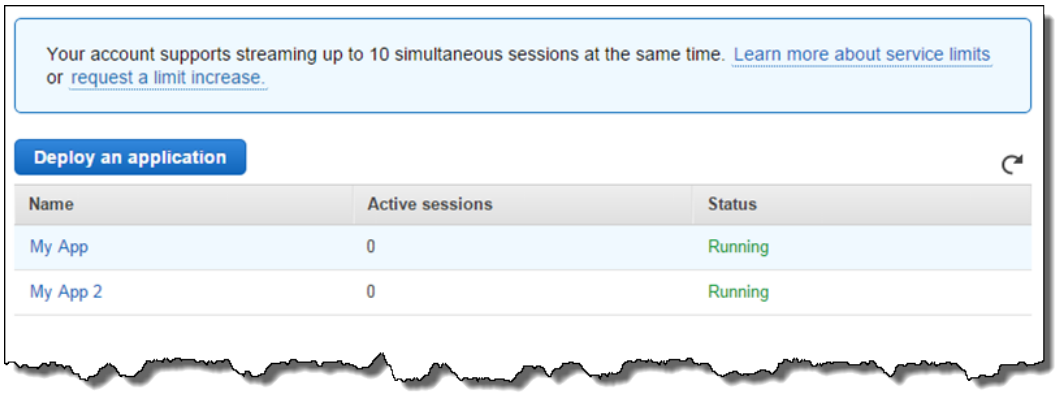

2. Click **Connect**.

### To stream from the Amazon AppStream console

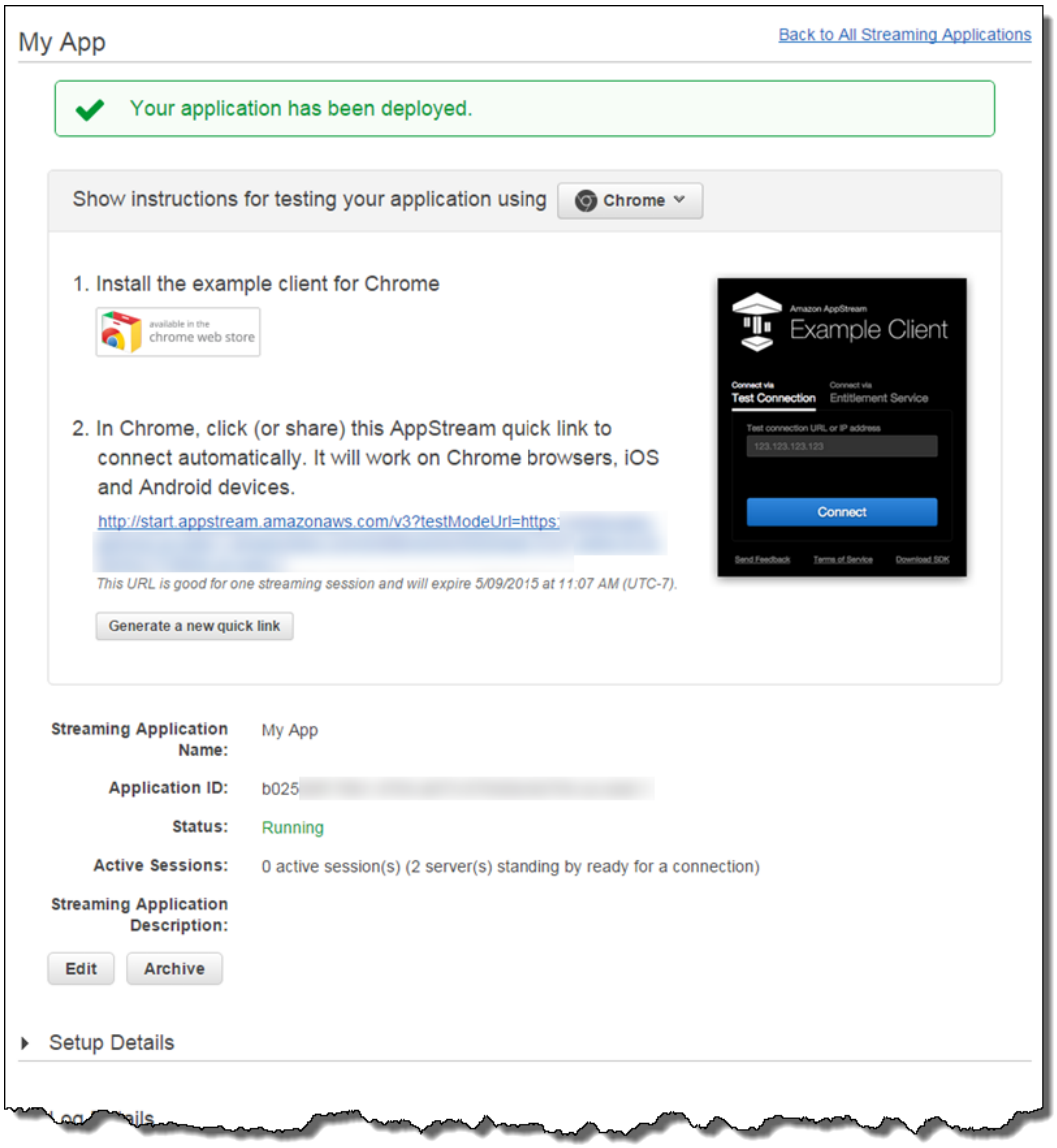
1. On your device, sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.

   Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.

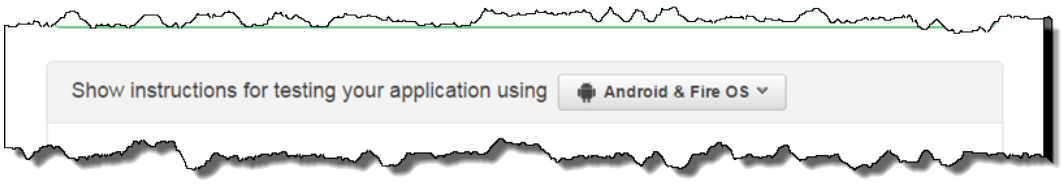

3. In the list of applications, click the name of the streaming application to view.



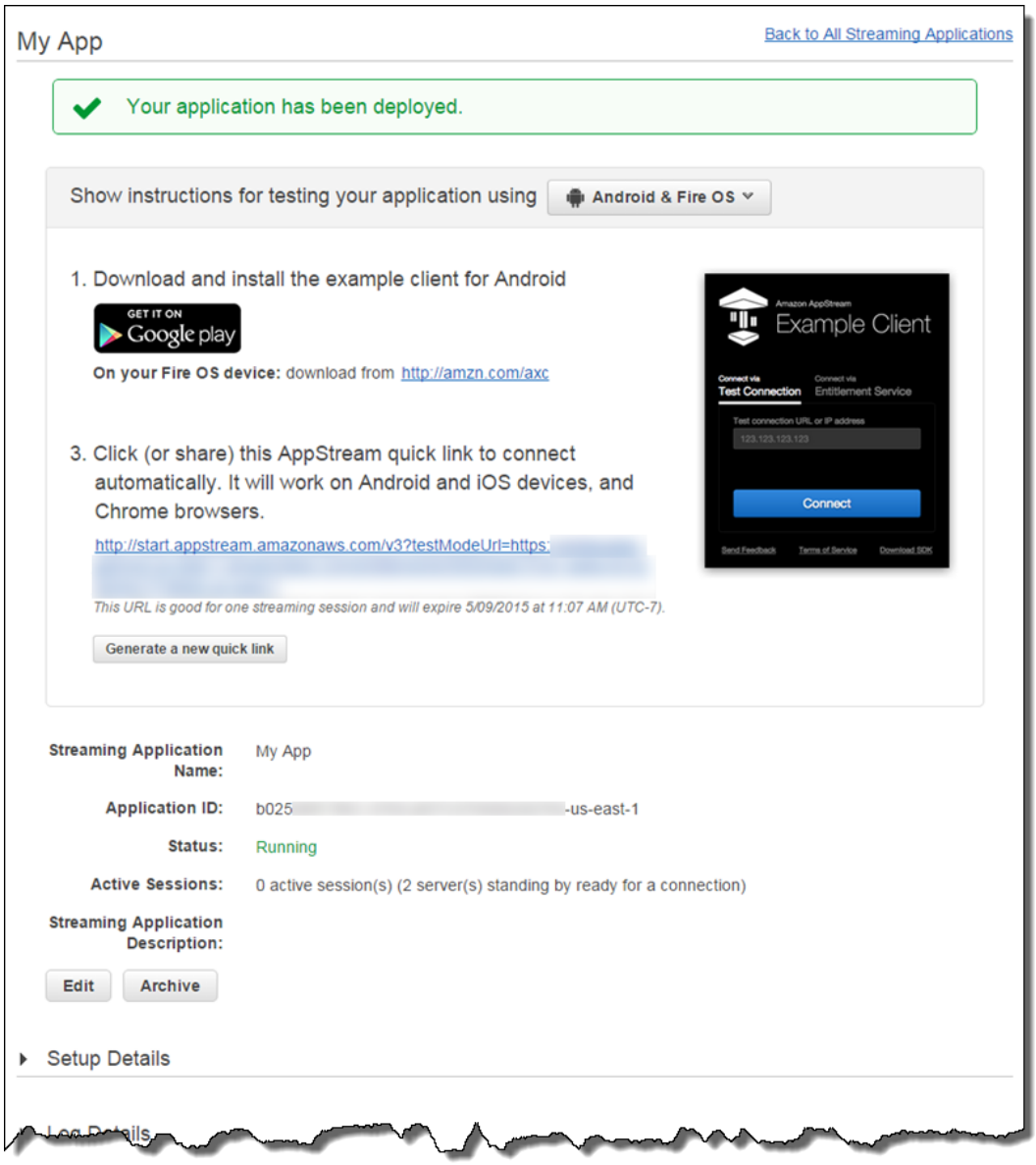

The application summary page displays information about the streaming application.




4. In **Show instructions for testing your application using**, select **Android and Fire OS**.




5. Click the link in step 2 to connect to your streaming application. The URL is valid for 72 hours and can only be used once. You can share this URL within the 72 hour period, but only one user can use this URL.

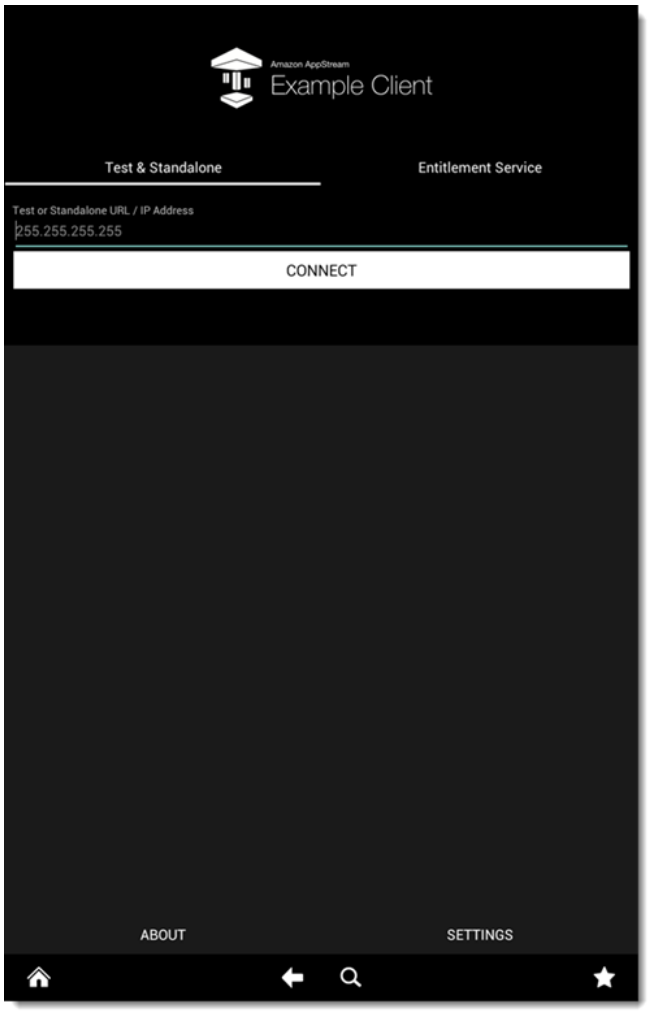



If you have already used this link, you can generate a new test connect URL. For more information, see Generating a New Test Connect Link to Stream Your Application (p. 56).

### To stream an application in standalone mode

Learn how to connect to the streaming application running in test mode.

1. Start the client application and choose **Connect via Test Mode**. In **Test Mode URL or IP address**, type the IP address of your EC2 instance.



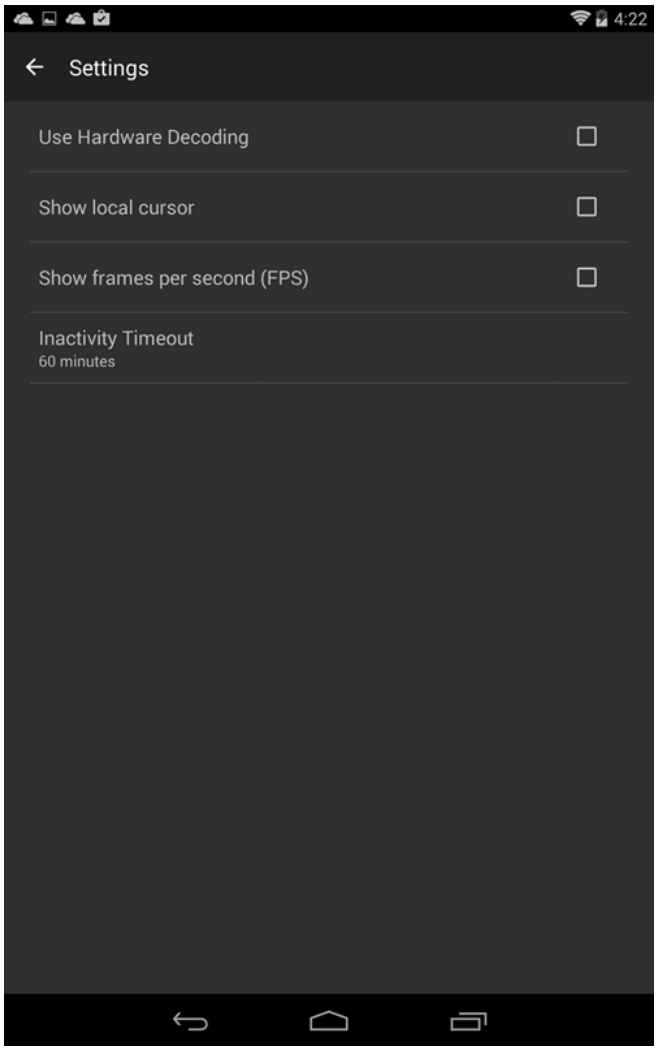

2. Click **Connect**.

### To change the settings of the client application

Learn how to change the settings of the Amazon AppStream Example Client.

1. Start the client application and choose **Settings**.
2. In **Settings**, do the following:

   - To use hardware decoding if the device is equipped with the hardware, select **Use Hardware Decoding**. If device cannot do hardware decoding, this option is disabled. If you experience performance issues, such as dropped frames or latency increases, disable this setting.
   - To show the mouse pointer only from the client application, select **Show local cursor**. When this check box is clear, the mouse pointers from both the client application and the server application appear.
   - To show the frame rate in frames per second (FPS), select **Show frames per second (FPS)**.
   - To change the length of inactivity time, select **Inactivity Timeout** and then specify the length of time. You are still incurring charges if a client application is connected to a streaming application but the client application is not sending any user inputs. Use this setting to reduce your costs when the client application is not sending user inputs.




# iPad, iPhone, or iPod Touch

The **Basic AppStream Client** is an application that you install on an iPad, iPhone, or iPod Touch.

This section contains the following procedures:

Installing the client application (p. 28) – shows how to install the **Basic AppStream Client** in a device.

Streaming an application through an entitlement service (p. 29) – use this method if you have the entitlement service URL, the application ID of the streaming application, and a user ID that can connect to the streaming application.

Streaming an application from the Amazon AppStream console (p. 29) – use this method to test your streaming application by using a link in the application summary page.

Stream an application in standalone mode (p. 32) – use this method for a streaming application in standalone mode.

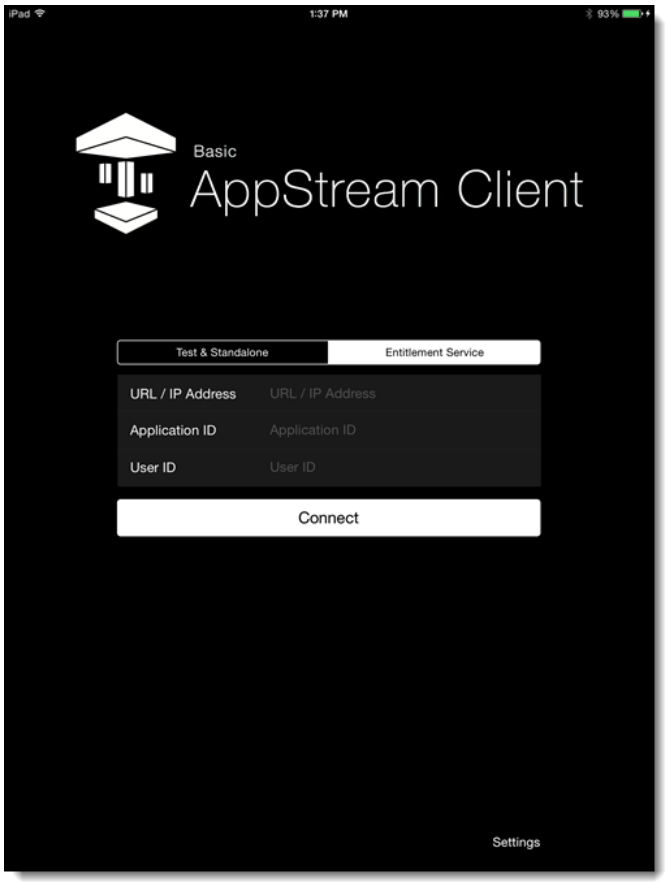
Changing the settings of the app (p. 33) – change the settings to show the frame rate and change the inactivity time.

### To install the Basic AppStream Client

- Go to https://itunes.apple.com/us/app/amazon-appstream-basic-client/id974559711?ls=1&mt=8 to download the **Basic AppStream Client** from the AppStore and install it.

### To stream an application through an entitlement service

1. Start the **Basic AppStream Client** and choose **Entitlement Service**. Type the entitlement service URL or IP address, application ID, and user ID in the appropriate text boxes.

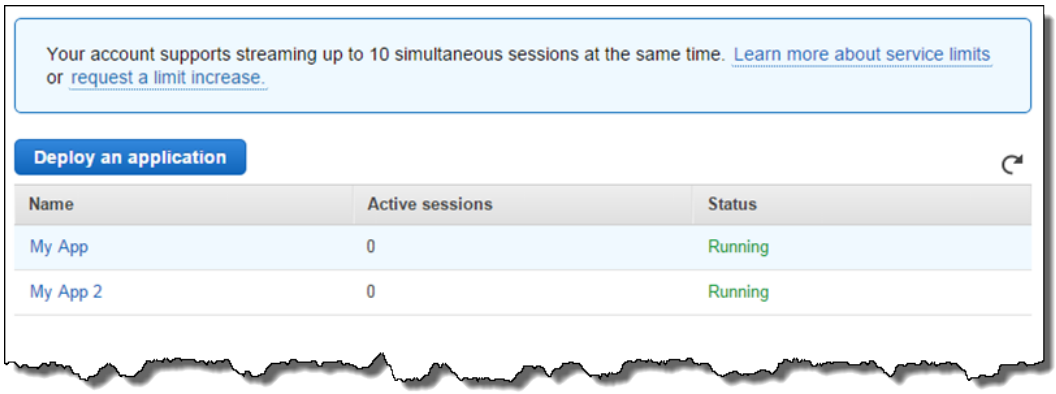



2. Click **Connect**.

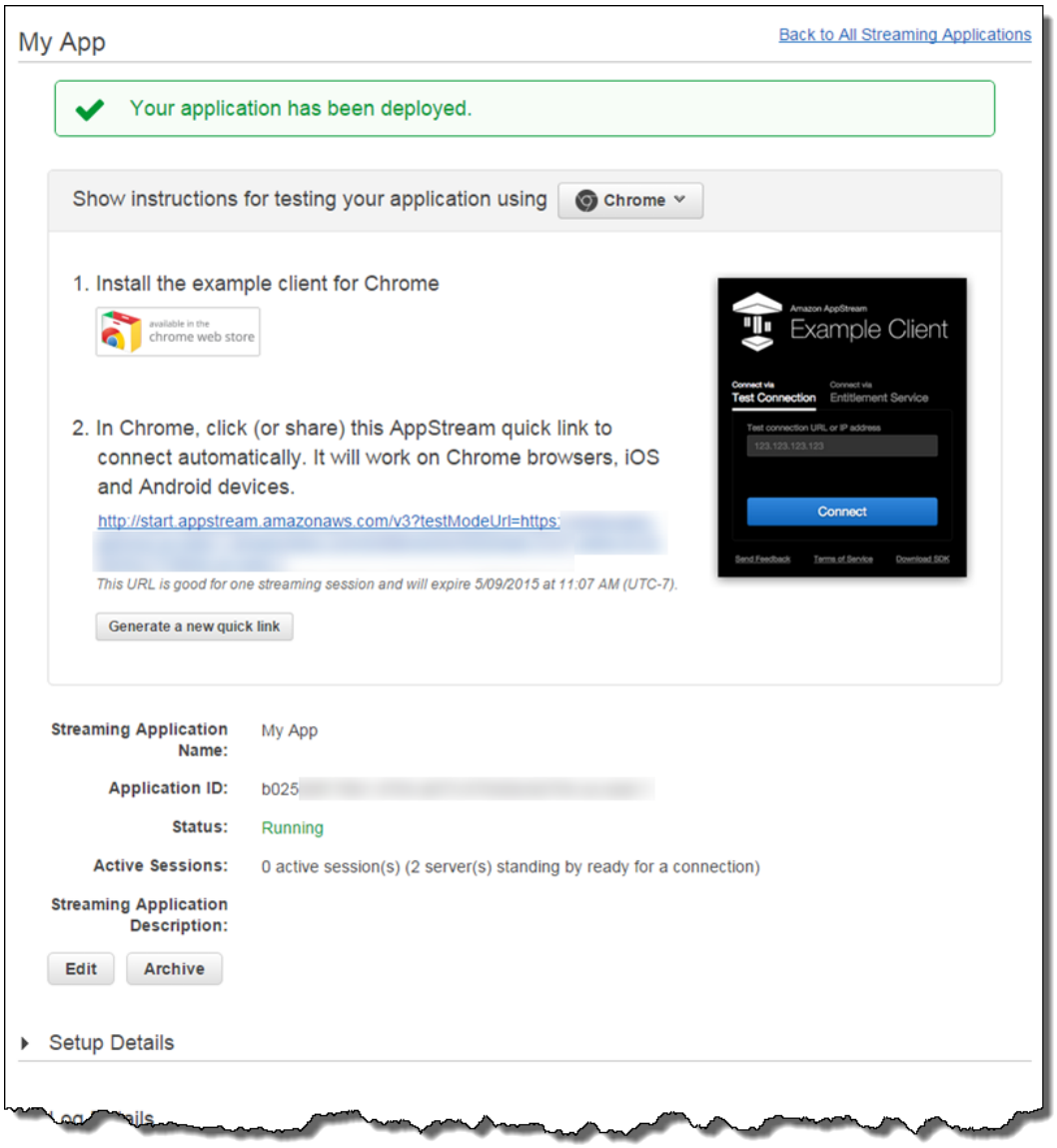
### To stream an application from the Amazon AppStream console

1. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.

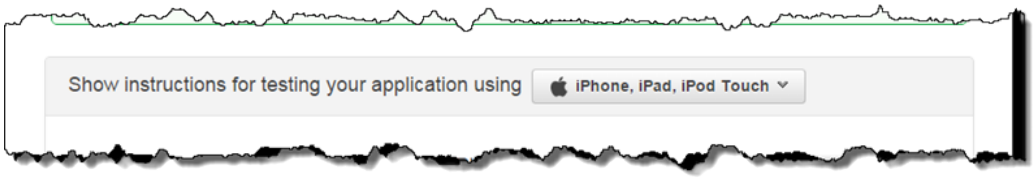

3. In the list of applications, click the name of the streaming application to view.



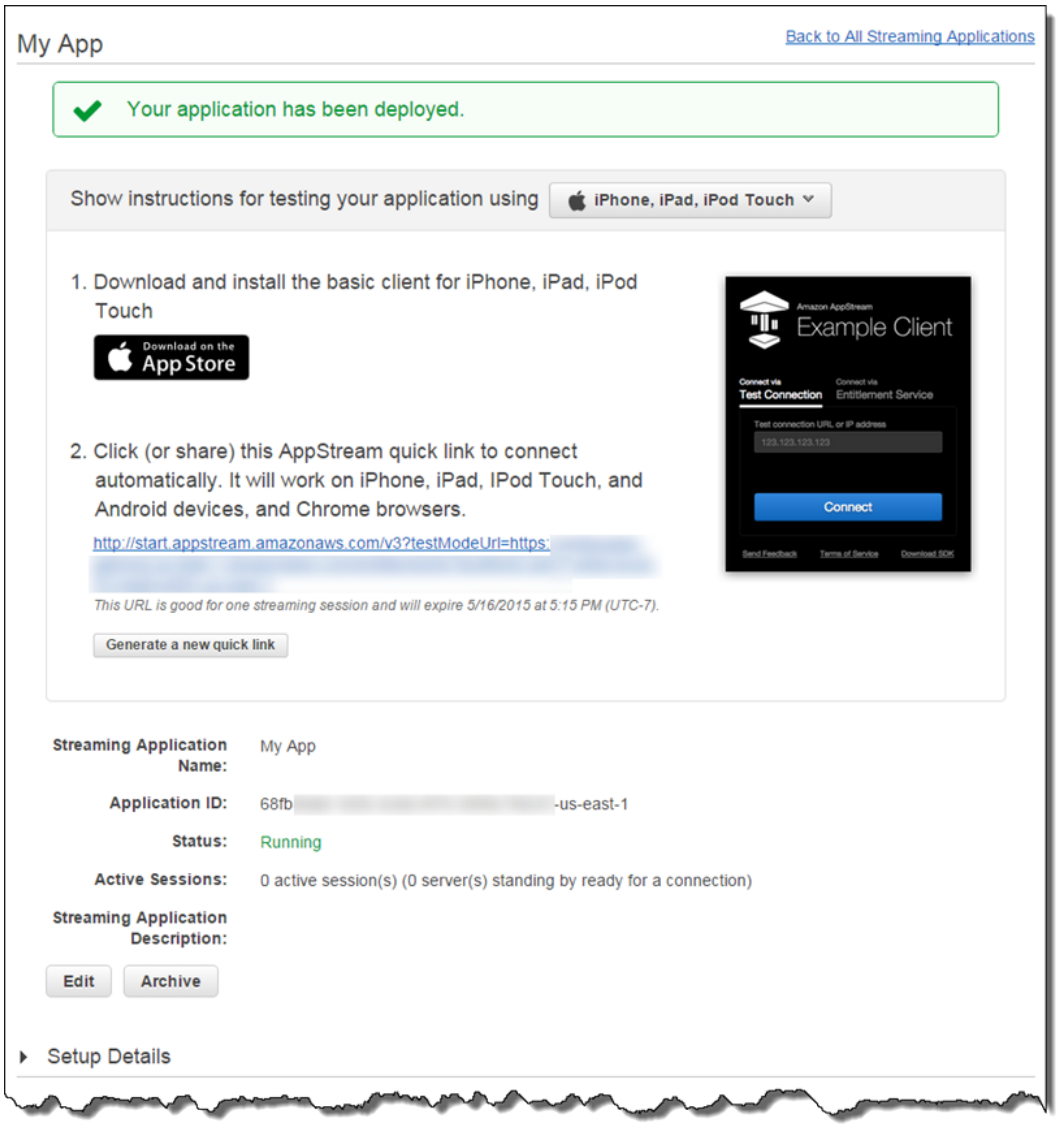

The application summary page displays information about the streaming application.




4. In **Show instructions for testing your application using**, select **iPhone, iPad, iPod Touch**.




5. Click the link in step 2 to start the **Basic AppStream Client** and connect to your streaming application. The URL is valid for 72 hours and can only be used once. You can share this URL within the 72 hour period, but only one user can use this URL.




If you have already used this link, you can generate a new test connect URL. For more information, see Generating a New Test Connect Link to Stream Your Application (p. 56).

### To stream an application in standalone mode

1. Start the **Basic AppStream Client** and choose **Test & Standalone**. In **URL / IP address**, type the IP address of your EC2 instance.


2. Click **Connect**.

### To change the settings of the Basic AppStream Client

1. Start the client application and choose **Settings**.
2. In **Settings**, do the following:

   - To show the frame rate in frames per second, select **Show frames per second**.
   - To set the number of minutes of inactivity before the client application shuts down, Choose **Inactivity timeout** and select a time. You are still incurring charges if a client application is connected to a streaming application but the client application is not sending any user inputs. You can reduce your cost by choosing an appropriate inactivity time.

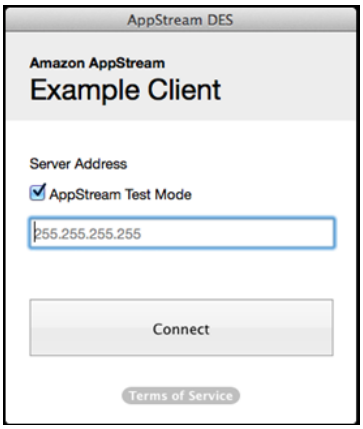

# OS X

### To stream to the OS X client application

1. Download the client application from https://s3.amazonaws.com/appstream-clients/latest/appstreamclient-osx.zip and install it on your device.
2. Start the **Amazon AppStream Example Client**.
3. Do one of the following:

   - To connect to an entitlement service and stream from Amazon AppStream, type the entitlement service URL or IP address, application ID, and user ID in the appropriate text boxes.


- To stream an application running in standalone mode or test connect mode, click **AppStream Test Mode** and type the EC2 instance or test connect URL.




4. Click **Connect**.

# Windows

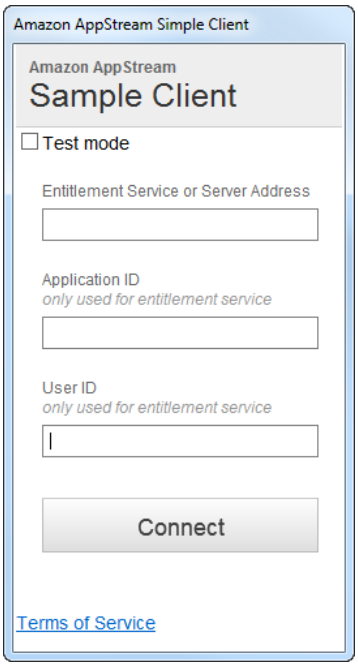
### To stream to the Windows client application

1. Install the following dependency files:

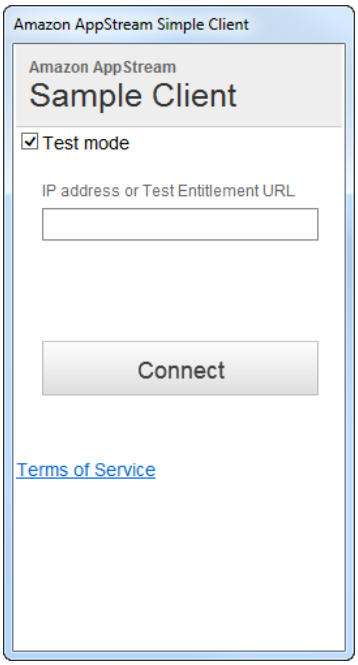
    1. Download the DirectX Software Development Kit from the Microsoft Download Center and install it.
    2. Download the Visual C++ Redistributable for Visual Studio 2012 from the Microsoft Download Center and install it.

2. Download the client application from https://s3.amazonaws.com/appstream-clients/latest/appstreamclient-windows.zip and install it on your device.
3. Start the **Amazon AppStream Example Client**.
4. Do one of the following:

   - To connect to an entitlement service and stream from Amazon AppStream, type the entitlement service URL or IP address, application ID, and user ID in the appropriate text boxes.




   - To stream from an application running in standalone or test connect mode, click **Connect to Standalone Instance**. In **Standalone instance URL or IP address**, type the EC2 instance or test connect URL.




5. Click **Connect**.

# Managing Your Streaming Application

This chapter contains topics on preparing and managing your streaming application.

**Topics**


- Deploying Your Streaming Application to Amazon AppStream (p. 38)
- Displaying Mouse Pointers (p. 53)
- Generating a New Test Connect Link to Stream Your Application (p. 56)
- Viewing All Applications (p. 57)
- Viewing Application Summary (p. 58)
- Editing an Application (p. 60)
- Archiving an Application (p. 63)
- Enabling Logging on a Streaming Application (p. 66)
- Increasing Your Amazon AppStream Service Limits (p. 71)


## Deploying Your Streaming Application to Amazon AppStream

You can use the interactive deployment wizard or the advanced wizard to deploy your streaming application to Amazon AppStream. The wizard that you use depends on your browser and application installer.

The following table shows the requirements for each deployment wizard:

| | Interactive | Advanced |
|---|---|---|
| Browser | Google Chrome<br>Mozilla Firefox | Apple Safari<br>Google Chrome<br>Internet Explorer<br>Mozilla Firefox |

| | Interactive | Advanced |
|---|---|---|
| Application installer | Any installer | Silent installer. For more information, see Build an Application Installer (p. 192) |
| Installer location | Any file hosting service such as Amazon Cloud Drive or Dropbox | presigned URL to an Amazon S3 bucket. For more information, see Generate a presigned URL to your Application Installer (p. 42) |

**Topics**


- Deploying with the Interactive Deployment Wizard (p. 39)
- Deploying with the Advanced Deployment Wizard (p. 42)

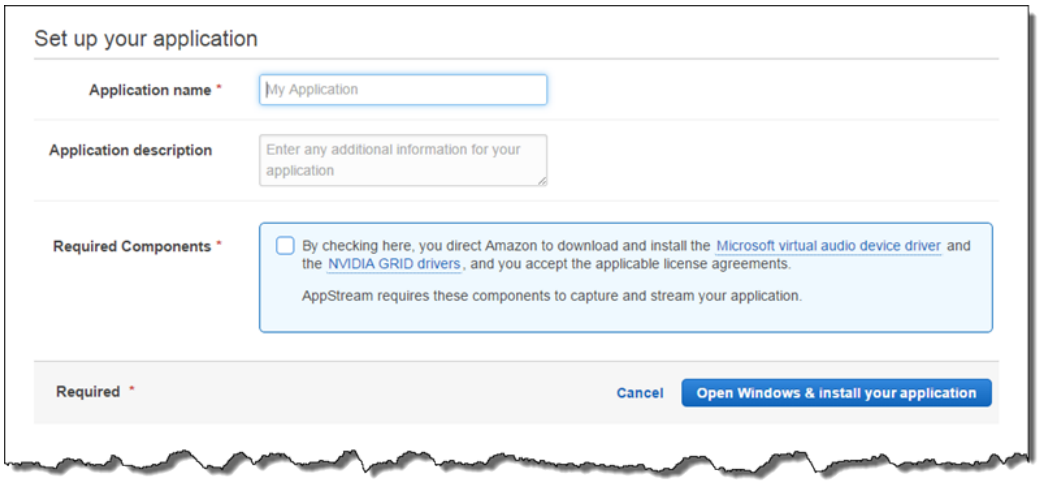

# Deploying with the Interactive Deployment Wizard

You will need the following to deploy your streaming application using the interactive deployment wizard:

- Google Chrome or Mozilla Firefox web browser
- The installer file for your streaming application. Upload your installer file to a file hosting service such as Amazon Cloud Drive or Dropbox.
- Installers for dependency files or data files that are required by your streaming application. Upload these files to a file hosting service such as Amazon Cloud Drive or Dropbox.

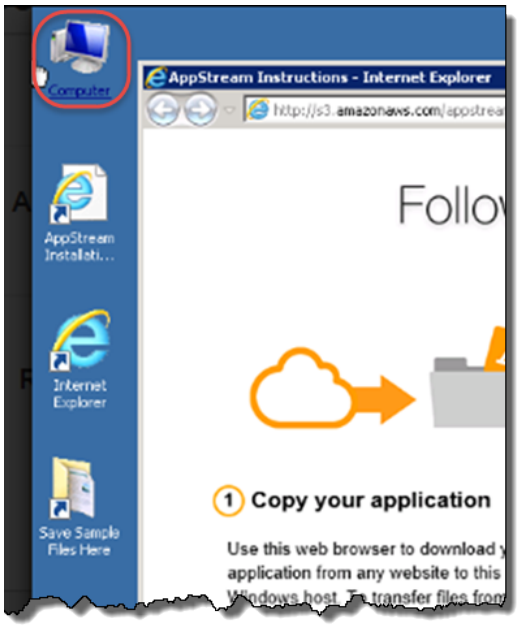
### To deploy your application to Amazon AppStream with the Interactive Deployment Wizard

1. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. Click **Deploy an application**.
3. In **Set up your application**, do the following:
   - For **Application name**, type the name of your streaming application. This text identifies your streaming application in the console.
   - For **Application description**, type a short description of the streaming application. This text appears in the application summary page.
   - Select **Required Components**.
   - Click **Open Windows & install your application** to display the Windows live host desktop.




4. Download the installer for your streaming application to the Windows live host desktop from the file hosting service.
5. Start your installers and install your application and dependency files.
6. If you do not have to set environment variables, skip to the next step. To set environment variables required by your streaming application, do the following:

    1. In the live Windows host desktop, click **Computer**.



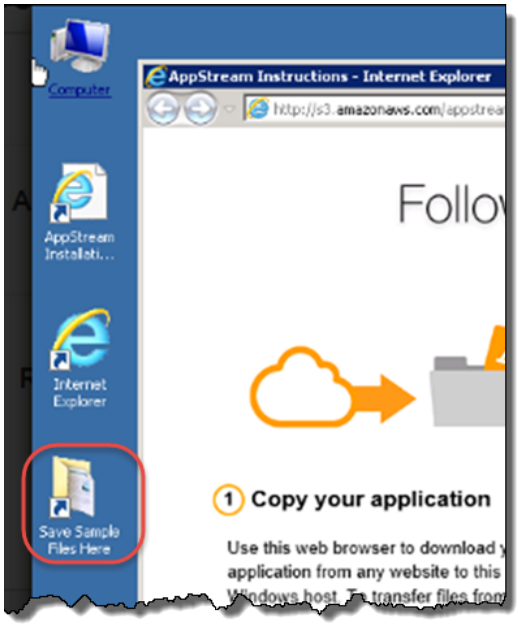

    2. Navigate to the `C:\Windows\System32` folder.
    3. Click `cmd.exe`.
    4. In the command prompt, type **`setx`** ***`<environment variable>`*** where *`<environment variable>`* is the environment variable and setting. For example, to configure Amazon AppStream to use the mouse pointer of the streaming application, type the following:

        ```
        setx VDI_CAPTUREMOUSECURSOR true /m
        ```

5. Press Enter.
6. Type **`exit`** and press Enter to close the command prompt.

7. If your streaming application does not require additional files, skip to the next step. To add other files required by your streaming application, do the following:

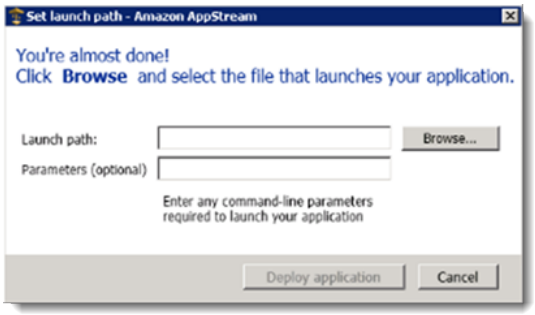
   1. Download your files to the Windows live host desktop. Double-click **Save Sample Files Here** to open the **My Documents** folder.




   2. Move or copy your files to the **My Documents** folder.

8. Click **Set Launch Path**.
9. In **Set launch path - Amazon AppStream**, do the following:

   - For **Launch path**, click **Browse**, navigate to the folder, and then select your executable file.
   - For **Parameters (optional)**, type the parameters required to start your streaming application.
   - Click **Deploy application**.




10. Follow the on-screen instructions.

# Deploying with the Advanced Deployment Wizard

You will need the following to deploy your streaming application using the advanced wizard:

- A supported browser
- An application that can be installed using a compatible silent installer (see Build an Application Installer (p. 192))
- An Amazon Simple Storage Service (Amazon S3) bucket that contains your application installer
- Visual Studio 2010 or later to use the AWS Toolkit for Visual Studio
- AWS Toolkit for Visual Studio to create a presigned URL to your application installer

You will do these three tasks:

- Upload your application installer to Amazon S3 (p. 42)
- Generate a presigned URL to your Application Installer (p. 42)
- Deploy your Streaming Application to Amazon AppStream (p. 47)

## Upload your application installer to Amazon S3

Before your can upload your application to an Amazon S3 bucket, you must first create the bucket. If you already have an Amazon S3 bucket, skip toTo Upload the Application Installer to Amazon S3 (p. 42).

### To create an Amazon S3 bucket

1. In the Amazon S3 console, click **Create Bucket**.
2. In the **Create a Bucket—Select a Bucket Name and Region** dialog box, type a name in the **Bucket Name** box.

   The bucket name you choose must be unique across all existing bucket names in Amazon S3. Your bucket name must be between 3 and 63 characters long, composed of lowercase letters and numbers.

   **Important**
   After you create a bucket, you cannot change the bucket name.
3. Click **Create**.

   When Amazon S3 successfully creates your bucket, the console displays the properties of your empty bucket. This is the bucket where you upload the application installer.

### To Upload the Application Installer to Amazon S3

1. On the Amazon S3 console, click your bucket name, and then click **Upload**.
2. In the **Upload—Select Files and Folders** dialog box, click **Add Files**.
3. In the **File Upload** dialog box, select the application installer that you downloaded and saved to your local drive.
4. Click **Open**.
5. Click **Start Upload**.

## Generate a presigned URL to your Application Installer

After you have uploaded the application installer to Amazon S3, you need to create a presigned URL that points to the file in the bucket. A presigned URL is a special URL that allows Amazon AppStream to

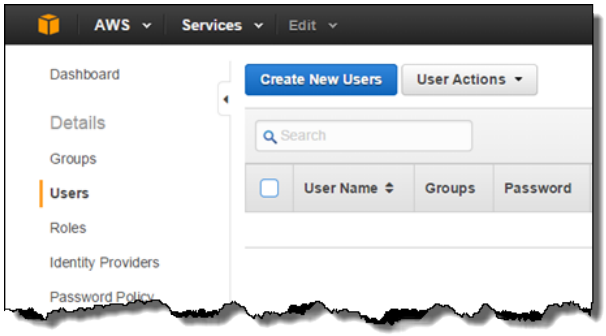
download and run your application installer without exposing any security credentials. This URL is valid for a specific time period only.

To generate a presigned URL, you need to create a user with the proper permissions in AWS Identity and Access Management.

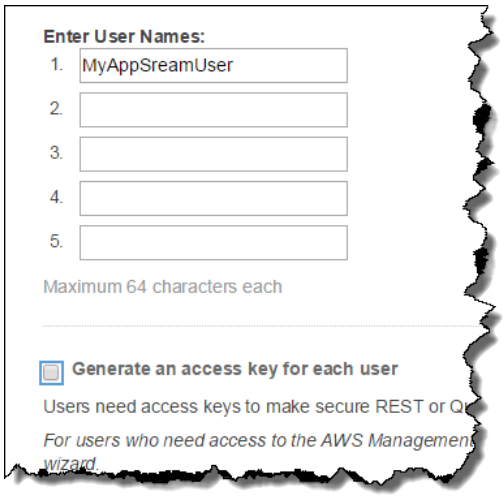
After creating the user, you need to generate a presigned URL to the application installer. You can use AWS Toolkit for Visual Studio to generate the presigned URL.

### To create the user

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Users** and then click **Create New Users**.

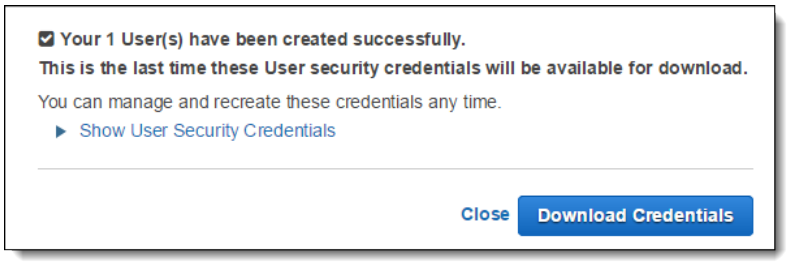



3. For **Enter User Names**, type a name for the user that will install the streaming application. Select **Generate an access key for each user** and then click **Create**.

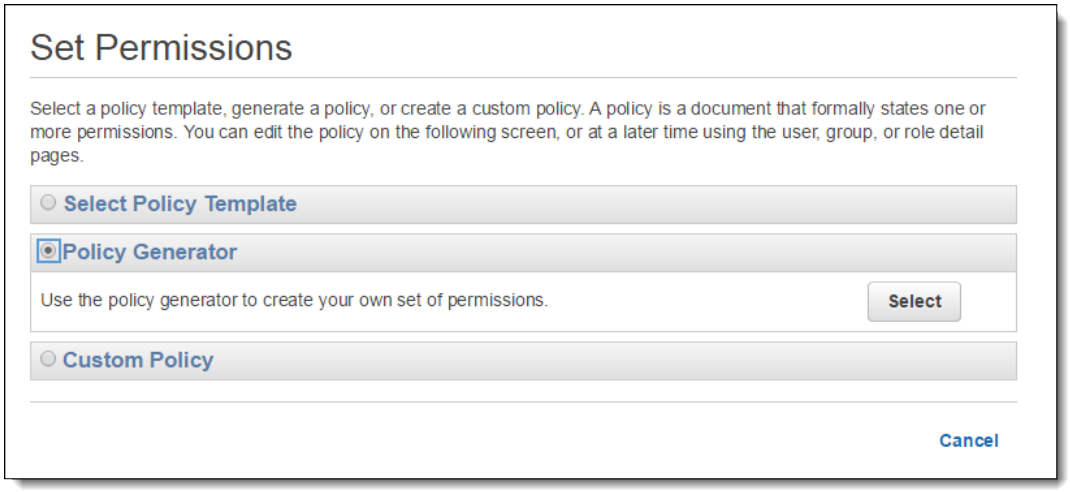



4. In the **Create User** dialog box, click **Download Credentials** and then save the file to a safe place. The comma separated value file contains the access key and secret key that you will need to generate a presigned URL.

   **Note**
   Store user credentials in a secure place. This is the only time you can get the secret key. If you lose this information, you must create a new access key and secret key.

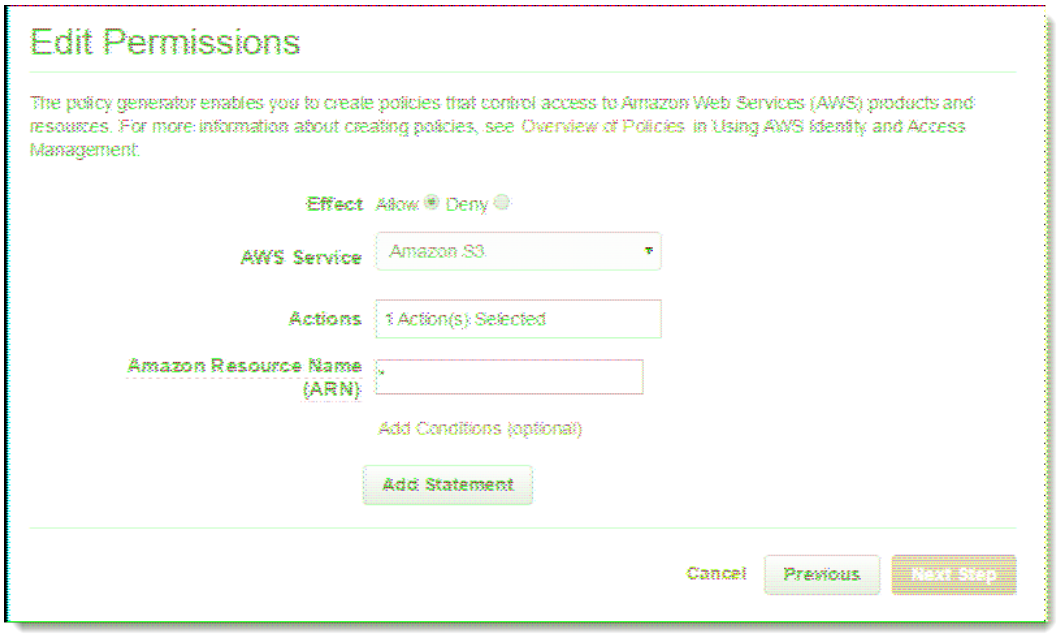



5. Click **Close**.
6. On the console, double-click your new user.
7. Under **Permissions**, click **Attach User Policy**.
8. In **Set Permissions**, select **Policy Generator**, and then click **Select** .




9. In **Edit Permissions**, do the following:

   - For **Effect**, click **Allow**
   - For **AWS Service**, select **Amazon S3**.
   - For **Actions**, select **ListAllMyBuckets**.
   - For **Amazon Resource Name (ARN)**, type an asterisk (*****).
   - Click **Add Statment** and then click **Next Step**.




10. In **Set Permissions**, click **Apply Policy**.


### To install the AWS Toolkit for Visual Studio

1. Go to AWS Toolkit for Visual Studio and click **AWS Toolkit for Visual Studio**.
2. Run the installation wizard, which is packaged as an `.msi` file.

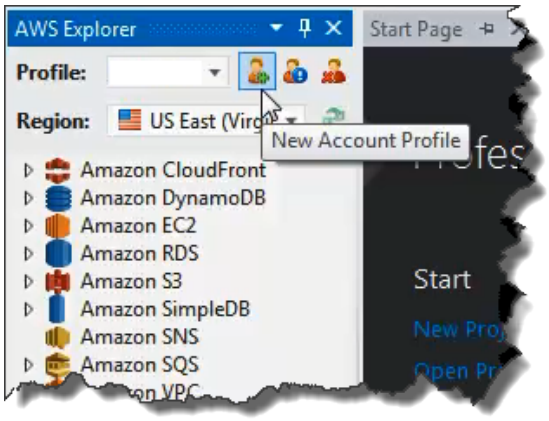
   - If your browser asks whether to save or run the `msi`, select **Run**.
   - If your browser automatically saves the `.msi` file to your system, navigate to the download directory and use Windows Explorer to launch the `.msi`.

The MSI file name depends on the version, but it will look something like `AWSToolsAndSDKForNet_sdk-2.0.15.0-ps-2.0.15.0-tk-1.8.0.2.msi`.

3. Follow the installation wizard's instructions to install the toolkit.

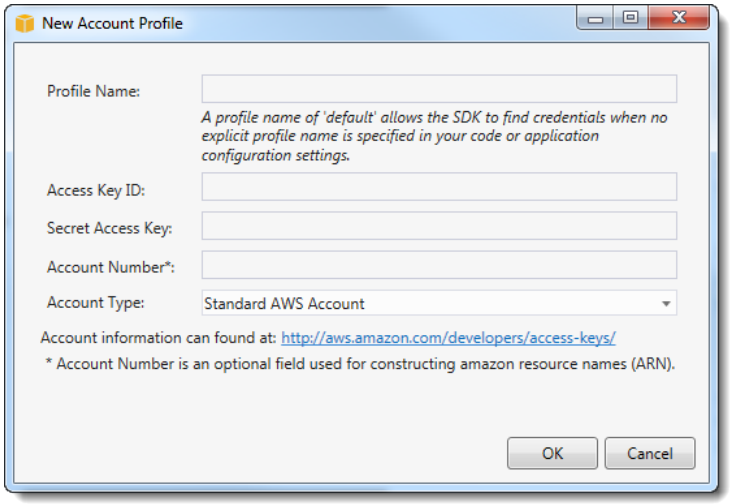
### To add a profile to the Toolkit for Visual Studio

1. In Visual Studio, open **AWS Explorer** by clicking the **View** menu and selecting **AWS Explorer**.
2. Select the region with your Amazon S3 bucket.
3. Click the new account profile icon to the right of the **Profile** list.




4. In the **New Account Profile** dialog box, do the following:

   - For **Profile Name**, type a name to identify the profile.
   - For **Access Key ID**, type the access key ID of the user with access to the Amazon S3 bucket with your application installer.
   - For **Secret Access Key**, type the secret key of the user with access to the Amazon S3 bucket with your application installer.
   - Click **OK**.



### To generate a presigned URL

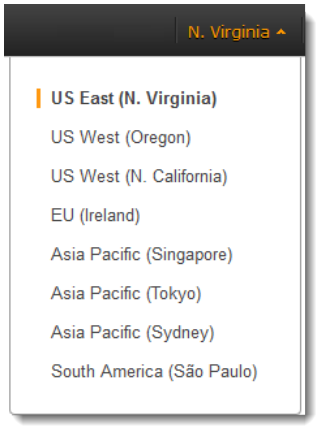
1. Select the region with your Amazon S3 bucket.
2. Expand the **Amazon S3** node and click your bucket name.
3. Right-click the application installer and then select **Create Pre-Signed URL**.
4. In the **Create Pre-Signed URL** dialog box, set an expiration date and time for the URL or go to the next step if you want to use the default setting which is one hour from the current time.
5. Click the **Generate** button.
6. Click **Copy** to copy the presigned URL to the clipboard. Save this URL to a file.

## Deploy your Streaming Application to Amazon AppStream

After uploading the application installer to Amazon S3 and then generating a presigned URL, you are ready to add your application to Amazon AppStream.

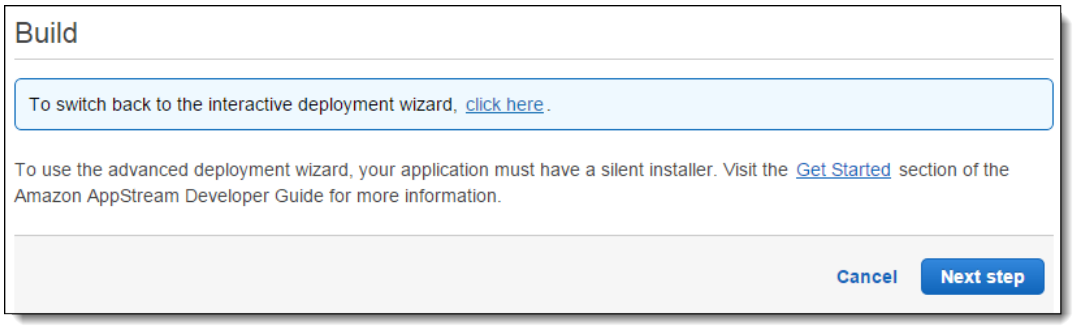
To deploy your streaming application, do the following:

1. Open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. From the region selector of the top navigation bar, select the region where you want to deploy your streaming application.



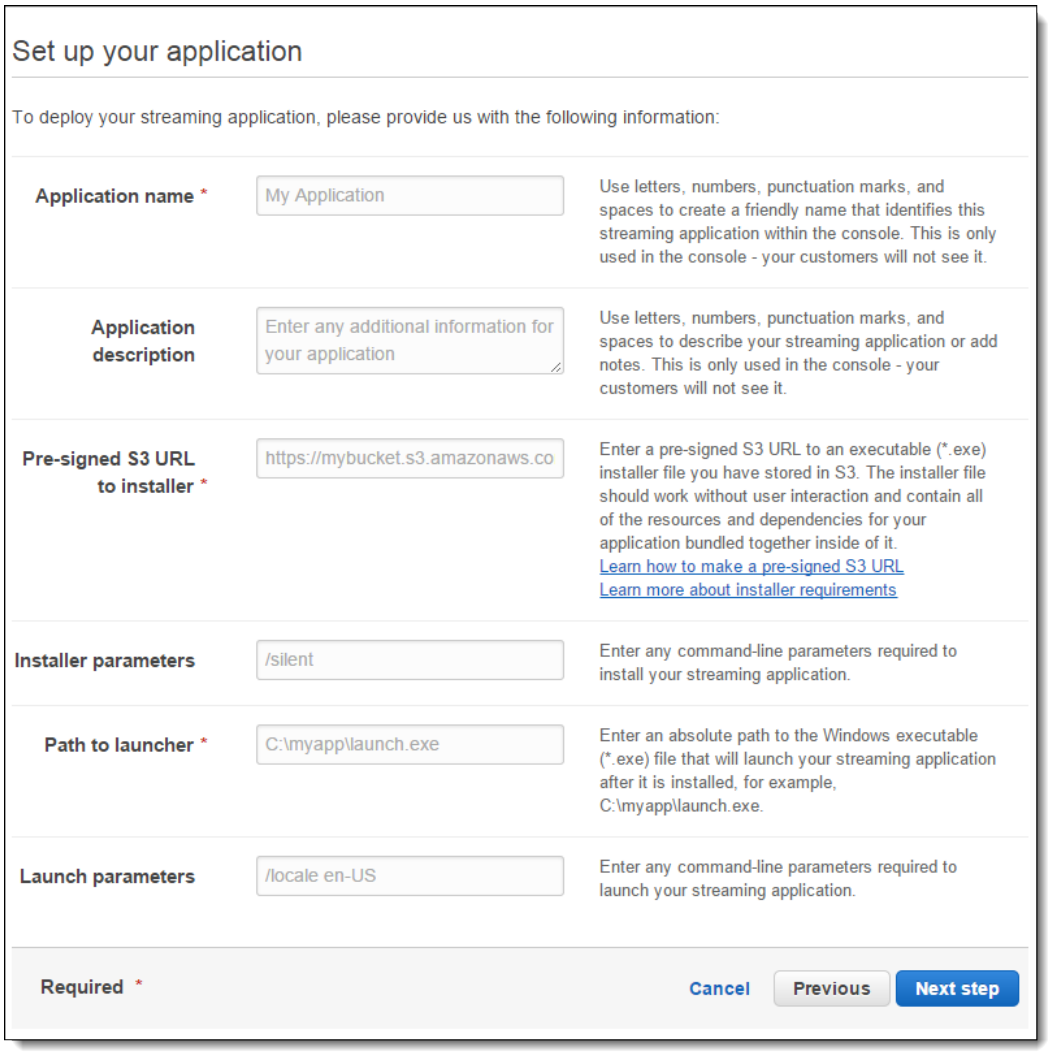

3. Click **Deploy an application**.
4. If you are using Google Chrome or Mozilla Firefox, switch to the advanced deployment wizard. For other browsers, skip to the next step.


5. On the **Build** page, click **Next step**.




6. On the **Set up your application** page, do the following and then click **Next step**:

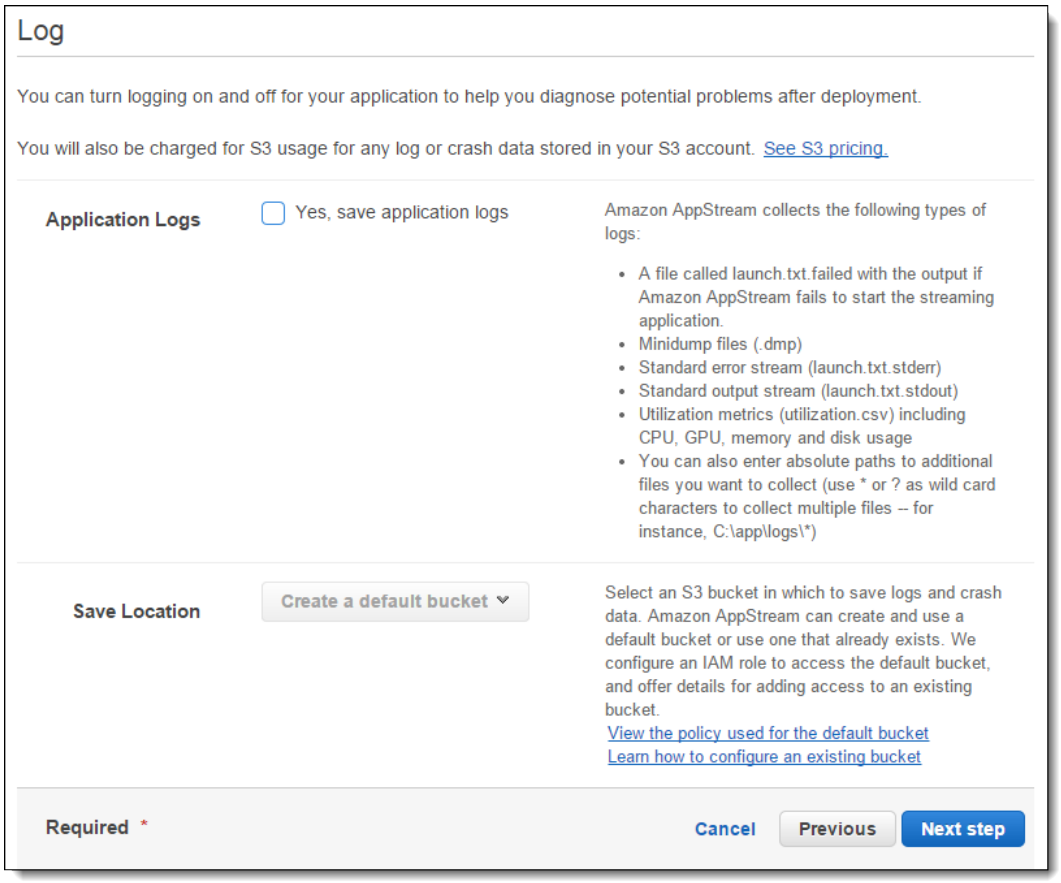
   - For **Application name**, type the name of your streaming application. This text identifies your streaming application in the console.
   - For **Application description**, type a short description of the streaming application. This text appears in the application summary page.
   - For **Pre-signed S3 URL to installer**, type the URL you created in the previous step.
   - For **Installer Parameters**, enter any command-line parameters required to run the installer.
   - For **Path to launcher**, type the absolute path to the executable file that starts your streaming application.
   - For **Launcher parameters**, enter any command-line parameters required to start your streaming application on Amazon AppStream.




7. You can enable logging on the **Log** page in **Application logs** or click **Next step** to continue deploying your streaming application.

   If you want to enable logging on your streaming application, select **Yes, save application logs**. After you select the check box, a text box appears below so that you can optionally specify another path to a directory that contains other log files or a path to a log file that will be added to the other log files. You can specify more than one path or file by filling out the other text box that appears after you fill in the text box. For more information about logging, see Enabling Logging on a Streaming Application (p. 66).

   After specifying the paths and log files, click **Next step**.




The custom log files and paths must meet the following requirements:

- Log files must be located in the `C:\` directory.
- Log paths must be fully qualified paths. Do not use relative paths. You can use wildcards.
- Log paths must be less than the path length limits. See Naming Files, Paths, and Namespaces.
- Log file names cannot contain Windows reserved characters. See Naming Conventions.
- Your application must have read access to the paths.

> **Note**
> The total size of logs in the `.zip` file is subject to the Amazon S3 key size limits.

For **Save Location**, you can choose to have Amazon AppStream create an Amazon S3 bucket. To use a bucket created by Amazon AppStream, click **Create a default bucket**. To use one of your existing Amazon S3 buckets, click the down arrow next to **Create a default bucket** and select one of your buckets.

> **Note**
> Your bucket must allow Amazon AppStream to use the `PutObject` method. You can allow this method by using a bucket policy similar to the following:

```
{  "Version": "2008-10-17",
   "Id": "Policy1396472828471",
   "Statement": [
      {
```

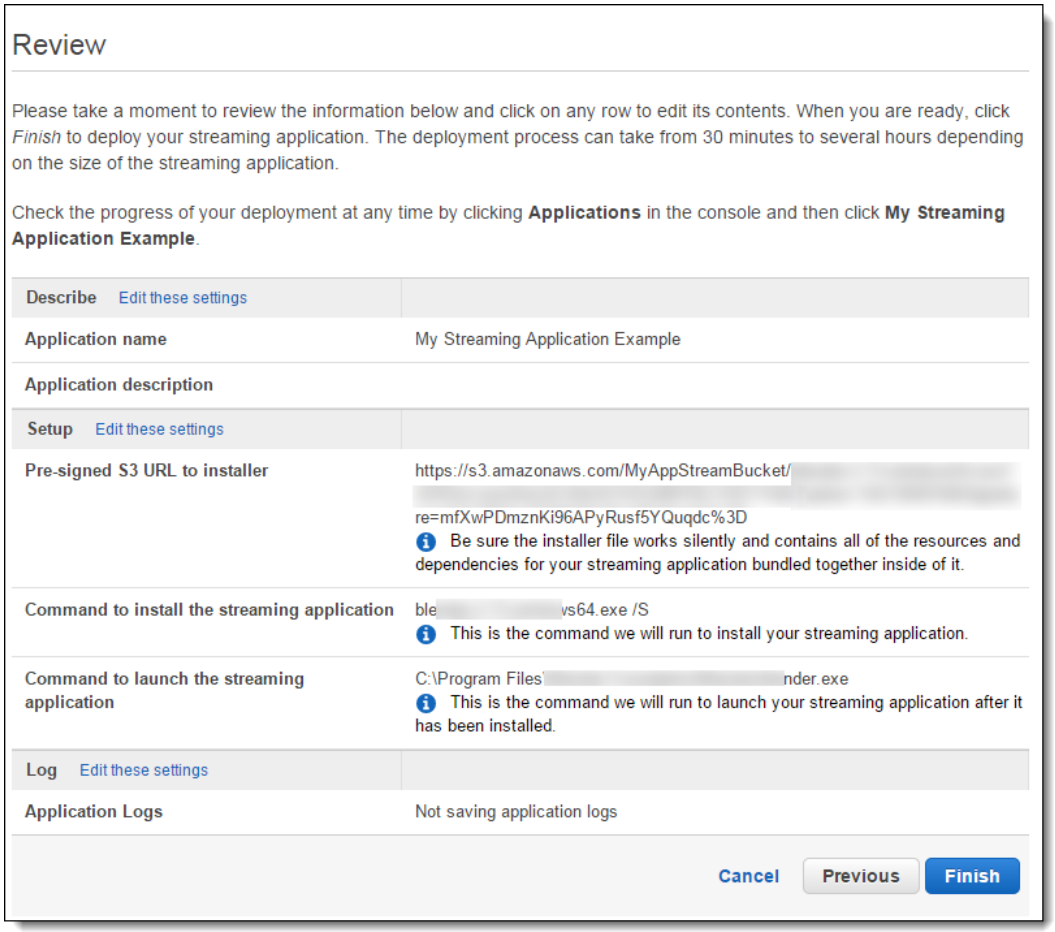
```
            "Sid": "Stmt1396472820469",
            "Effect": "Allow",
            "Principal": {"AWS":"arn:aws:iam::990116983315:root"},
            "Action": ["s3:PutObject"],
            "Resource": "arn:aws:s3:::your bucket name/*"
        }
    ]
}
```

For more information about modifying the Amazon S3 bucket policy, see Bucket Policy Examples in the Amazon S3 Developer Guide.

The log files are saved to a `.zip` file which is then uploaded to the bucket. Logs are saved with the following filename:

*<bucketName>*/AppStream/*<region>*/*<Application ID>*/yyyy-mm-dd/*<log file name>*.zip

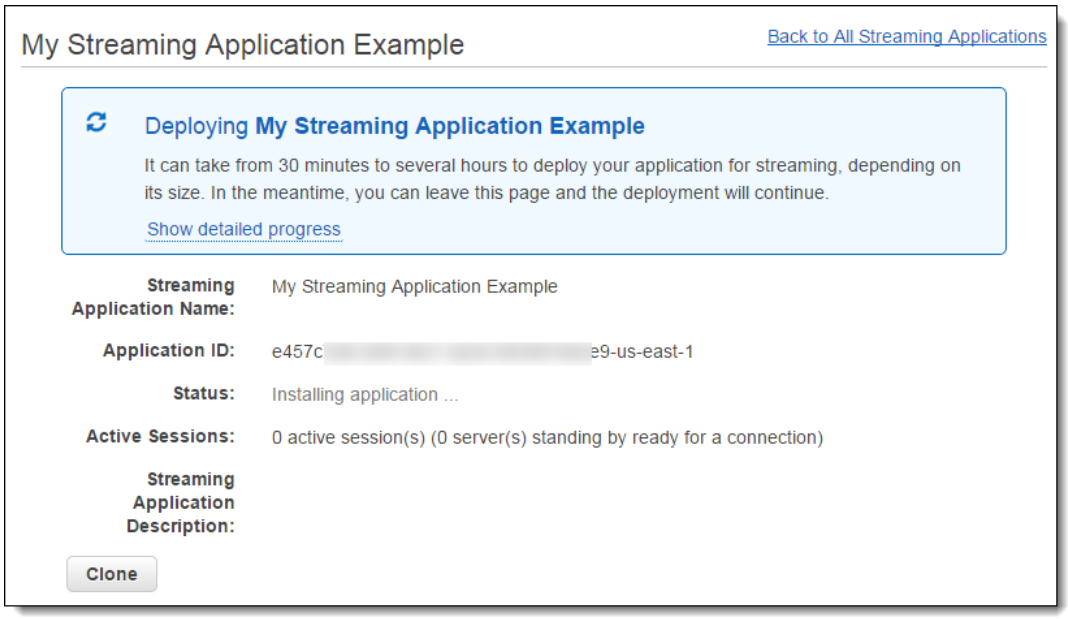
8. On the **Review** page, check that the values are correct. If the values are correct, then click **Finish** to add your streaming application to Amazon AppStream. If you need to change a value, click **Previous** until you return to page where you can change the value.



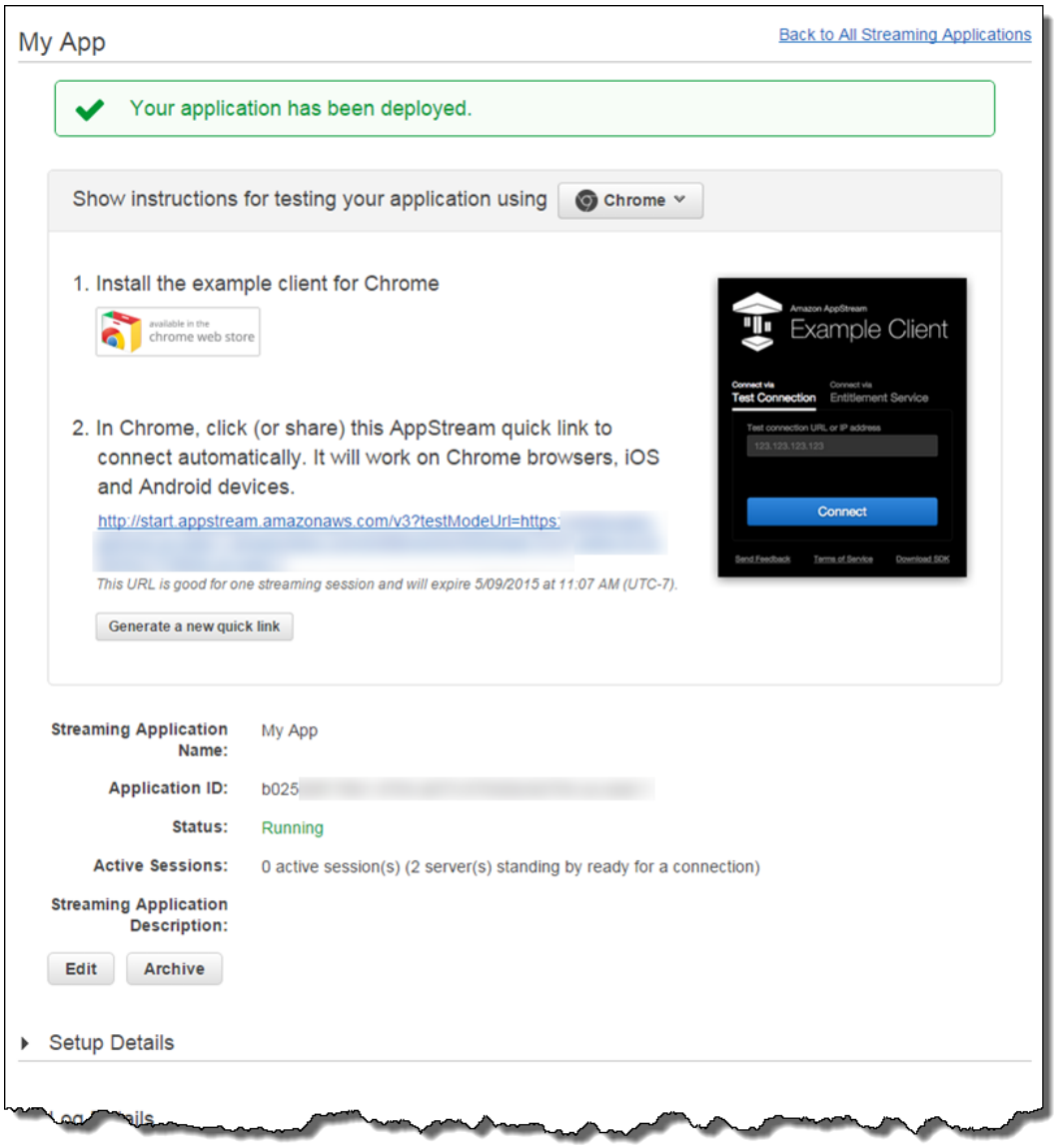
Wait while Amazon AppStream prepares your application. This may take 30 minutes or more while Amazon AppStream performs the following tasks:

- Copies your application installer from your Amazon S3 bucket.
- Prepares your Amazon AppStream environment.
- Installs your streaming application on an Amazon AppStream.
- Creates an Amazon Machine Image (AMI) of the server configuration that includes your installed application.
- Starts your streaming application.

While Amazon AppStream is deploying your application, it displays the **Application Summary** page, which contains the metadata for your application. The **Application ID** line shows the identifier assigned to your application. Client applications specify this identifier when they call into your entitlement service to connect to your application.




When your deployment is finished, Amazon AppStream displays a message indicating whether the deployment succeeded or failed.




# Displaying Mouse Pointers

Because your streaming application runs on a Windows instance running on Amazon EC2 and your users interact with your streaming application using a client application, you have two mouse pointers to consider:

- The mouse pointer of the streaming application.
- The mouse pointer of the client application.

This topic explains how to configure the remote and local mouse pointers.

**Topics**


- Remote mouse pointer (p. 54)
- Local mouse pointer (p. 54)

# Remote mouse pointer

The remote mouse pointer is the mouse pointer of the streaming application that runs on a Windows instance running on Amazon EC2. When you deploy your streaming application, the remote mouse pointer is disabled. If the remote mouse pointer is displayed, your users may get the perception of sluggish performance. The perception may occur because the user input from the client application must go through the network before the streaming application can process the input.

However, some streaming applications may require the remote mouse pointer. For example, your streaming application uses the remote mouse pointer for context sensitive tasks. If your streaming application requires a remote mouse pointer, set the environment variable `VDI_CAPTUREMOUSECURSOR` to true when you deploy your streaming application. For more information about setting this environment variable, see Using Environment Variables (p. 205).

This environment variable has the following effects on the client applications:

- Chrome App – the local mouse pointer, the remote mouse pointer, or both mouse pointers appear depending on the settings in **Mouse cursor** in **Settings**. For more information, see Chrome App (p. 54).
- Android and Fire OS – the remote mouse pointer appears.
- iOS – the remote mouse pointer appears.
- OS X – the local mouse pointer and the remote mouse pointer appear.
- Windows – the local mouse pointer and the remote mouse pointer appear.

The next section explains how to display the local mouse pointers in the client application.

# Local mouse pointer

The local mouse pointer is the mouse pointer in the client application. Which mouse pointer appears depends on the client application.

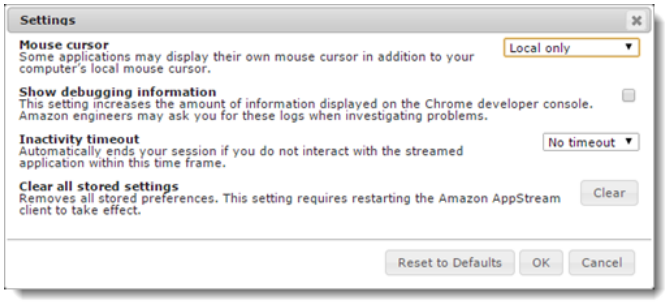
**Topics**

- Chrome App (p. 54)
- Android and Fire OS (p. 55)
- iOS (p. 55)
- OS X (p. 55)
- Windows (p. 56)

## Chrome App

The Chrome App can display the local mouse pointer, the remote mouse pointer, or both local and remote mouse pointers.

To set which the mouse pointer appears, choose **Settings** and then select an option in **Mouse Cursor**.




The following table summarizes what appears when you choose a setting. The **Default Deployment** column lists which mouse pointer appears when the streaming application is deployed using the default settings. The **`VDI_CAPTUREMOUSECURSOR` is true** column lists which mouse pointer appears after the environment variable `VDI_CAPTUREMOUSECURSOR` is set to true during the streaming application deployment.

### Chrome App Mouse Pointer Settings

| Dialog Box Setting | Default Deployment | `VDI_CAPTUREMOUSECURSOR` is true |
|---|---|---|
| **Local only** | Local mouse pointer | Local mouse pointer |
| **Remote only** | No mouse pointer | Remote mouse pointer |
| **Local and Remote** | Local mouse pointer | Local and remote mouse pointers (Two mouse pointers) |

For more information about setting the environment variable, see Using Environment Variables (p. 205).

## Android and Fire OS

No mouse pointer appears because mobile devices rely on touch for user input.

To display a mouse pointer, set the environment variable `VDI_CAPTUREMOUSECURSOR` to true during streaming application deployment.

For more information, see Remote mouse pointer (p. 54) and Using Environment Variables (p. 205).

## iOS

No mouse pointer appears because mobile devices rely on touch for user input.

To display a mouse pointer, set the environment variable `VDI_CAPTUREMOUSECURSOR` to true during streaming application deployment.

## OS X

The local mouse pointer appears because the operating system displays a mouse pointer in all applications.

Both the local and remote mouse pointers appear if you set the environment variable `VDI_CAPTUREMOUSECURSOR` to true during streaming application deployment.

## Windows

The local mouse pointer appears because the operating system displays a mouse pointer in all applications.

Both the local and remote mouse pointers appear if you set the environment variable `VDI_CAPTUREMOUSECURSOR` to true during streaming application deployment.

# Generating a New Test Connect Link to Stream Your Application

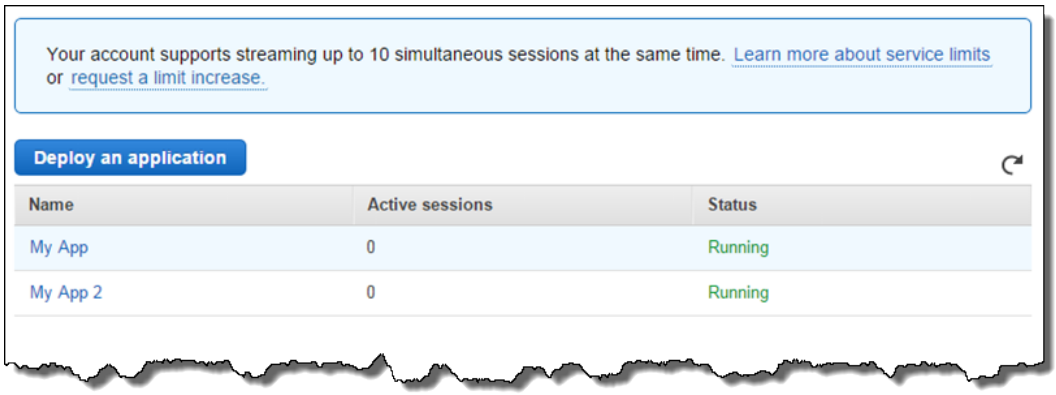
After you deploy a streaming application, you can stream by connecting a client application to an entitlement service or by streaming in standalone mode by using the test connect link. A test connect link is a unique URL with which you can stream your application without using an entitlement service. The URL is valid for 72 hours and can only be used once. You can share this URL within the 72 hour period, but only one user can use this URL.

If you want to stream your application after using the test connect link or you want another user to connect to your streaming application, you need to generate another test connect link.

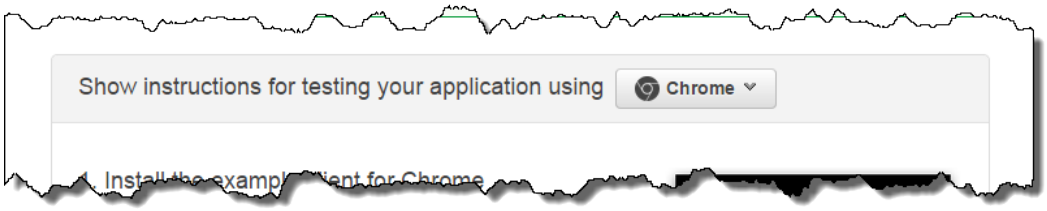
### To generate a new test connect link

1. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.


2. In the list of applications, click the name of the streaming application to view.




3. In **Show instructions for testing your application using**, select the operating system of your device or computer.




4. Do one of the following:

   - For Chrome, click **Generate a new quick link** in Step 2.
   - For other operating systems, click **Generate a new test URL** in Step 3.

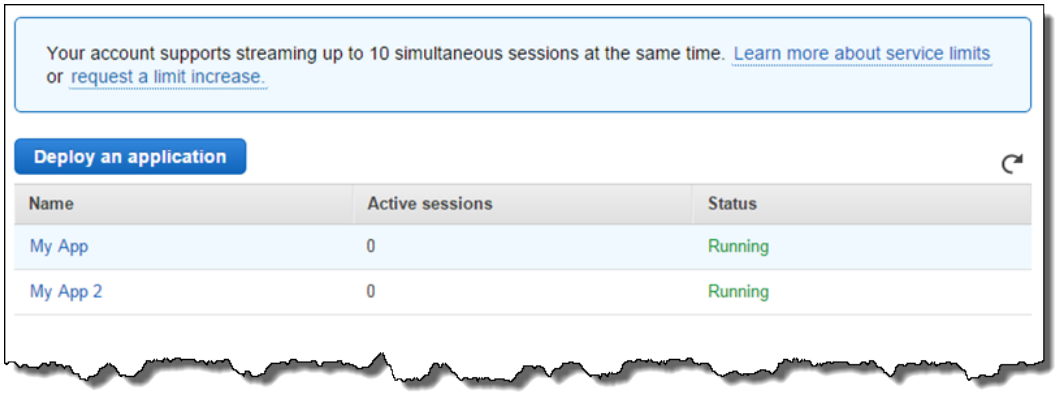
The new URL appears in the console. Use this URL to connect your client application to the streaming application in standalone mode. For more information, see Streaming Your Amazon AppStream Application (p. 16).

# Viewing All Applications

**To view a list of your applications in Amazon AppStream**

- In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.


Amazon AppStream displays a list of your applications.



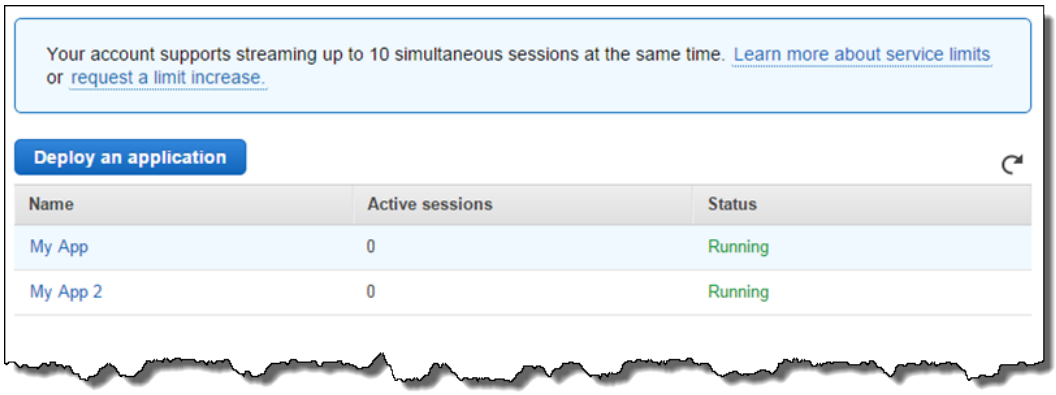

# Viewing Application Summary

The application summary page displays the settings associated with your application, including the application identifier.

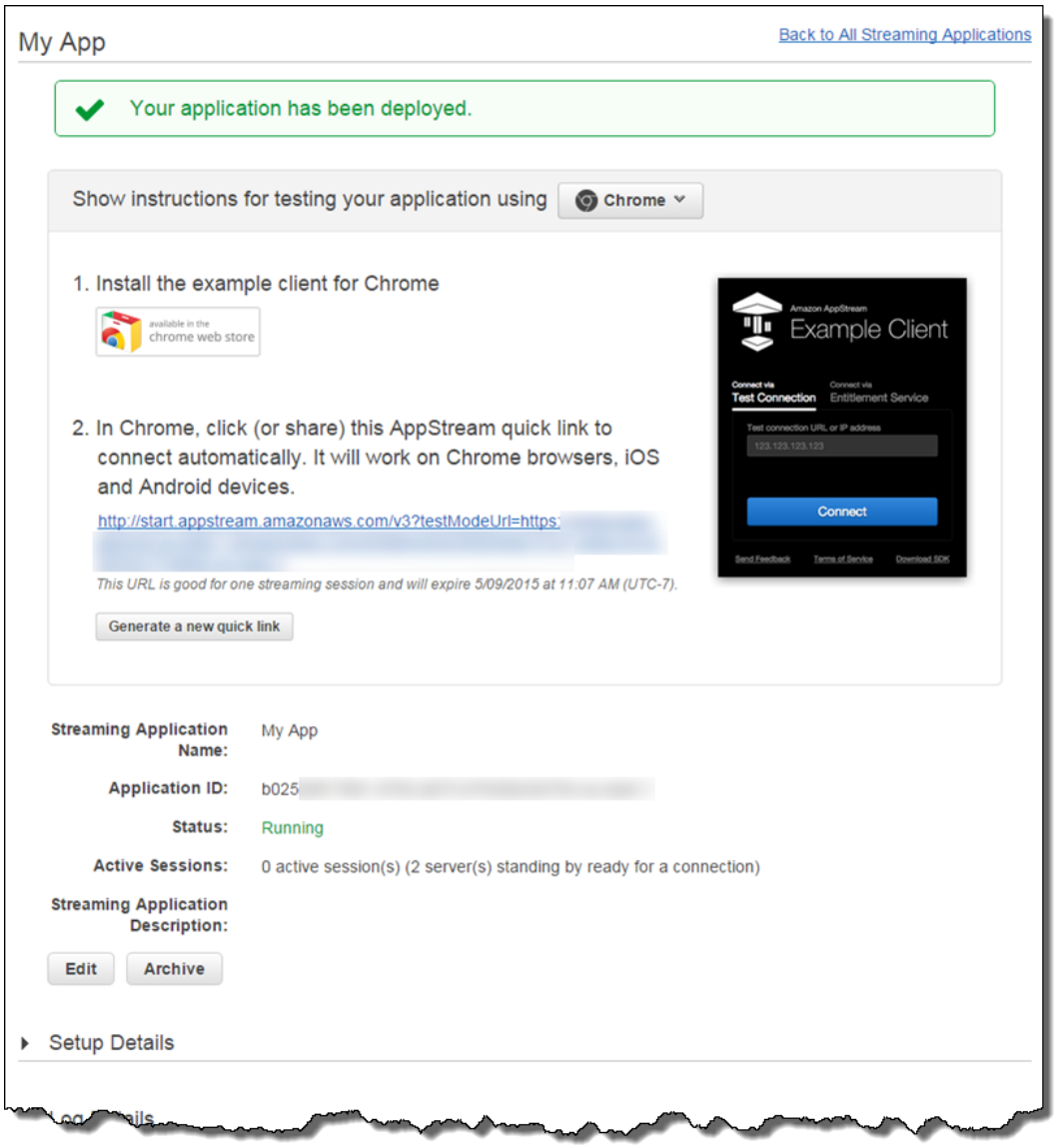
### To view detailed information about an application

1. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.


2. In the list of applications, click the name of the streaming application to view.




The **Streaming Application Summary** page displays information about the streaming application.

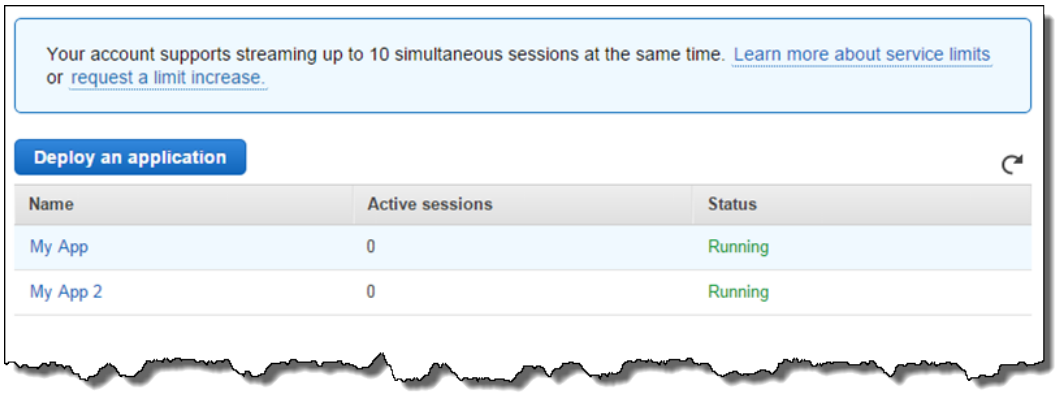



# Editing an Application

### To modify an application

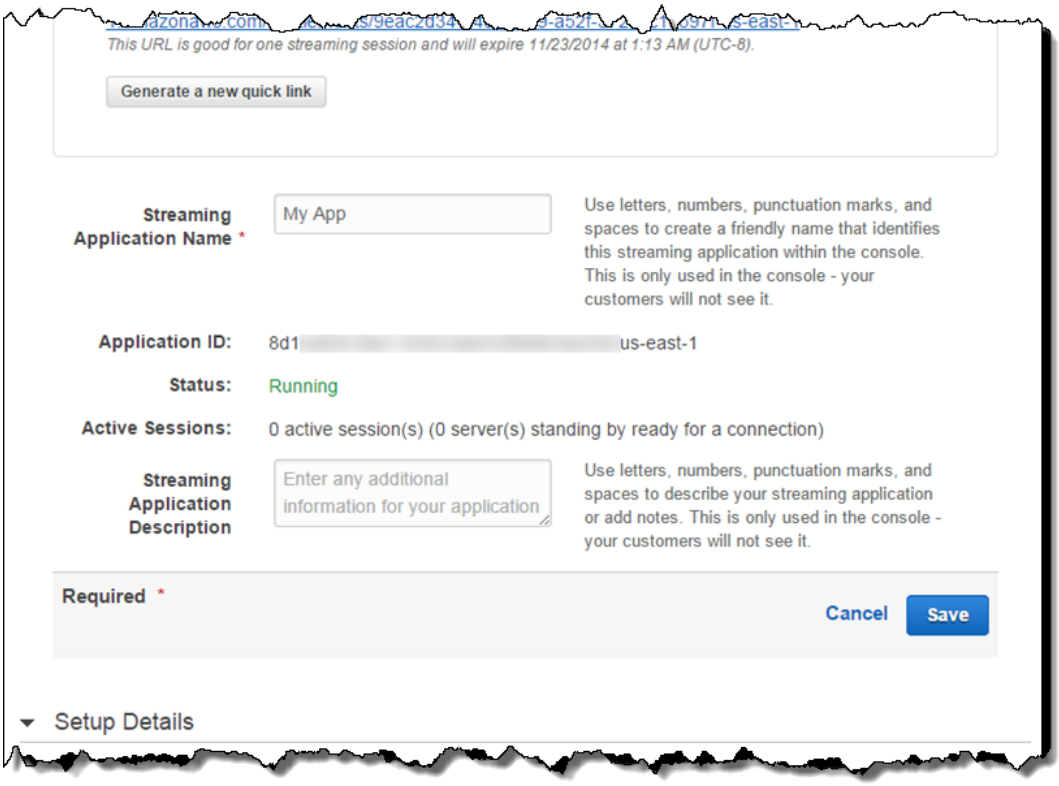
1. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.


2. In the list of applications, click the name of the streaming application to view.

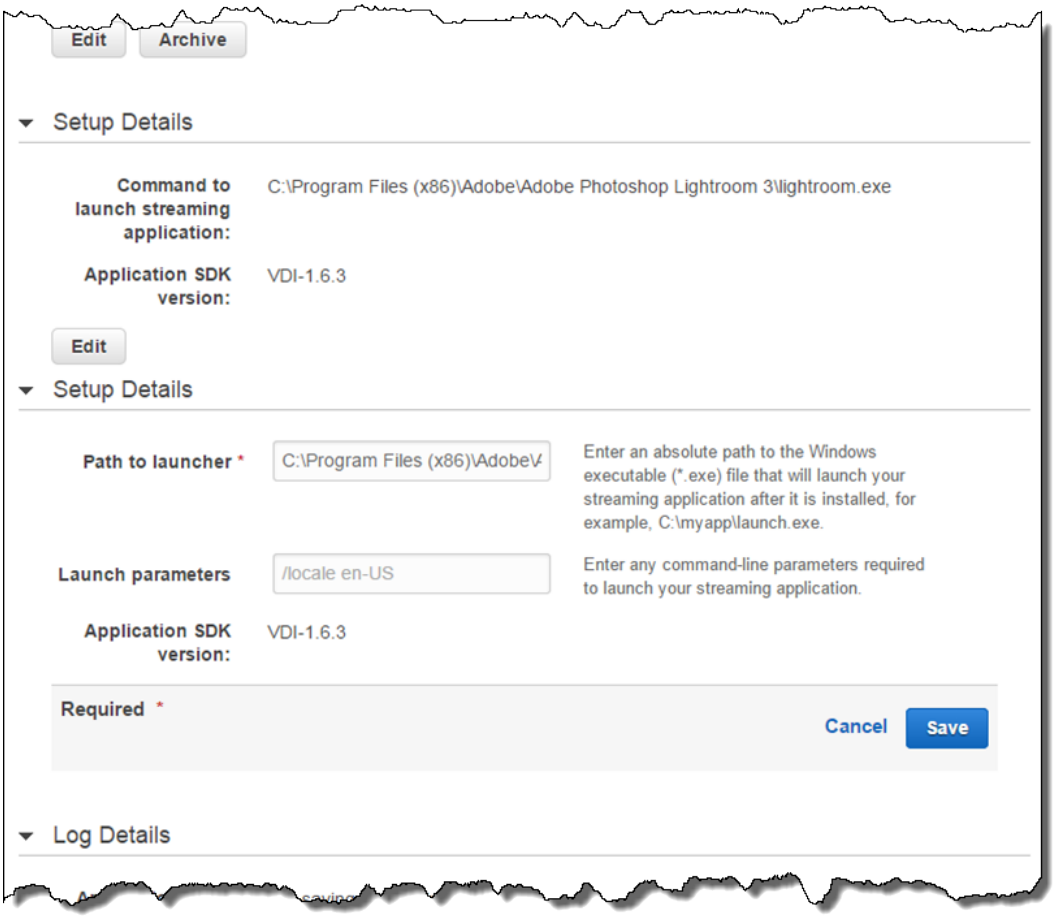



3. On the **Streaming Application Summary** page, do one of the following:

   - To change the streaming application name or description, click **Edit**. Change the name and description and then click **Save**.




- To change the launch path or launch parameters of the streaming application, click **Edit** under **Setup Details**. Change the launch path and launch parameters and then click **Save**.




# Archiving an Application

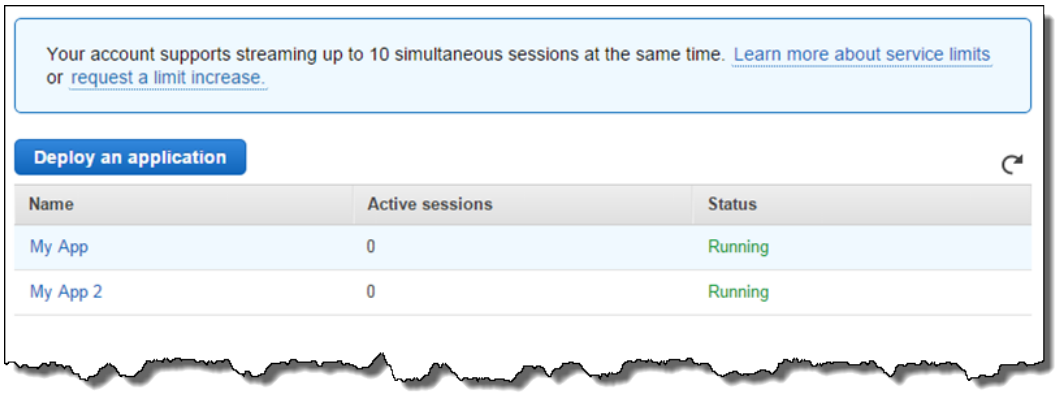
You can deprecate a streaming application and prevent it from accepting new client connections. When you do this, the streaming application is terminated and removed from the application list.

While the streaming application is being archived, the streaming application continues to stream content to current client sessions, but does not accept new client sessions. When all the existing client sessions have finished, the streaming application enters the archived state.

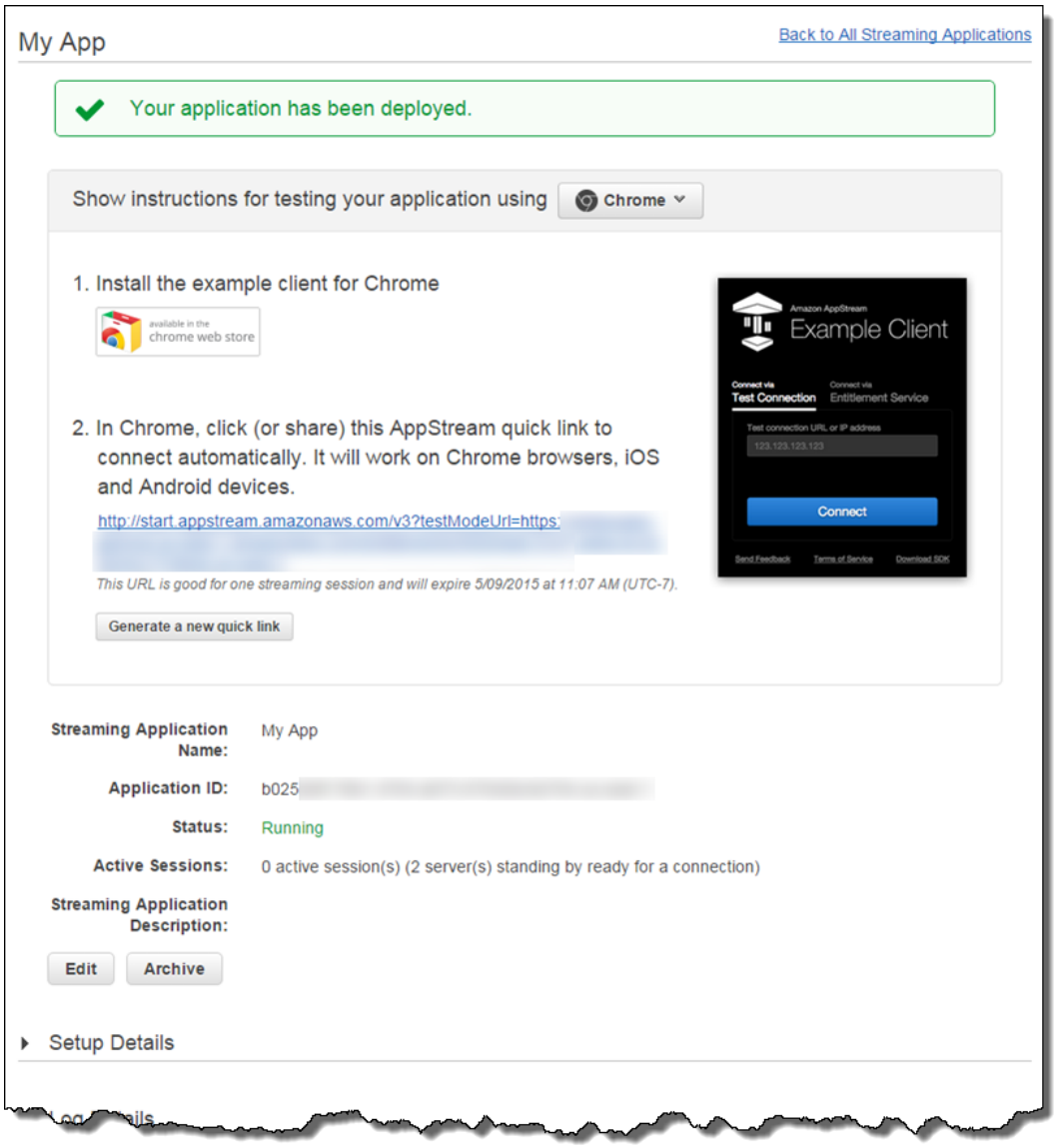
### To archive an application

1. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.

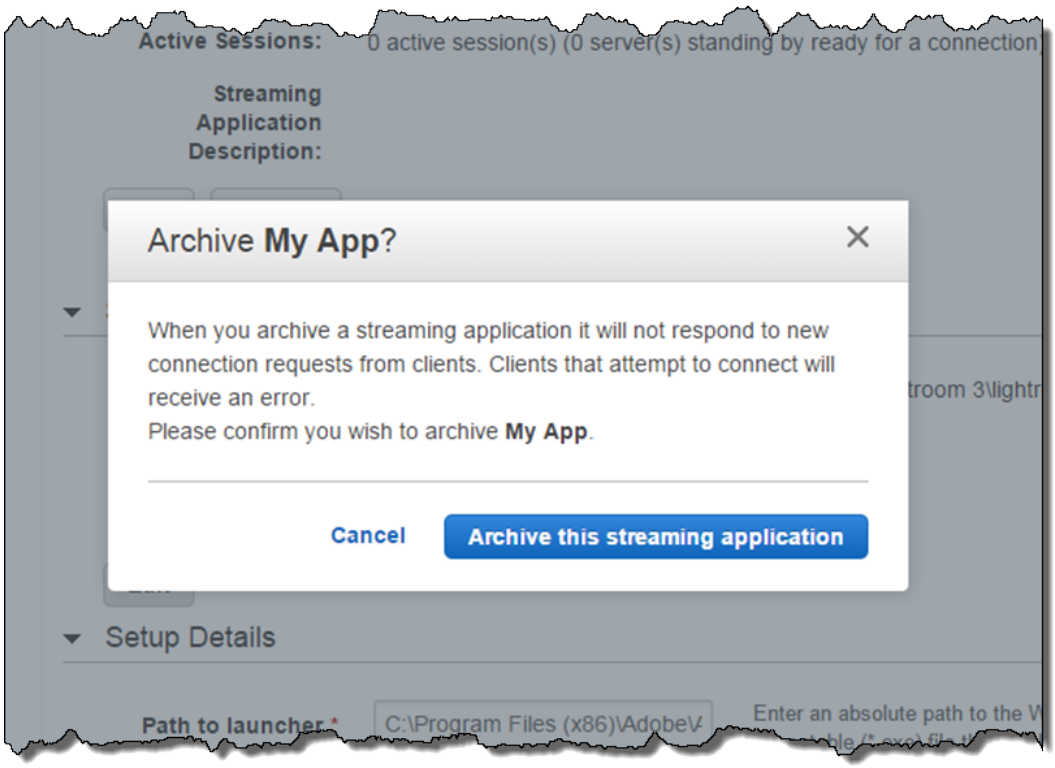

2. In the list of applications, click the name of the streaming application to view.




3. On the **Application Summary** page, click **Archive**.




4. In the **Archive** message box, click **Archive**.




Amazon AppStream moves the application into the archived state. This may take a few minutes.

When the application is archived, it no longer appears in the list of applications.

# Enabling Logging on a Streaming Application

When your application exhibits unusual behaviors, such as unexpected shutdowns, intermittent streaming, or poor performance, or you want to know what your streaming application is doing, you can enable logging to see what is happening.

When logging is enabled, Amazon AppStream creates a `.zip` file of all the logs that you specify and uploads that `.zip` file is then uploaded to a specified Amazon S3 bucket in your account. Once the logs are stored in Amazon S3, you can download them and see what was happening within your streaming application during the session.

Although Amazon AppStream only upload logs after a session terminates, log creation during a session can adversely affect the streaming experience. Balance the need for logging events with the effect on the streaming experience by enabling logging.

**Note**
Amazon AppStream does not encrypt the log files by default. If you want encrypted logs, encrypt them when they are generated (client side encryption) or on the Amazon S3 bucket (server side encryption).

## AppStream Log Names

Amazon AppStream uses this pattern to name the `.zip` file:

*<bucketName>*/AppStream/*<region>*/*<Application ID>*/*yyyy-mm-dd*/*<log file name>*.zip

# Default Amazon AppStream Logs

By default, Amazon AppStream saves the following logs:

- A file called `launch.txt.failed` with the output if Amazon AppStream fails to start the streaming application.
- Minidump files (`.dmp`)
- Standard error stream (`launch.txt.stderr`)
- Standard output stream (`launch.txt.stdout`)
- Utilization metrics (`utilization.csv`) which is a file that provides operational metrics. The first row contains column headings and the following rows are comma-separated values:
  - **UTC Time**—Time in Universal Coordinated Time (UTC) when the row was created.
  - **Processor (*n*) %**—The percentage of use for each CPU core starting with Core 0.
  - **Memory in Use (Bytes)**—Memory that was used in bytes.
  - **Memory Available (Bytes)**—Memory that was available in bytes.
  - **Disk Read (Bytes)**—Disk read speed in bytes per second.
  - **Disk Write (Bytes)**—Disk write speed in bytes per second.
  - **GPU Utilization %—**The percentage of use for the GPU.
  - **GPU Memory Utilization %—**The percentage of GPU memory that was used.
  - **GPU Memory Total (MB)—**The total GPU memory available in the megabytes (MB).
  - **GPU Memory Used (MB)—**The amount of GPU memory used in MB.
  - **GPU Memory Free (MB)—**The amount of GPU memory available in MB.
  - **GPU Temperature (C)—**The temperature of the GPU in degrees centigrade.
  - **GPU Power Draw (W),—**The number of watts the GPU is using.
  - **GPU Power Limit (W)—**The maximum number of watts required by the GPU.

  The following is an example of the metrics in this utilization log.

```
UTC Time,Processor(0) %,Processor(1) %,Processor(2) %,Processor(3) %,Memory
In Use (Bytes),Memory Available (Bytes),Disk Write (Bytes/sec),Disk Read
(Bytes/sec),GPU Utilization %,GPU Memory Utilization %,GPU Memory Total
(MB),GPU Memory Used (MB),GPU Memory Free (MB),GPU Temperature (C),GPU Power
 Draw (W),GPU Power Limit (W)
2014/05/15
11:17:39,0.92,3.14,6.58,1.88,2724400196.27,14014595549.87,614818.20,103422.13,0,
 1, 4095, 4063, 32, 33, 18.44, 125.00
2014/05/15
11:18:40,0.95,1.15,0.32,1.24,2745420390.40,14010372983.47,8689.31,99415.41,0,
 1, 4095, 4063, 32, 33, 18.26, 125.00
2014/05/15
11:19:41,0.36,0.67,0.20,1.30,2734278997.33,14029225574.40,6994.11,18322.61,0,
 1, 4095, 4063, 32, 33, 18.44, 125.00
2014/05/15
11:20:42,0.08,0.69,0.10,0.97,2724477064.53,14033759027.20,27142.17,14072.01,0,
 1, 4095, 4063, 32, 33, 18.54, 125.00
2014/05/15
11:21:43,0.18,0.27,0.10,1.42,2719389627.73,14038990438.40,40446.51,203.51,0,
 1, 4095, 4063, 32, 33, 18.53, 125.00
2014/05/15
```

```
11:22:44,0.13,0.44,0.12,0.87,2720666555.73,14039756800.00,43574.68,1696.60,0,
 1, 4095, 4063, 32, 32, 18.67, 125.00
2014/05/15
11:23:45,0.18,0.44,0.24,0.92,2718720887.47,14040533674.67,44824.72,271.31,0,
 1, 4095, 4063, 32, 32, 18.43, 125.00
2014/05/15
11:24:46,0.30,0.50,0.14,1.73,2707566796.80,14053706888.53,72334.31,209776.90,0,
 1, 4095, 4063, 32, 32, 18.21, 125.00
2014/05/15
11:25:46,0.34,0.44,0.14,1.52,2654716040.53,14106968064.00,43219.86,1479.45,0,
 1, 4095, 4063, 32, 33, 18.41, 125.00
2014/05/15
11:26:47,0.05,0.88,0.16,1.08,2655204010.67,14107344554.67,43875.24,0.00,0, 1,
 4095, 4063, 32, 32, 18.31, 125.00
2014/05/15
11:27:48,0.38,0.79,0.16,1.25,2653159219.20,14107987626.67,40567.09,0.00,0, 1,
 4095, 4063, 32, 33, 18.23, 125.00
2014/05/15
11:28:49,0.17,0.59,0.09,1.32,2650824772.27,14108805666.13,43111.49,0.00,0, 1,
 4095, 4063, 32, 33, 18.20, 125.00
2014/05/15
11:29:50,0.16,0.45,0.07,0.76,2651427157.33,14109135803.73,43341.33,0.00,0, 1,
 4095, 4063, 32, 33, 18.33, 125.00
2014/05/15
11:30:51,0.05,0.83,0.11,0.92,2650787703.47,14109916501.33,44690.80,0.00,0, 1,
 4095, 4063, 32, 33, 18.30, 125.00
2014/05/15
11:31:52,0.13,0.64,0.29,0.79,2651821602.13,14110050781.87,44109.33,0.00,0, 0,
 4095, 4063, 32, 33, 18.18, 125.00
2014/05/15
11:32:52,0.14,0.99,0.44,1.34,2655279445.33,14108282675.20,56575.66,2570.57,0,
 0, 4095, 4063, 32, 32, 17.78, 125.00
```

# Custom Amazon AppStream Logs

In addition to the above files, Amazon AppStream also collects logs located in directories that you specify. To specify custom files or directories simply include that information during the create application process or by updating the metadata of your application.

If you want Amazon AppStream to create the bucket, the new bucket name will use the following naming convention:

`AppStream-`*`<region>`*`-`*`<Account ID>`*

If you want to use your own bucket, you must allow Amazon AppStream to use the `PutObject` method. To allow this method, use a bucket policy similar to the following:

```
{  "Version": "2008-10-17",
   "Id": "Policy1396472828471",
   "Statement": [
      {
         "Sid": "Stmt1396472820469",
         "Effect": "Allow",
         "Principal": {"AWS":"arn:aws:iam::990116983315:root"},
```

```
        "Action": ["s3:PutObject"],
        "Resource": "arn:aws:s3:::your bucket name/*"
      }
    ]
}
```

For more information about modifying the Amazon S3 bucket policy, see Bucket Policy Examples in the Amazon S3 Developer Guide.

You can also specify log files from other directories by filling the text box with the paths to directories that contain the log files or the path and filename of the log. Separate each path with a semi-colon (`;`). The paths and files must meet the following requirements:

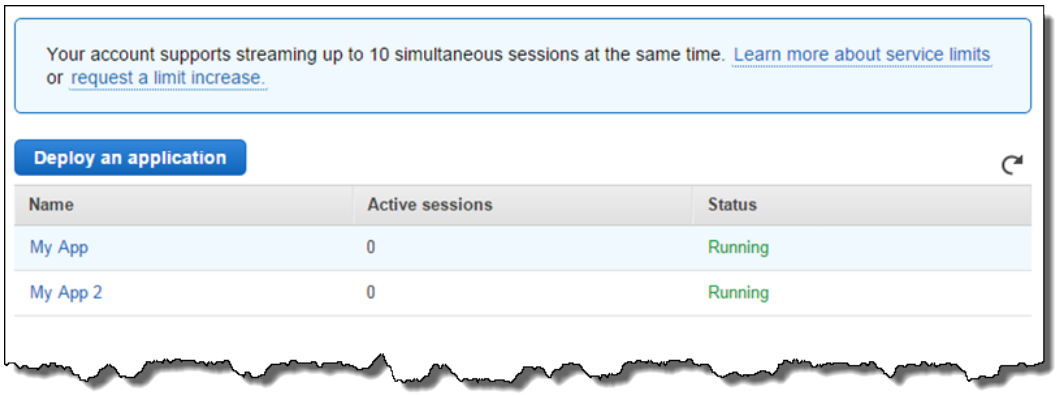
- Log files must be located in the `C:\` directory.
- Log paths must be fully qualified paths. Do not use relative paths. You can use wildcards.
- Log paths must be less than the path length limits. See Naming Files, Paths, and Namespaces.
- Log file names cannot contain Windows reserved characters. See Naming Conventions.
- Your application must have read access to the paths.

**Note**
The total size of logs in the `.zip` file is subject to the Amazon S3 key size limits.

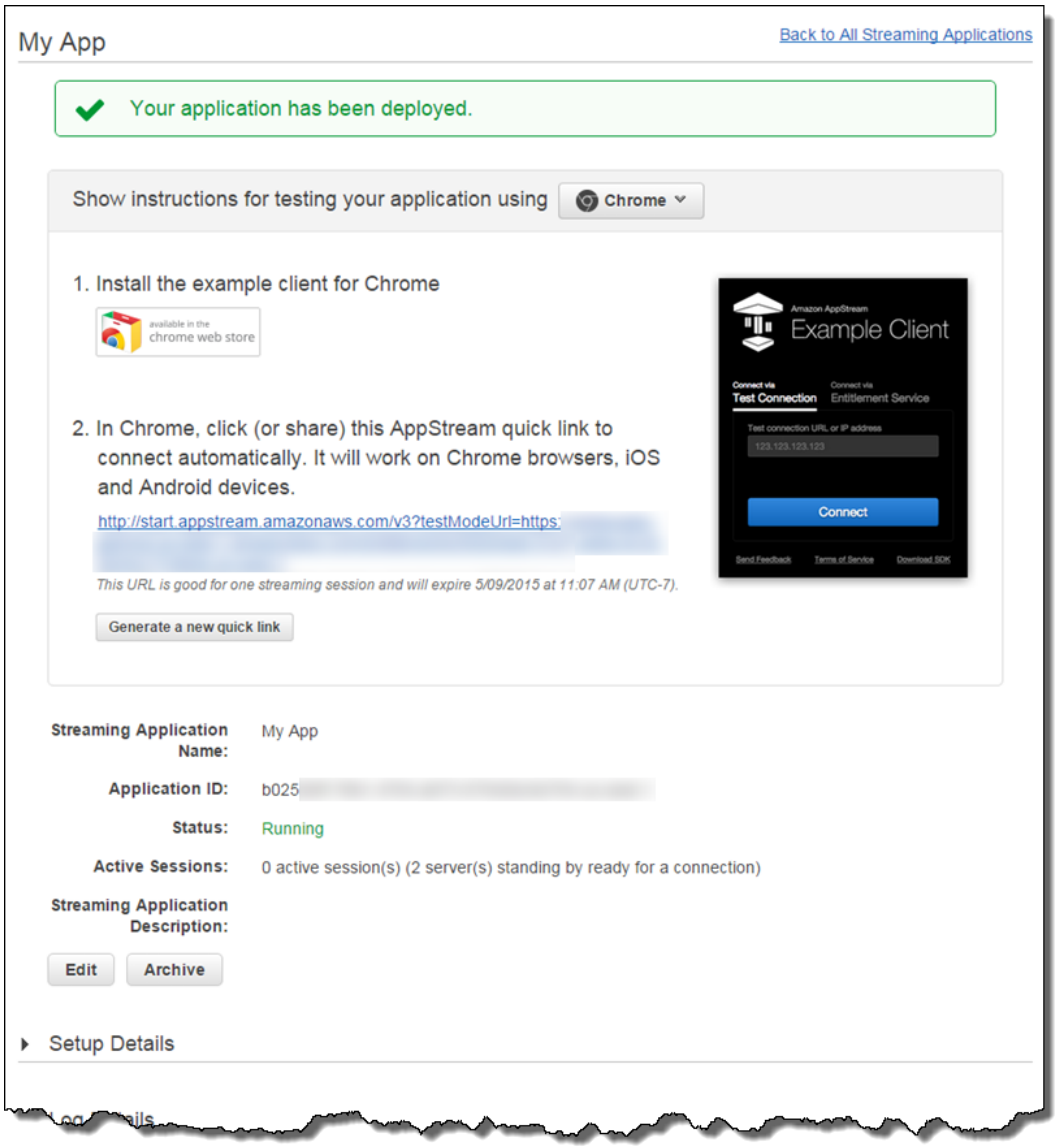
### Enabling Amazon AppStream Logging through The Console

1. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. In the Amazon AppStream console, click **View your streaming applications**. This button appears only if you have added applications to Amazon AppStream.


3. In the list of applications, click the name of the streaming application to view.




The **Streaming Application Summary** page displays information about the streaming application.



4. On the **Streaming Application Summary** page, open **Log Details**.
5. Under **Log Details**, click **Edit**.
6. Select **Yes, save application logs**.
7. (Optional) Use the text box that appears to specify other paths or log files to include in the `.zip` file. Each time you fill in a text box, another text box appears so that you can specify other paths or logs.
8. To have Amazon AppStream send your logs to the default Amazon S3 bucket, click **Create a default bucket**. To use one of your existing Amazon S3 buckets, click the down arrow next to **Create a default bucket** and select one of your buckets.
9. Click **Save**.

## Enabling Amazon AppStream Logging Programatically

You can enable logging programmatically when you add your streaming application to Amazon AppStream or update the metadata for an existing streaming application. In either case, use one of the methods to provide the relevant data in the `applicationManifest` input field.

**To enable logging when creating a new streaming application:**

1. Access the root by calling `/`.
2. Follow the Applications (p. 226) link to get a list of applications.
3. Follow the application:create (p. 234) link to create the application.
4. Fill in the necessary information in the `applicationManifest` input field.

**To enable logging on an existing streaming application:**

1. Access the root by calling `/`.
2. Follow the Applications (p. 226) link to get a list of applications.
3. Follow the application:by-id (p. 234) link to get the proper application representation.
4. Follow the application:update (p. 236) link to update the streaming application.
5. Fill in the necessary information in the `applicationManifest` input field.

# Increasing Your Amazon AppStream Service Limits

Your streaming application has a service limit of up to five concurrent streaming sessions:

- Up to two concurrent streaming application deployments that use the interactive wizard.
- Up to three streaming applications in the **Building**, **Active**, and **Error** states. For more information about these states, seeAmazon AppStream Application Lifecycle (p. 7).

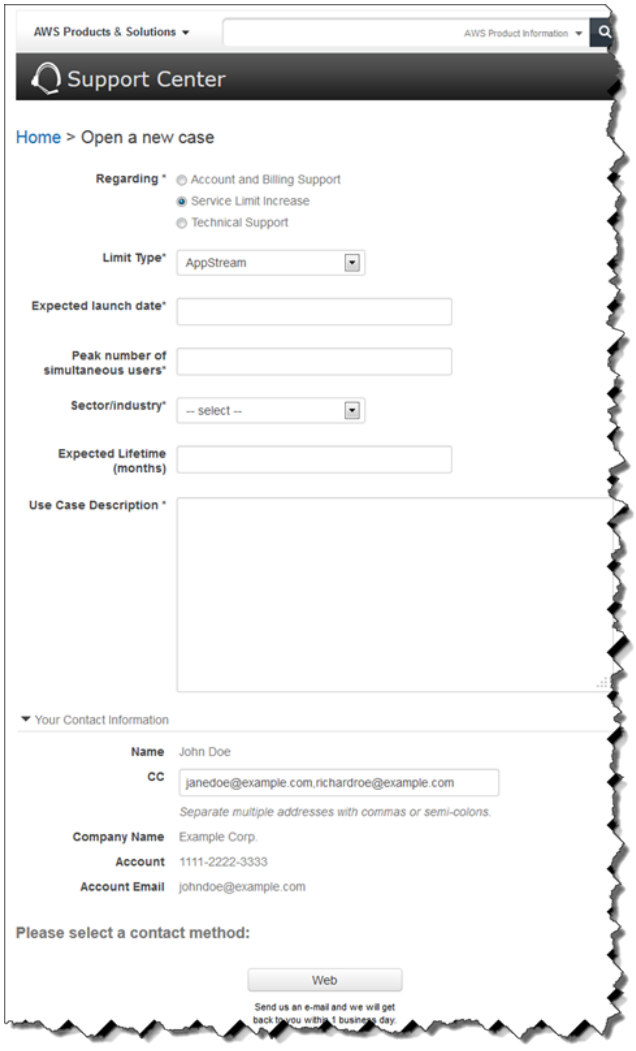
This means that your streaming application can stream to a maximum of five users at any given time. New AWS accounts that have not previously used the Amazon AppStream service may have a lower limit for the number of simultaneous streaming sessions. If you need more simultaneous connections, you can request a service limit increase by starting a new case at the Support Center .

Before contacting the Support Center, you should gather the following information:

- The number of simultaneous sessions you need
- Expected date on which you need your new limit.
- How many months the limit needs to be in effect.
- A description of your use case.

### To request a service limit increase

1. Go to the Support Center and click **Open a new case**.
2. Select **Service Limit increase**. The form changes after you make this selection.




3. For **Regarding**, click **Service Limit Increase**.
4. For **Limit Type**, click **AppStream**.
5. Fill in all of the required options in the form and then click **Web**.

# Managing Your Users

To control who can connect to your streaming application, you use an *entitlement service*. An entitlement service is a web service that authenticates and authorizes users to connect to your streaming application.

Amazon AppStream has an example entitlement service that you can deploy to use for testing. For more information on deploying the example entitlement service, see Amazon AppStream Sample Entitlement Service (p. 200). You can download the source code from Getting Started with Amazon AppStream (p. 13).

**Topics**

# Building a Client Application

The client renders the video and audio stream from the application on a device. The client also sends the input on the device to the application. This section provides guidelines on building clients for the different platforms. The Amazon AppStream SDK includes both pre-compiled clients and sample code that you can compile into a client. This section has excerpts from those samples.

**Topics**


- Design Considerations for Your Clients (p. 74)
- Building a Client for Android (p. 74)
- Building a Chrome App (p. 82)
- Building a Client for iOS (p. 83)
- Building a Client for OS X (p. 96)
- Building a Client for Windows (p. 107)
- Codec and Open Source Licensing (p. 116)
- Amazon AppStream Client Application License and Notices (p. 117)


## Design Considerations for Your Clients

When creating your client, take into account the following:

- **Resolution differences between the application and the device**. Your application may be designed for an resolution that is different from the resolution of your client. Consider how you will deal with that discrepancy. the resolution difference is especially important if the client will send inputs at a high precision to the application. An example of high precision input is if you are editing bitmap graphics at the pixel level.
- **Applications can have more mouse events that the client has input events**. Amazon AppStream interprets events from the client as key or mouse events. However, the number of device input events are less than available mouse events. If your application handles mouse events that do not have corresponding device events, you will need to design for those discrepancies.

## Building a Client for Android

The Amazon AppStream SDK provides libraries that you build into your client to receive and decode video and audio from the application and send device inputs and raw inputs to the application.

The following object classes are provided by the client libraries in the Amazon AppStream SDK.

| Class | Description |
|---|---|
| AppStreamInterface | This class manages client sessions between clients and applications. |
| ConnectDialogFragment | This class creates sessions between clients and applications. |
| DesQuery | This class is for use with the entitlement service. |
| ErrorDialogFragment | This class is to display messages. |
| GL2JNIView | This class is for the video in the client. |
| SampleClientActivity | This class is for the client interface. |

Your client implements and populate the following structures in order to receive session and application events.

| Interface | Description |
|---|---|
| ConnectDialogListener | Receives callbacks from a client. |
| DesQueryListener | Receives callbacks from the entitlement service |

# Lifecycle of a Client for Android

Your client communicates with the application on Amazon AppStream through libraries of the Amazon AppStream SDK.

The lifecycle of a client for Android is as follows:

1. **Request authorization from the entitlement service to connect to the application.** The entitlement service will return the URL to connect to your application.
2. **Connect to the application.** The object manages the session between the client and the application.
3. **Send client inputs to the application.**
4. **Close the client.**

# Building the Example Client for Android

The Amazon AppStream SDK contains source code that you can build into an example client file. To learn how to build the source code, see the `\doc` directory.

# Choosing a Color Subsampling Rate

Amazon AppStream streams the video at the YUV420 color subsampling rate to client applications. If your streaming application, client application and device support YUV444, you can set your client application to display the video at YUV444. The example client application in Amazon AppStream SDK is configured to accept and render a stream in the YUV444 color subsampling rate.

When the client application connects to the streaming application, Amazon AppStream compares the color subsampling options available on the client application with the options advertised on the streaming application. Amazon AppStream then selects the highest color resolution supported by both the client

application and streaming application. Amazon AppStream then calls the `XStxIClientListener2FcnSetConfiguration` callback function that the client application supplied and passes the structure with the *`XStxChromaSampling`* setting.

The following excerpt from `AndroidVideoDecoder.cpp` illustrates this step. This file is in the *<SDK_dir>*`\example_src\client\src\android\jni` directory.

```
/**
 * Provide Chroma sampling capability.
 */
bool AndroidVideoDecoder::isChromaSamplingSupported(XStxChromaSampling chro
maSampling)
{
    // If we have hardware, then ask the hardware whether we have
XSTX_H264_PROFILE_HIGH444.
    // If not, then we fall back to 420.
    if (androidHWDecodeAvailable() && (androidGetProfile()!=XSTX_H264_PRO
FILE_HIGH444))
    {
        return chromaSampling == XSTX_CHROMA_SAMPLING_YUV420;
    }
    else
    {
        return chromaSampling == XSTX_CHROMA_SAMPLING_YUV420
            || chromaSampling == XSTX_CHROMA_SAMPLING_YUV444;
    }
}
```

The following excerpt from `VideoPipeline.cpp` illustrates this step. This file is in the *<SDK_dir>*`\example_src\client\src\android\jni` directory.

```
decoderCapabilities->mProfile = XSTX_H264_PROFILE_HIGH444;
```

# Getting Authorization to Connect to Your Application

The first activity in your client is to get the Entitlement URL from the endpoint of the entitlement service. The entitlement service needs information in order to provide the Entitlement URL.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `makeQuery` method of `DesQuery.java`.

```
public void makeQuery(String address, final String appid, final String userid)
 {

    if (!address.startsWith("http"))
    {
        address = "http://" + address;
    }

    if (!address.endsWith("/"))
    {
```

```
        address += "/";
    }

    // if there's no path, add api/entitlements/
    if (!address.matches("http[s]?://.+/.+"))
    {
        address += "api/entitlements/";
    }

    final String cleanAddress = address;

    Thread httpThread = new Thread() {

@Override
public void run() {
    HttpParams httpParameters = new BasicHttpParams();

    // timeout for a connection in ms
    HttpConnectionParams.setConnectionTimeout(httpParameters, 1200);

    // timeout when waiting for data
    HttpConnectionParams.setSoTimeout(httpParameters, 1500);

    HttpClient client = new DefaultHttpClient();
    HttpPost request = new HttpPost();
    String queryURL = cleanAddress+appid;
    try {
            request.setURI(new URI(queryURL));
        }

    catch (URISyntaxException e) {

            sendError("Either address or appid isn't in the correct format.
Generated URL:"+queryURL+" Error:"+e.getLocalizedMessage());
            return;
   }

 request.setHeader("Authorization", "Username"+userid );
    request.setHeader("Content-Type", "application/x-www-form-urlencoded" );

    List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(1);
       nameValuePairs.add(new BasicNameValuePair("terminatePrevious", "true"));


    String entitlementResult = null;
    try {
            request.setEntity(new UrlEncodedFormEntity(nameValuePairs));
            HttpResponse response = client.execute(request);
            if (response.getStatusLine().getStatusCode()<200 || response.get
StatusLine().getStatusCode()>201)
        {
            if (response.getStatusLine().getStatusCode()==503) {
                sendError("All of our streaming servers are currently in use.
Please try again in a few minutes. [503]");
        } else if (response.getStatusLine().getStatusCode()==404) {
            sendError("The link to the streaming server has expired. [404]");
        } else {
        sendError("Query failed with response: "+response.getStatusLine());
```

```
        }
        return;
    }

    BufferedReader result = new BufferedReader( new InputStreamReader( re
sponse.getEntity().getContent()
          )
        );

    final String url = result.readLine().trim();
    Log.v(TAG,"Resulting URL: "+url);
    result.close();

    HttpGet entitlementRequest = new HttpGet(url);
    response = client.execute(entitlementRequest);

   result = new BufferedReader(new InputStreamReader( response.getEntity().get
Content() ) );

    StringBuilder buffer = new StringBuilder(1024);

    char[] charBuf = new char[400];
    while (result.read(charBuf,0,400)>0)
        {
            buffer.append(charBuf);
        }

 entitlementResult = buffer.toString();

    JSONObject object = (JSONObject) new JSONTokener(entitlementResult).next
Value();
    final String sessionID = object.optString("sessionID");
    final String ec2Host = object.optString("ec2Host");

    if (sessionID==null || ec2Host==null)
        {
            final String message = object.optString("message");
            if (message!=null)
                {
                    sendError("Entitlement query failed with response: "+mes
sage+"["+response.getStatusLine()+"]");
                    return;
                }
                sendError("Error parsing entitlement result: "+entitlementRes
ult);
    return;
        }

    mActivity.runOnUiThread(new Runnable(){

 @Override
    public void run() {
        String url = String.format(Locale.US,"%s:80?sessionId=%s",ec2Host,ses
sionID);

        Log.v(TAG,"Sending host url "+url);
        mListener.onDesQuerySuccess(url);
 }});
```

```
 } catch (ClientProtocolException e) {
     sendError("Protocol Exception: "+e.getLocalizedMessage());

    } catch (IOException e) {
        sendError("Problem With Connection: "+e.getLocalizedMessage());

    } catch (JSONException e) {
       sendError("Error reading entitlement result: "+e.getLocalizedMessage()+"
 Result: "+entitlementResult);
           }
        }
    };

    httpThread.start();
}
```

The function is successful when the entitlement service calls a callback function with the Entitlement URL to start the session with the application.

The following excerpt from the example client in the Amazon AppStream SDK illustrates the callback function is called to connect the client to the application. The excerpt is from the `onDesQuerySuccess` method of `SampleClietActivity.java`.

```
public void onDesQuerySuccess(String address) {
    AppStreamInterface.connect(address);
    AppStreamInterface.newFrame();
}
```

# Sending Your Client Inputs to Your Application

The client sends inputs to the application through several methods, depending on the input type.

The following excerpt from the example client in the Amazon AppStream SDK illustrates a method that sends motion inputs to the application. The excerpt is from the `dispatchGenericMotionEvent` method in `SampleClientActivity.java`.

```
public boolean dispatchGenericMotionEvent (MotionEvent event) {
    if (event.getSource()==InputDevice.SOURCE_MOUSE)
    {
  event.getPointerCoords(0, mCoordHolder);
     switch (event.getAction())
     {
     case MotionEvent.ACTION_HOVER_MOVE :
      AppStreamInterface.mouseEvent((int)mCoordHolder.x,(int)mCoordHolder.y,
0);   break;

     default:
      return super.dispatchGenericMotionEvent(event);
     }
     return true;
    }
    else if (event.getSource()==InputDevice.SOURCE_JOYSTICK
```

```
            && event.getAction() == MotionEvent.ACTION_MOVE) {
        Log.v(TAG,"Joystick event:"+event.toString());
        /// @todo: Handle motion event
        return true;
    }

 return super.dispatchGenericMotionEvent(event);
}
```

The following excerpt from the example client in the Amazon AppStream SDK illustrates a method that sends touch inputs to the application. The excerpt is from the `dispatchTouchEvent` method in `SampleClientActivity.java`.

```
public boolean dispatchTouchEvent (MotionEvent event) {

 if (mScaleGestureDetector.onTouchEvent(event))
 {
  if (mScaleGestureDetector.isInProgress())
   return true;
 }

    int flags = 0;
    if (event.getSource()==InputDevice.SOURCE_TOUCHSCREEN)
    {
        flags = AppStreamInterface.CET_TOUCH_FLAG ;
    }

    event.getPointerCoords(0, mCoordHolder);
    switch (event.getAction())
    {
        case MotionEvent.ACTION_MOVE :
            AppStreamInterface.mouseEvent((int)mCoordHolder.x,(int)mCoordHold
er.y, flags); break;
        case MotionEvent.ACTION_DOWN :
            AppStreamInterface.mouseEvent((int)mCoordHolder.x,(int)mCoordHold
er.y, AppStreamInterface.CET_MOUSE_1_DOWN|flags); break;
        case MotionEvent.ACTION_UP :
            AppStreamInterface.mouseEvent((int)mCoordHolder.x,(int)mCoordHold
er.y, AppStreamInterface.CET_MOUSE_1_UP|flags);
           AppStreamInterface.mouseEvent((int)-1000,(int)-1000, flags); break;


        default:
            return super.dispatchGenericMotionEvent(event);
    }
    return super.dispatchTouchEvent(event);
}
```

The following excerpt from the example client in the Amazon AppStream SDK illustrates methods that sends key inputs to the application. The excerpt is from `SampleClientActivity.java`.

```
private void onKey(KeyEvent msg, boolean down)
{
    int keyCode =msg.getKeyCode();
```

```
switch (keyCode)
{
case KeyEvent.KEYCODE_DPAD_RIGHT :
    AppStreamInterface.keyPress(0x27,down); break;
case KeyEvent.KEYCODE_DPAD_LEFT :
    AppStreamInterface.keyPress(0x25,down); break;
case KeyEvent.KEYCODE_DPAD_UP :
    AppStreamInterface.keyPress(0x26,down); break;
case KeyEvent.KEYCODE_DPAD_DOWN :
    AppStreamInterface.keyPress(0x28,down); break;
case KeyEvent.KEYCODE_FORWARD_DEL:
    AppStreamInterface.keyPress(0x2E,down); break;
case KeyEvent.KEYCODE_DEL:
    AppStreamInterface.keyPress(0x08,down); break;
case KeyEvent.KEYCODE_ALT_LEFT:
    AppStreamInterface.keyPress(0x12,down);
    AppStreamInterface.keyPress(0xA4,down); break;
case KeyEvent.KEYCODE_ALT_RIGHT:
    AppStreamInterface.keyPress(0x12,down);
    AppStreamInterface.keyPress(0xA5,down); break;
case KeyEvent.KEYCODE_CTRL_LEFT:
    AppStreamInterface.keyPress(0x11,down);
    AppStreamInterface.keyPress(0xA2,down); break;
case KeyEvent.KEYCODE_CTRL_RIGHT:
    AppStreamInterface.keyPress(0x11,down);
    AppStreamInterface.keyPress(0xA3,down); break;
case KeyEvent.KEYCODE_SHIFT_LEFT:
    AppStreamInterface.keyPress(0x10,down);
    AppStreamInterface.keyPress(0xA0,down); break;
case KeyEvent.KEYCODE_SHIFT_RIGHT:
    AppStreamInterface.keyPress(0x10,down);
    AppStreamInterface.keyPress(0xA1,down); break;
case KeyEvent.KEYCODE_APOSTROPHE:
    AppStreamInterface.keyPress('\'',down); break;
case KeyEvent.KEYCODE_AT:
    AppStreamInterface.keyPress('@',down); break;
case KeyEvent.KEYCODE_ENTER:
    AppStreamInterface.keyPress(0x0D,down); break;
case KeyEvent.KEYCODE_BREAK:
    AppStreamInterface.keyPress(0x03,down); break;
case KeyEvent.KEYCODE_VOLUME_UP:
    AppStreamInterface.keyPress(0xAF,down); break;
case KeyEvent.KEYCODE_VOLUME_DOWN:
    AppStreamInterface.keyPress(0xAE,down); break;
case KeyEvent.KEYCODE_MEDIA_NEXT:
    AppStreamInterface.keyPress(0xB0,down); break;
case KeyEvent.KEYCODE_MEDIA_PREVIOUS:
    AppStreamInterface.keyPress(0xB1,down); break;
case KeyEvent.KEYCODE_MEDIA_STOP:
    AppStreamInterface.keyPress(0xB2,down); break;
case KeyEvent.KEYCODE_MEDIA_PLAY:
    AppStreamInterface.keyPress(0xFA,down); break;
case KeyEvent.KEYCODE_MEDIA_PLAY_PAUSE:
    AppStreamInterface.keyPress(0xB3,down); break;
case KeyEvent.KEYCODE_ESCAPE:
case KeyEvent.KEYCODE_BACK:
    AppStreamInterface.keyPress(0x1b,down); break;
```

```
    default:
        {
            String key = new String(Character.toString((char)msg.getUnico
deChar()));

            keyCode = key.toUpperCase(Locale.US).codePointAt(0);
            AppStreamInterface.keyPress(keyCode,down);
        }
    }
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent msg) {
    onKey(msg,true);
    return super.onKeyDown(keyCode, msg);
}

@Override
public boolean onKeyUp(int keyCode, KeyEvent msg) {
    onKey(msg,false);
    return super.onKeyUp(keyCode, msg);
}
```

# Building a Chrome App

The Amazon AppStream Chrome App SDK contains the files for the example Chrome app. You can modify these files to create and publish your own Chrome app. This section assumes you are familiar with creating and publish a Chrome app. If this is your first time creating and publishing your own Chrome app, the following topics in the Chrome developer site will help you get started:

Create Your First App at https://developer.chrome.com/apps/first_app is a step-by-step tutorial on creating your first Chrome app.

Publish Your App at https://developer.chrome.com/apps/publish_app explains how to publish your app to the Chrome Web Store.

The *<SDK_dir>*\chrome_app directory in the Amazon AppStream Chrome App SDK contains the following directories and files:

| Directory or File | Description |
|---|---|
| `css` | Directory that contains the `.css` files required by the HTML files. |
| `js` | Directory that contains the JavaScript files required by the HTML files. |
| `media` | Directory that contains the graphic files required by the HTML files. |
| `AppStreamClient.nmf` | The native client manifest file that specifies the location of the portable native client file. |
| `AppStreamClient.pexe` | The native client file that communicates between the Chrome app and the streaming application. |

| Directory or File | Description |
|---|---|
| `manifest.json` | The manifest file that is required by the Chrome app. |
| `window.html` | The HTML file that displays the client. |

## Lifecycle of a Chrome App

The lifecycle of the Chrome app is described at App lifecycle at a glance at https://developer.chrome.com/apps/app_architecture#lifecycle in the Chrome Developer site.

## Modifying the Example Chrome App

The example Chrome app in the *<SDK_dir>*`\chrome_app` directory of the Amazon AppStream Chrome App SDK is composed of `.html`, `.css`, and JavaScript files. You can modify these files, just like any other web files.

## Choosing a Color Subsampling Rate

Amazon AppStream streams the video at the YUV420 color subsampling rate to client apps. If your streaming application, Google Chrome browser, or Chromebook support YUV444, then your Chrome app will display the video at YUV444. The example Chrome app in the Amazon AppStream Chrome App SDK is configured to accept and render a stream in the YUV444 color subsampling rate.

# Building a Client for iOS

The Amazon AppStream SDK provides client libraries that you build into your client to receive and decode video and audio from the application and send device inputs and raw inputs to the application.

The following object classes are provided by the client libraries in the Amazon AppStream SDK.

| Class | Description |
|---|---|
| XStxClientLibrary | The top level object of the client libraries. Your client uses this object to ensure that it is calling into the version of the libraries it was compiled against and to create the `XStxClient` object. |
| XStxClient | This object manages client sessions and sends events to your application when Amazon AppStream assigns your application a client session or terminates a client session. |

Your client implements and populate the following structures in order to receive session and application events.

| Interface | Description |
|---|---|
| XStxIClientListener | Receives callbacks from a client. |
| XStxIVideoDecoder | Receives and decodes the video frame. |
| XStxIVideoRenderer | Renders the video frame. |

| Interface | Description |
|---|---|
| XStxIRawVideoFrameAllocator | Receives and recycles the video frame. |
| XStxIAudioDecoder | Receives and decodes the audio frame. |
| XStxIAudioRenderer | Renders the audio frame. |
| XStxIRawAudioFrameAllocator | Receives and recycles the audio frame. |

# Lifecycle of a Client for iOS

Your client communicates with the application on Amazon AppStream through libraries of the Amazon AppStream SDK.

The lifecycle of a client is as follows:

1. **Create the `XStxClientLibraryHandle` object by calling the `XStxClientLibraryCreate` function to return a `clientLibraryHandle`.** The object interacts with the client side portion of the API through the `clientLibraryHandle`.
2. **Create the `XStxClient` object by calling the `XStxClientCreate` function to return a `XStxClientHandle` client handle.** The object manages the session between the client and the application.
3. **Instantiate and populate the `XStxIClientListener` structure.** This object responds to the callback functions from the application when a session is established between the client and the application. After populating the listener structure, call the `XStxClientSetListener` function with the client handle and a pointer to the `XStxIClientListener` structure.
4. **Initialize a video module to get, decode, and render the video frames.** To initialize this module, the client has to instantiate the following structures:

   - **`XStxIVideoDecoder`** Gets and decodes the video frame.
   - **`XStxIVideoRenderer`** Renders the video frame.
   - **`XStxIRawVideoFrameAllocator`** Gets and recycles the video frame.

5. **Initialize an audio module to get, decode, and render the audio frames.** To initialize this module, the client has to instantiate the following structures:

   - **`XStxIAudioDecoder`** Gets and decodes the audio frame.
   - **`XStxIAudioRenderer`** Renders the audio frame.
   - **`XStxIRawAudioFrameAllocator`** Gets and recycles the audio frame.

6. **Configure a session between the client and application.** Call the `XStxClientSetEntitlementUrl` function with the endpoint of the entitlement service to configure this session.
7. **Call the `XStxClientStart` function to start the session.** Once the session is established, the `XStxClient` object will invoke the `XStxIClientListenerFcnClientReady` function on the `XStxIClientListener` structure. You can then call the `xStxClientLibraryRecycle` function to recycle the client library handle.
8. **The client sends user inputs to the application**
9. **The client closes the session.**

# Building the Example Client for iOS

The Amazon AppStream SDK contains source code that you can build into a example client file. To learn how to build the source code, see the `\doc` directory.

# Creating Your Client

To create a client, create an `XStxClientLibraryHandle` object by using the `XStxClientLibraryCreate` function. The `XStxClientLibraryHandle` object is the top level object that you will use to interact with the libraries you will use to connect to sessions, render the audio and video, and send inputs to the application.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `initXStxClient` function of `AppStreamSampleClientViewController.m`.

```
-(XStxResult) initXStxClient
{
    XStxResult createResult = XSTX_RESULT_OK;


    /// create Singleton of client library handle
    createResult = XStxClientLibraryCreate(XSTX_CLIENT_API_VERSION_MAJOR,
                                           XSTX_CLIENT_API_VERSION_MINOR,
&mClientLibraryHandle);

    if (createResult != XSTX_RESULT_OK)
    {
        [self printResult:createResult withMessage:@"XStxClientLibraryCreate
failed"];
        return createResult;
    }

    return XSTX_RESULT_OK;
}
```

The function creates a client library handle (`clientLibraryHandle`) that the client will use to create a client object. The client then creates a client object by calling the `XStxClientCreate(mClientLibraryHandle` function with the client library handle. The function returns a handle to the client.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `connectXStxClient` function of `AppStreamSampleClientViewController.m`.

```
// Create a XStx Client instance
result = XStxClientCreate(mClientLibraryHandle, &mClientHandle);
if (result != XSTX_RESULT_OK)
{
    [self printResult:result withMessage:@"XStxClientCreate failed"];
    return;
}
```

The client object needs a structure to respond to the callback functions from the application. The client creates and populates this structure.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `H264ToYuv::init` function of `H264to Yuv.cpp`.

```
/** initialize */
XStxResult H264ToYuv::init()
{

    AvHelper::initialize();

    AVCodec *codec = avcodec_find_decoder(CODEC_ID_H264);
    if(NULL == codec)
    {
        return XSTX_RESULT_AUDIO_DECODER_NULL;
    }

    mCodecContext = avcodec_alloc_context3(codec);
    if(NULL == mCodecContext)
    {
        return XSTX_RESULT_VIDEO_DECODING_ERROR;
    }

    mCodecContext->codec_type           = AVMEDIA_TYPE_VIDEO;
    mCodecContext->codec_id             = CODEC_ID_H264;
    mCodecContext->pix_fmt              = PIX_FMT_YUV420P;

    if(avcodec_open2(mCodecContext, codec,NULL) < 0)
    {
        return XSTX_RESULT_VIDEO_DECODING_ERROR;
    }

    av_init_packet(&mAvPacket);

    return XSTX_RESULT_OK;
}
```

After populating the structure, call a function to configure a listener for callback functions.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `clientReady` function of `AppStreamSampleClientViewController.m`.

```
// start a client listener
if ((result = XStxClientSetListener(mClientHandle, &mClientListener))
    != XSTX_RESULT_OK)
{
    [self printResult:result withMessage:@"XStxClientSetListener failed"];
    return;
}
```

The client uses structures to get, render, and decode video and audio frames. The next step is to create and populate these structures. The example client instantiates a class to create and populate the structures.

To handle a video frame, the client create and populate the following structures defined in `VideoModule.h`:

- **`XStxIVideoDecoder`**. Used to get and decode a video frame.
- **`XStxIVideoRenderer`**. Used to render a video frame.

- **`XStxIRawVideoFrameAllocator`**. Used to get and recycle a video frame.

The following excerpt from the example client in the Amazon AppStream SDK illustrates populating the structures for video. The excerpt is from `VideoModule.m`.

```
bool VideoModule::initialize(XStxClientHandle mClientHandle,
                             VideoDecoder* decoder, GLKView  & rw)
{
    // initialize frame allocator


    if(decoder == nil)
    {
        return false;
    }

    mDecoder = decoder;

    mStxFrameAllocator.mInitFcn = &allocatorInit;
    mStxFrameAllocator.mInitCtx = this;
    mStxFrameAllocator.mGetVideoFrameBufferFcn = &allocatorGetFrame;
    mStxFrameAllocator.mGetVideoFrameBufferCtx = this;
    mStxFrameAllocator.mRecycleVideoFrameBufferFcn = &allocatorRecycleFrame;
    mStxFrameAllocator.mRecycleVideoFrameBufferCtx = this;
    mStxFrameAllocator.mSize = sizeof(mStxFrameAllocator);
    if (XStxClientSetVideoFrameAllocator(mClientHandle, &mStxFrameAllocator)
            != XSTX_RESULT_OK)
    {
        printf("Failed to SetVideoFrameAllocator.\n");
        return false;
    }


    mRenderer = &rw;

    mStxRenderer.mRenderVideoFrameFcn = &renderFrame;
    mStxRenderer.mRenderVideoFrameCtx = this;
    mStxRenderer.mSetMaxResolutionFcn = &rendererMaxResolution;
    mStxRenderer.mSetMaxResolutionCtx = this;
    mStxRenderer.mSize = sizeof(mStxRenderer);
    if (XStxClientSetVideoRenderer(mClientHandle, &mStxRenderer)
            != XSTX_RESULT_OK)
    {
        printf("Failed to set VideoRenderer.\n");
        return false;
    }


   ((GLViewManager *)mRenderer.delegate).decodeType = mDecoder->getDecodeType();


    mStxDecoder.mGetCapabilitiesCtx = mDecoder;
    mStxDecoder.mGetCapabilitiesFcn = &videoDecoderGetCapabilities;
    mStxDecoder.mDecodeVideoFrameFcn = &decodeFrame;
    mStxDecoder.mDecodeVideoFrameCtx = mDecoder;
    mStxDecoder.mStartFcn = &videoDecoderStart;
```

```
    mStxDecoder.mStartCtx = mDecoder;
    mStxDecoder.mSize = sizeof(mStxDecoder);
    if (XStxClientSetVideoDecoder(mClientHandle, &mStxDecoder)
            != XSTX_RESULT_OK)
    {
        printf("Failed to set SetVideoDecoder.\n");
        return false;
    }
    return true;
}
```

To handle an audio frame, the client create and populate the following structures defined in `AudioModule.h`:

- **`XStxIAudioDecoder`**. Used to get and decode an audio frame.
- **`XStxIAudioRenderer`**. Used to render an audio frame.
- **`XStxIRawAudioFrameAllocator`**. Used to get and recycle an audio frame.

The following excerpt from the example client in the Amazon AppStream SDK illustrates populating the structures for audio. The excerpt is from `AudioModule.m`.

```
/**
 * Initialize audio module
 * @param[in] clientHandle handle to XStx client
 */
bool AudioModule::initialize(XStxClientHandle clientHandle)
{
    mStxFrameAllocator.mInitFcn = &allocatorInit;
    mStxFrameAllocator.mInitCtx = this;
    mStxFrameAllocator.mGetAudioFrameBufferFcn = &allocatorGetFrame;
    mStxFrameAllocator.mGetAudioFrameBufferCtx = this;
    mStxFrameAllocator.mRecycleAudioFrameBufferFcn = &allocatorRecycleFrame;
    mStxFrameAllocator.mRecycleAudioFrameBufferCtx = this;
    mStxFrameAllocator.mSize = sizeof(mStxFrameAllocator);

    if (XStxClientSetAudioFrameAllocator(clientHandle, &mStxFrameAllocator)
            != XSTX_RESULT_OK)
    {
        printf("XStxClientSetAudioFrameAllocator() failed.\n");
        return false;
    }

    // initialize decoder
    mDecoder = new (std::nothrow) OpusToPcm();
    if (!mDecoder)
    {
        printf("Failed to create audio decoder.\n");
        return false;
    }

    mStxDecoder.mStartFcn = &decoderStart;
    mStxDecoder.mStartCtx = mDecoder;
    mStxDecoder.mDecodeAudioFrameFcn = &decoderDecodeFrame;
    mStxDecoder.mDecodeAudioFrameCtx = mDecoder;
```

```
    mStxDecoder.mSize = sizeof(mStxDecoder);
    if (XStxClientSetAudioDecoder(clientHandle, &mStxDecoder)
            != XSTX_RESULT_OK)
    {
        printf("XStxClientSetAudioDecoder() failed.\n");
        return false;
    }

    // initialize renderer
    mRenderer = new (std::nothrow) AudioRenderer(mFramePool,
        clientHandle);
    if (!mRenderer)
    {
        return false;
    }
    mStxRenderer.mStartFcn = &start;
    mStxRenderer.mStartCtx = mRenderer;
    mStxRenderer.mSize = sizeof(mStxRenderer);
    if (XStxClientSetAudioRenderer(clientHandle, &mStxRenderer)
            != XSTX_RESULT_OK)
    {
        printf("XStxClientSetAudioRenderer() failed. \n");
        return false;
    }

    return true;
}
```

The client is now ready to configure and start a session with the application. The client configures a session by calling a function with the Entitlement URL.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `connectXStxClient` function in `AppStreamSampleClientViewController.m`.

```
if ((result = XStxClientSetEntitlementUrl(mClientHandle, [url UTF8String]))
        != XSTX_RESULT_OK)
    {
        [self printResult:result withMessage:@"XStxClientSetEntitlementUrl
failed"];
        return;
    }
```

The client starts a session by calling the `XStxClientStart` function.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `connectXStxClient` function in `AppStreamSampleClientViewController.m`.

```
    // this method is non-blocking
    // defer XStxClientLibraryRecycle call for unload methods
    if ((result = XStxClientStart(mClientHandle)) != XSTX_RESULT_OK)
    {
        [self printResult:result withMessage:@"XStxClientStart failed"];
        return;
    }
```

```
    NSString *message = [NSString stringWithFormat:@"STX client initialized.
Connecting to %@:",url];
    [self printResult:result withMessage:message];

    [self hideBackground];
    [self setupInputs];
```

# Choosing a Color Subsampling Rate

Amazon AppStream streams the video at the YUV420 color subsampling rate to client applications. If your streaming application, client application and device support YUV444, you can set your client application to display the video at YUV444. The example client application in Amazon AppStream SDK is configured to accept and render a stream in the YUV444 color subsampling rate.

When the client application connects to the streaming application, Amazon AppStream compares the color subsampling options available on the client application with the options advertised on the streaming application. Amazon AppStream then selects the highest color resolution supported by both the client application and streaming application. Amazon AppStream then calls the `XStxIClientListener2FcnSetConfiguration` callback function that the client application supplied and passes the structure with the *`XStxChromaSampling`* setting.

The following excerpt from `H264ToYUV.cpp` illustrates this step. This file is in the *<SDK_dir>*`\example_src\client\src\apple\ios` directory.

```
bool H264ToYuv::isChromaSamplingSupported(XStxChromaSampling chromaSampling)
{
    return chromaSampling == XSTX_CHROMA_SAMPLING_YUV420
        || chromaSampling == XSTX_CHROMA_SAMPLING_YUV444;
}
```

The following excerpt from `VideoPipeline.cpp` illustrates this step. This file is in the *<SDK_dir>*`\example_src\client\src\apple\ios` directory.

```
decoderCapabilities->mProfile = XSTX_H264_PROFILE_HIGH444;
```

# Sending Your Client Inputs to the Application

The client can send key, and touch inputs to the application.

The following excerpt from the example client in the Amazon AppStream SDK illustrates how keyboard events are handled. The excerpt is from the `setupInputs` function of `AppStreamSampleClientViewController.m`.

```
-(void) setupInputs
{
    // KEYBOARD EVENTS -- pass through iOS key presses

    CGFloat y = 30.0f;
```

```
    // add a button for keyboard input
    UIButton *kbbutton = [UIButton buttonWithType:UIButtonTypeCustom];
    [kbbutton addTarget:self action:@selector(toggleKeyboard:) forContro
lEvents:UIControlEventTouchUpInside];
    CGFloat buttonWidth = 72.0f;
    kbbutton.frame = CGRectMake(self.view.bounds.size.width-buttonWidth,
                                y,buttonWidth,buttonWidth);
    [kbbutton setImage:[UIImage imageNamed:@"button-keyboard"] forState:UICon
trolStateNormal];
    kbbutton.autoresizingMask = UIViewAutoresizingFlexibleBottomMar
gin|UIViewAutoresizingFlexibleLeftMargin;
    [self.view addSubview:kbbutton];

    // add offscreen text field for keyboard input and set delegate
    // note: text field is off screen to hide input from user
   mInputText = [[UITextField alloc]initWithFrame:CGRectMake(-500,-500,100,20)];

    [self.view addSubview:mInputText];
    mInputText.delegate = self;

    /// SENDING TOUCH AS MOUSE EVENTS
    mTreatTouchesAsMouse = NO;
    // add a button for mouse input
    UIButton *mousebutton = [UIButton buttonWithType:UIButtonTypeCustom];
    [mousebutton addTarget:self action:@selector(toggleMouse:) forContro
lEvents:UIControlEventTouchUpInside];
   mousebutton.frame = CGRectMake(self.view.bounds.size.width-(buttonWidth*2),

                                  y,buttonWidth,buttonWidth);
    [mousebutton setImage:[UIImage imageNamed:@"button-mouse"] forState:UICon
trolStateNormal];
    mousebutton.autoresizingMask = UIViewAutoresizingFlexibleBottomMar
gin|UIViewAutoresizingFlexibleLeftMargin;
    [self.view addSubview:mousebutton];

    mShouldTrackGesture = false;
    UIButton *handbutton = [UIButton buttonWithType:UIButtonTypeCustom];
    [handbutton addTarget:self action:@selector(toggleGesture:) forContro
lEvents:UIControlEventTouchUpInside];
    handbutton.frame = CGRectMake(self.view.bounds.size.width-(buttonWidth*3),

                                  y,buttonWidth,buttonWidth);
   [handbutton setImage:[UIImage imageNamed:@"button-hand"] forState:UIControl
StateNormal];
    handbutton.autoresizingMask = UIViewAutoresizingFlexibleBottomMar
gin|UIViewAutoresizingFlexibleLeftMargin;
    [self.view addSubview:handbutton];

    mPanGestureRecognizer = [[UIPanGestureRecognizer alloc]initWithTarget:self
 action:@selector(handlePanGesture:)];
    mPanGestureRecognizer.enabled = false;
    [self.view addGestureRecognizer:mPanGestureRecognizer];
}
```

The following excerpt from the example client in the Amazon AppStream SDK illustrates how gesture events are handled. The excerpt is from the `handlePanGesture` function of `AppStreamSampleClientViewController.m`.

```
void) handlePanGesture:(UIPanGestureRecognizer*) gestureRecognizer
{

    CGPoint location = [gestureRecognizer locationInView:self.view];

    NSString *jsonString = [NSString stringWith
Format:@"{\"state\":\"%d\",\"xy\":[%.2f,%.2f]}",gestureRecognizer.state,loca
tion.x, location.y ];


    switch (gestureRecognizer.state) {
        case UIGestureRecognizerStateEnded:
            [self showStatus:@""];
            break;
        case UIGestureRecognizerStatePossible:
        case UIGestureRecognizerStateBegan:
        case UIGestureRecognizerStateChanged:
        case UIGestureRecognizerStateFailed:
        case UIGestureRecognizerStateCancelled:
        default:
            [self sendRawInput:jsonString];
            break;
    }

}
```

Touch events are handled as mouse events.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from `AppStreamSampleClientViewController.m`.

```
// In this example we send the touch events as mouse input
-(void) toggleMouse:(UIButton*)sender
{
    if(mShouldTrackGesture)
    {
        mShouldTrackGesture = false;
        mPanGestureRecognizer.enabled = false;
        mTreatTouchesAsMouse = true;

    } else {

        if( mTreatTouchesAsMouse)
        {
            mTreatTouchesAsMouse = false;
            [self showStatus:@"touches passed as touches"];
        }
        else
        {
            mTreatTouchesAsMouse = true;
            [self showStatus:@"mouse input emulated by touch "];
        }
    }
}
```

```
#pragma mark UIResponder touch methods

-(void) touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    if (!mTreatTouchesAsMouse)
    {
        [self sendTouchEvent: XSTX_TOUCH_DOWN withTouches: touches withEvent:
event];
        return;
    }

    UITouch *touch = [touches anyObject];
    [self sendMouseEvent:[touch locationInView:self.view] flags:RI_MOUSE_BUT
TON_1_DOWN];
}

-(void) touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event
{
    //When passing touches as touches we need to separate out touch_cancelled
    if (!mTreatTouchesAsMouse)
    {
        [self sendTouchEvent: XSTX_TOUCH_CANCELLED withTouches: touches
withEvent: event];
        return;
    }
    //Handle touchesCancelled exactly like touchesEnded
    [self touchesEnded:touches withEvent:event];
}

-(void) touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
{
    if (!mTreatTouchesAsMouse)
    {
        [self sendTouchEvent: XSTX_TOUCH_UP withTouches: touches withEvent:
event];
        return;
    }

    UITouch *touch = [touches anyObject];
    [self sendMouseEvent:[touch locationInView:self.view] flags:RI_MOUSE_BUT
TON_1_UP];
    //After the mouseUp move the mouse way offscreen to prevent any unwanted
mouse hover effects
    [self sendMouseEvent:CGPointMake(-10000, -10000) flags:0];

    // clear status
    [self showStatus:@""];

}

-(void) touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
{
    if (!mTreatTouchesAsMouse)
    {
        [self sendTouchEvent: XSTX_TOUCH_MOVE withTouches: touches withEvent:
event];
        return;
    }
```

```
    UITouch *touch = [touches anyObject];
    [self sendMouseEvent:[touch locationInView:self.view] flags:0];
}

#pragma mark XStx touch event handling

-(void) sendTouchEvent: (XStxTouchType) theType withTouches: (NSSet *) touches
 withEvent: (UIEvent *) event
{
    if (mWidthScale==0)
    {
        return;
    }

    // here we are just going to emulate the mouse as a touch event,
    // just to demonstrate how to use it
    XStxInputEvent inputEvent;
    inputEvent.mTimestampUs = CACurrentMediaTime() * 1000000.0;
    inputEvent.mUserId = 0;
    inputEvent.mType = XSTX_INPUT_EVENT_TYPE_TOUCH;

    //How many touches we are sending
    inputEvent.mInfo.mTouch.mPointerCount = [touches count];
    //Array to hold the pointer data
    inputEvent.mInfo.mTouch.mPointers = new XStxPointer[[touches count]];

    int currPointer = 0;
    for (UITouch *currTouch in touches) {
        //Convert the touch location from the view to the server coordinates
        CGPoint touchLoc = [currTouch locationInView:self.view];
        //Make sure touchLoc is within the viewport area
        touchLoc = CGPointMake(MIN(MAX(touchLoc.x - mVideoRect.origin.x, 0),
mVideoRect.size.width), MIN(MAX(touchLoc.y - mVideoRect.origin.y, 0), mVideoR
ect.size.height));

        float scaledX = (touchLoc.x * mWidthScale);
        float scaledY = (touchLoc.y * mHeightScale);
        //Use the address of the touchObject as the pointerID
        inputEvent.mInfo.mTouch.mPointers[currPointer].mPointerId =
(uint64_t)currTouch;
        //Set the X & Y
        inputEvent.mInfo.mTouch.mPointers[currPointer].mX = scaledX;
        inputEvent.mInfo.mTouch.mPointers[currPointer].mY = scaledY;
        //No pressure support so always pass 1.0
        inputEvent.mInfo.mTouch.mPointers[currPointer].mPressure = 1.0f;

        inputEvent.mInfo.mTouch.mPointers[currPointer].mTouchType = theType;
        currPointer++;
    }
    inputEvent.mSize = sizeof(inputEvent);
    inputEvent.mDeviceId = 0;

    //Send the touches to the server
    [self sendInput:inputEvent];

    //Clean up the mPointers array
    delete [] inputEvent.mInfo.mTouch.mPointers;
```

```
}

// fill out XStxInputEvent struct emulating win32 RAWMOUSE
-(void) sendMouseEvent:(CGPoint ) xy flags:(uint32_t)flags
{
 if (mWidthScale==0)
 {
     return;
 }

 //Make sure xy is within the viewport area
 xy = CGPointMake(MIN(MAX(xy.x - mVideoRect.origin.x, 0), mVideoRect.size.width),
 MIN(MAX(xy.y - mVideoRect.origin.y, 0), mVideoRect.size.height));

 int scaledX = (xy.x * mWidthScale);
 int scaledY = (xy.y * mHeightScale);

 XStxInputEvent xstxevent = { 0 };
 xstxevent.mTimestampUs = CACurrentMediaTime() * 1000000.0;
 xstxevent.mType = XSTX_INPUT_EVENT_TYPE_MOUSE;
 xstxevent.mInfo.mMouse.mLastX = scaledX;
 xstxevent.mInfo.mMouse.mLastY = scaledY;
 xstxevent.mInfo.mMouse.mButtonFlags = flags ;
 xstxevent.mInfo.mMouse.mFlags = 1; //absolute, 0 would be relative
 xstxevent.mInfo.mMouse.mButtons = 0;// not needed
 [self sendInput:xstxevent];
 NSString *outputString = [NSString stringWithFormat:@"sending mouse event with
 coordinates:x:%i y:%i",scaledX,scaledY];
 [self showStatus:outputString];
}
```

# Terminating Your Client

The client ends the session by calling the `XStxClientLibraryRecycle` function and then free the resources.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `recycleXStxClient` and `dealloc` functions of `AppStreamSampleClientViewController.m`.

```
-(void) recycleXStxClient
{
    XStxClientLibraryRecycle(mClientLibraryHandle);
}

// clean up
-(void) dealloc {
    [[NSNotificationCenter defaultCenter] removeObserver:self];
    [self recycleXStxClient];
}
```

# Building a Client for OS X

The Amazon AppStream SDK provides client libraries that you build into your client to receive and decode video and audio from the application and send device inputs and raw inputs to the application.

The following object classes are provided by the client libraries in the Amazon AppStream SDK.

| Class | Description |
|---|---|
| `XStxClientLibrary` | The top level object of the client libraries. Your client uses this object to ensure that it is calling into the version of the libraries it was compiled against and to create the `XStxClient` object. |
| `XStxClient` | This object manages client sessions and sends events to your application when Amazon AppStream assigns your application a client session or terminates a client session. |

Your client implements and populates the following structures in order to receive session and application events.

| Interface | Description |
|---|---|
| `XStxIClientListener` | Receives callbacks from a client. |
| `XStxIVideoDecoder` | Receives and decodes the video frame. |
| `XStxIVideoRenderer` | Renders the video frame. |
| `XStxIRawVideoFrameAllocator` | Receives and recycles the video frame. |
| `XStxIAudioDecoder` | Receives and decodes the audio frame. |
| `XStxIAudioRenderer` | Renders the audio frame. |
| `XStxIRawAudioFrameAllocator` | Receives and recycles the audio frame. |

## Lifecycle of a Client for OS X

Your client communicates with the application on Amazon AppStream through libraries of the Amazon AppStream SDK.

The lifecycle of a client is as follows:

1. **Create the `XStxClientLibraryHandle` object by calling the `XStxClientLibraryCreate` function to return a `clientLibraryHandle`.** The object interacts with the client side portion of the API through the `clientLibraryHandle`.
2. **Create the `XStxClient` object by calling the `XStxClientCreate` function to return a `XStxClientHandle` client handle.** The object manages the session between the client and the application.
3. **Instantiate and populate the `XStxIClientListener` structure.** This object responds to the callback functions from the application when a session is established between the client and the application. After populating the listener structure, call the `XStxClientSetListener` function with the client handle and a pointer to the `XStxIClientListener` structure.

4. **Initialize a video module to get, decode, and render the video frames.** To initialize this module, the client has to instantiate the following structures:

   - **`XStxIVideoDecoder`** Gets and decodes the video frame.
   - **`XStxIVideoRenderer`** Renders the video frame.
   - **`XStxIRawVideoFrameAllocator`** Gets and recycles the video frame.

   The Amazon AppStream SDK contains `VideoPipeline.cpp` which shows how to create the `VideoDecoder` and `VideoRenderer` objects. The source code is in `<install_dir>\example_src\client\src\apple\osx`. The implementation of `VideoDecoder` in the example client attempts to use `VDADecoder` for hardware accelerated H.264 video decoding. If the hardware decoder is not available, then the client will use the software decoder FFmpeg.
5. **Initialize an audio module to get, decode, and render the audio frames.** To initialize this module, the client has to instantiate the following structures:

   - **`XStxIAudioDecoder`** Gets and decodes the audio frame.
   - **`XStxIAudioRenderer`** Renders the audio frame.
   - **`XStxIRawAudioFrameAllocator`** Gets and recycles the audio frame.

   The Amazon AppStream SDK contains `AudioPipeline.cpp` which shows how to create the `AudioDecoder` and `AudioRenderer` objects. The source code is in `<install_dir>\example_src\client\src\apple\shared`.
6. **Configure a session between the client and application.** Call the `XStxClientSetEntitlementUrl` with the endpoint of the entitlement service to configure this session.
7. **Call the `XStxClientStart` function to start the session.** Once the session is established, the `XStxClient` object will invoke the `XStxIClientListenerFcnClientReady` function on the `XStxIClientListener` structure. You can then call `xStxClientLibraryRecycle` to recycle the client library handle.
8. **The client sends user inputs to the application**
9. **The client closes the session.**

# Building the Example Client for OS X

The Amazon AppStream SDK contains source code that you can build into a example client application. To learn how to build the source code, see the `\doc` directory.

# Creating Your Client

To create a client, your client must create an `XStxClientLibraryHandle` object. This is the top level object that you will use to interact with the libraries you will use to connect to sessions, render the audio and video, and send inputs to the application.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::init()` function of `XStxModule.cpp`.

> **Note**
> We strongly recommend using the provided source code and pre-compiled clients rather than creating your own implementation based on these excerpts.

```
XStxResult XStxModule::init()
```

```
{
    XStxResult createResult = XSTX_RESULT_OK;
    if ((createResult = XStxClientLibraryCreate(XSTX_CLIENT_API_VERSION_MAJOR,

                                                XSTX_CLIENT_API_VERSION_MINOR,
 &mClientLibraryHandle)) != XSTX_RESULT_OK)
    {
        const char *name; const char *desc;
        XStxResultGetInfo(createResult, &name, &desc);
        LOGE("XStxClientLibraryCreate failed with: %s", name);
        return createResult;
    }

    mVideoRenderer = newVideoRenderer();

    return XSTX_RESULT_OK;
}
```

The function creates a client library handle (`mclientLibraryHandle`) that the client will use to create a client object. The example client creates the client object by instantiating an object from a user-defined class.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::connect` function of `XStxModule.cpp`.

```
if ((result = XStxClientCreate(mClientLibraryHandle, &mClientHandle))
        != XSTX_RESULT_OK)
    {
        LOGE("Failed to create client.");
        const char *name; const char *desc;
        XStxResultGetInfo(result, &name, &desc);
        LOGE("XStxClientCreate failed with: %s", name);
        return result;
    }
```

The object needs a structure to respond to the callback functions from the application. The client creates and populates a `XStxIClientListener` structure to respond to the callback functions.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::connect` function of `XStxModule.cpp`.

```
...
memset(&mStxListener, 0, sizeof(mStxListener));
...

mStxListener.mClientReadyFcn = &::clientReady;
mStxListener.mClientReadyCtx = this;
mStxListener.mClientStoppedFcn = &::clientStopped;
mStxListener.mClientStoppedCtx = this;
mStxListener.mMessageReceivedFcn = &::messageReceived;
mStxListener.mMessageReceivedCtx = this;
mStxListener.mStreamQualityMetricsReceivedFcn = &::clientQoS;
mStxListener.mStreamQualityMetricsReceivedCtx = this;
mStxListener.mSize = sizeof(mStxListener);
```

After populating the structure, call the `XStxClientSetListener` function to configure a listener that responds to the callback functions.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::connect` function of `XStxModule.cpp`.

```
if ((result = XStxClientSetListener(mClientHandle, &mStxListener))
    != XSTX_RESULT_OK)
{
    LOGE("Failed to set listener");
    const char *name; const char *desc;
    XStxResultGetInfo(result, &name, &desc);
    LOGE("XStxClientSetListener failed with: %s", name);
    platformErrorMessage(true, desc);
    return result;
}
```

The client uses structures to get, render, and decode video and audio frames. The next step is to create and populate these structures. The example client instantiates a class to create and populate the structures.

To handle a video frame, the client create and populate the following structures defined in `VideoModule.h`:

- **`XStxIVideoDecoder`**. Used to get and decode a video frame.
- **`XStxIVideoRenderer`**. Used to render a video frame.
- **`XStxIRawVideoFrameAllocator`**. Used to get and recycle a video frame.

The following excerpt from the example client in the Amazon AppStream SDK illustrates populating the structures for video. The excerpt is from the `XStxModule::connect` function of `XStxModule.cpp`.

```
// initialize video module
if (!mVideoModule.initialize(mClientHandle, *mVideoRenderer))
{
    LOGE("Failed to create video decoder/renderer");
    return XSTX_RESULT_NOT_INITIALIZED_PROPERLY;
}
```

To handle an audio frame, the client create and populate the following structures defined in `AudioModule.h`:

- **`XStxIAudioDecoder`**. Used to get and decode an audio frame.
- **`XStxIAudioRenderer`**. Used to render an audio frame.
- **`XStxIRawAudioFrameAllocator`**. Used to get and recycle an audio frame.

The following excerpt from the example client in the Amazon AppStream SDK illustrates populating the structures for audio. The excerpt is from `XStxModule::connect` function of `XStxModule.cpp`.

```
// initialize audio module
if (!mAudioModule.initialize(mClientHandle))
{
    LOGE("Failed to create audio decoder/renderer");
```

```
    return XSTX_RESULT_NOT_INITIALIZED_PROPERLY;
}
```

The client is now ready to configure and start a session with the application. The client configures a session by calling the `XStxClientSetEntitlementUrl` function with the endpoint of the entitlement service.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::connect` function of `XStxModule.cpp`.

```
if ((result = XStxClientSetEntitlementUrl(mClientHandle, address.c_str()))
    != XSTX_RESULT_OK)
{
    const char *name; const char *desc;
    XStxResultGetInfo(result, &name, &desc);
    LOGE("XStxClientSetEntitlementUrl failed with: %s", name);
    return result;
}
```

The client starts a session by calling a function.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::connect` function of `XStxModule.cpp`.

```
// non-blocking!
if ((result = XStxClientStart(mClientHandle)) != XSTX_RESULT_OK)
{
    LOGE("Failed to start client.");
    const char *name; const char *desc;
    XStxResultGetInfo(result, &name, &desc);
    LOGE("XStxClientStart failed with: %s", name);
    return result;
}
```

# Choosing a Color Subsampling Rate

Amazon AppStream streams the video at the YUV420 color subsampling rate to client applications. If your streaming application, client application and device support YUV444, you can set your client application to display the video at YUV444. The example client application in Amazon AppStream SDK is configured to accept and render a stream in the YUV444 color subsampling rate.

When the client application connects to the streaming application, Amazon AppStream compares the color subsampling options available on the client application with the options advertised on the streaming application. Amazon AppStream then selects the highest color resolution supported by both the client application and streaming application. Amazon AppStream then calls the `XStxIClientListener2FcnSetConfiguration` callback function that the client application supplied and passes the structure with the *`XStxChromaSampling`* setting.

The following excerpt from `H264ToYUV.cpp` illustrates this step. This file is in the *<SDK_dir>*`\example_src\client\src\apple\osx` directory.

```
bool H264ToYuv::isChromaSamplingSupported(XStxChromaSampling chromaSampling)
{
    return chromaSampling == XSTX_CHROMA_SAMPLING_YUV420
        || chromaSampling == XSTX_CHROMA_SAMPLING_YUV444;
}
```

The following excerpt from `VideoPipeline.cpp` illustrates this step. This file is in the *<SDK_dir>*`\example_src\client\src\apple\osx` directory.

```
decoderCapabilities->mProfile = XSTX_H264_PROFILE_HIGH444;
```

# Sending Your Client Inputs to the Application

The client can send keyboard, mouse, touch, or raw inputs to the application. The client sends an input by filling out the `XStxInputEvent` structure that describes the input and then calling a function that passes the structure to the application.

## Keyboard

Before the client can fill out the `XStxInputEvent` structure, the keyboard input must first be converted to a Windows keyboard input from an OS X keyboard input. In the `processKeyEvent` function is the call to `getConvertedKeyUsingCharacter` which converts the keyboard input to a Windows keyboard input.

The following excerpt from the example client in the Amazon AppStream SDK illustrates the function that converts the OS X keyboard input to a Windows keyboard input. The excerpt is from the `getConvertedKeyUsingCharacter` function in `AppStreamSampleClientWindowController.m`.

```
(int) getConvertedKeyUsingCharacter: (NSEvent *) theEvent withModifierMask:
(uint16_t &) modifierKeyMask withModifierKeyIgnoreMask: (uint16_t &) modifier
KeysIgnoreMask
{
    //Get the characters from the key being pressed
    NSString *theKeyString = [theEvent charactersIgnoringModifiers];

    if ([theKeyString length] <= 0) {
        //Reject dead keys
        return -1;
    }

    //Get the characters as a char
    uint16_t theChar = [theKeyString characterAtIndex:0];

    //First check the generic key mapping, this should convert any non-system
    // specific keys
   bool didMapKey = getVirtualKeyUsingChar(theChar, modifierKeyMask, modifier
KeysIgnoreMask);

    if (!didMapKey) {
        //The generic key mapping didn't map the key so check the platform
specific ones
```

```
       didMapKey = appleGetVirtualKeyUsingChar(theChar, modifierKeyMask, modi
fierKeysIgnoreMask);
    }

    if (!didMapKey) {
        //Neither the generic nor the system specific mapping could remap the
key
        NSLog(@"\n======== Unmapped KEY! =========");
        unichar upperKeyChar = [[theKeyString uppercaseString] characterAtIn
dex:0];
        NSString *modKeyString = [theEvent characters];

        NSLog(@"Key: %hu String: %@ Mod: %@", upperKeyChar, theKeyString, mod
KeyString);

        return -1;
    }

    return theChar;
}
```

The following excerpt from the example client in the Amazon AppStream SDK illustrates the function that processes the keyboard action from the client. The excerpt is from the `processKeyEvent` function in `AppStreamSampleClientWindowController.m`.

```
(void) processKeyEvent: (NSEvent *) theEvent
{
    //Depending on your application you might want to block repeat keys
    //    if ([theEvent isARepeat]) {
    //        //Ignore repeats
    //        return;
    //    }

    //Setup a couple bitmasks for the command keys
    uint16_t modifierKeyMask = 0;
    uint16_t modifierKeysIgnoreMask = 0;
    int newKey = [self getConvertedKeyUsingCharacter:theEvent withModifier
Mask:modifierKeyMask withModifierKeyIgnoreMask:modifierKeysIgnoreMask];

    if (newKey < 0)
    {
        //Not a valid key mapping just return
        return;
    }

    BOOL needAlt = (modifierKeyMask & STX_ALT_KEY_MASK) == STX_ALT_KEY_MASK;
    BOOL needControl = (modifierKeyMask & STX_CONTROL_KEY_MASK) == STX_CON
TROL_KEY_MASK;
    BOOL needShift = (modifierKeyMask & STX_SHIFT_KEY_MASK) ==
STX_SHIFT_KEY_MASK;
    BOOL needWin = (modifierKeyMask & STX_WINDOWS_KEY_MASK) == STX_WIN
DOWS_KEY_MASK;

    //Check the status of the command keys
    if ((modifierKeysIgnoreMask & STX_SHIFT_KEY_MASK) != STX_SHIFT_KEY_MASK) {
```

```
        if (needShift && !shiftIsDown) {
            //Need shift pressed but it isn't
            [[AppStreamClient sharedClient] sendKeyDown:VK_SHIFT];
        } else if (!needShift && shiftIsDown)
        {
            //Need shift released but it is pressed
            [[AppStreamClient sharedClient] sendKeyUp:VK_SHIFT];
        }
    }

    if ((modifierKeysIgnoreMask & STX_ALT_KEY_MASK) != STX_ALT_KEY_MASK) {
        if (needAlt && !altIsDown) {
            [[AppStreamClient sharedClient] sendKeyDown:VK_MENU];
        } else if (!needAlt && altIsDown)
        {
            [[AppStreamClient sharedClient] sendKeyUp:VK_MENU];
        }
    }

   if ((modifierKeysIgnoreMask & STX_CONTROL_KEY_MASK) != STX_CONTROL_KEY_MASK)
{
        if (needControl && !controlIsDown) {
            [[AppStreamClient sharedClient] sendKeyDown:VK_CONTROL];
        } else if (!needControl && controlIsDown)
        {
            [[AppStreamClient sharedClient] sendKeyUp:VK_CONTROL];
        }
    }

   if ((modifierKeysIgnoreMask & STX_WINDOWS_KEY_MASK) != STX_WINDOWS_KEY_MASK)
{
        if (needWin && !windowsIsDown) {
            [[AppStreamClient sharedClient] sendKeyDown:VK_LWIN];
        } else if (!needWin && windowsIsDown)
        {
            [[AppStreamClient sharedClient] sendKeyUp:VK_LWIN];
        }
    }

    //Now send the actual keydown event
    if (theEvent.type == NSKeyDown) {
        [[AppStreamClient sharedClient] sendKeyDown:newKey];
    } else
    {
        //Not keyDown so must be keyUp
        [[AppStreamClient sharedClient] sendKeyUp:newKey];
    }

    //Reset the command keys state
    if ((modifierKeysIgnoreMask & STX_SHIFT_KEY_MASK) != STX_SHIFT_KEY_MASK) {

        if (needShift && !shiftIsDown) {
            //Need shift pressed but it isn't
            [[AppStreamClient sharedClient] sendKeyUp:VK_SHIFT];
        } else if (!needShift && shiftIsDown)
        {
            //Need shift released but it is pressed
            [[AppStreamClient sharedClient] sendKeyDown:VK_SHIFT];
```

```
        }
    }

    if ((modifierKeysIgnoreMask & STX_ALT_KEY_MASK) != STX_ALT_KEY_MASK) {
        if (needAlt && !altIsDown) {
            [[AppStreamClient sharedClient] sendKeyUp:VK_MENU];
        } else if (!needAlt && altIsDown)
        {
            [[AppStreamClient sharedClient] sendKeyDown:VK_MENU];
        }
    }

   if ((modifierKeysIgnoreMask & STX_CONTROL_KEY_MASK) != STX_CONTROL_KEY_MASK)
 {
        if (needControl && !controlIsDown) {
            [[AppStreamClient sharedClient] sendKeyUp:VK_CONTROL];
        } else if (!needControl && controlIsDown)
        {
            [[AppStreamClient sharedClient] sendKeyDown:VK_CONTROL];
        }
    }

   if ((modifierKeysIgnoreMask & STX_WINDOWS_KEY_MASK) != STX_WINDOWS_KEY_MASK)
 {
        if (needWin && !windowsIsDown) {
            [[AppStreamClient sharedClient] sendKeyUp:VK_LWIN];
        } else if (!needWin && windowsIsDown)
        {
            [[AppStreamClient sharedClient] sendKeyDown:VK_LWIN];
        }
    }
}
```

After converting the keyboard input, the client populates the `XStxInputEvent` structure. The structure is then sent to AppStream.

The following excerpt from the example client in the Amazon AppStream SDK illustrates a keyboard action that populates the structures and then sending that structure to the application. The excerpt is from the `XStxModule::keyPress` function in `XStxModule.cpp`.

```
void XStxModule::keyPress(int key, bool down)
{
    XStxInputEvent xstxevent = { 0 };
    xstxevent.mTimestampUs = mud::TimeVal::mono().toMilliSeconds();
    xstxevent.mType = XSTX_INPUT_EVENT_TYPE_KEYBOARD;
    xstxevent.mInfo.mKeyboard.mVirtualKey = key;
    xstxevent.mInfo.mKeyboard.mIsKeyDown = down;
    xstxevent.mSize = sizeof(XStsxInputEvent);

    sendInput(xstxevent);
}
```

## Mouse

The following excerpt from the example client in the Amazon AppStream SDK illustrates how the mouse actions are sent to the structure. The excerpt is from `AppStreamSampleClientWindowControll.m`.

```
(void) mouseDown:(NSEvent *)theEvent
{
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:RI_MOUSE_BUT
TON_1_DOWN];
}

(void) mouseUp:(NSEvent *)theEvent
{
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:RI_MOUSE_BUT
TON_1_UP];
}

(void) mouseDragged:(NSEvent *)theEvent
{
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:0];
}

(void) rightMouseDown:(NSEvent *)theEvent
{
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:RI_MOUSE_BUT
TON_2_DOWN];
}

(void) rightMouseUp:(NSEvent *)theEvent
{
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:RI_MOUSE_BUT
TON_2_UP];
}

(void) rightMouseDragged:(NSEvent *)theEvent
{
```

```
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:0];
}

(void) mouseMoved:(NSEvent *)theEvent
{
    NSPoint theLoc = [theEvent locationInWindow];

    theLoc.y = _glView.bounds.size.height - theLoc.y;

    [[AppStreamClient sharedClient] sendMouseEvent:theLoc flags:0];
}
```

The client then populates the `XStxInputEvent` structure. The structure is then sent to AppStream.

The following excerpt from the example client in the Amazon AppStream SDK illustrates a mouse action that populates the structures and then sends that structure to the application. The excerpt is from the `XStxModule::mouseEvent` function in `XStxModule.cpp`.

```
void XStxModule::mouseEvent(int x, int y, uint32_t flags)
{
    // no video renderer? Return!
    if (!mVideoRenderer)
    {
        return;
    }

    int xOff, yOff;
    float scale;
    mVideoRenderer->getScaleAndOffset(scale, xOff, yOff);

    // no scale yet; ignore!
    if (scale == 0) return;

    XStxInputEvent xstxevent = { 0 };
    xstxevent.mTimestampUs = mud::TimeVal::mono().toMilliSeconds();
    xstxevent.mType = XSTX_INPUT_EVENT_TYPE_MOUSE;
    xstxevent.mInfo.mMouse.mButtonFlags = flags;
    xstxevent.mInfo.mMouse.mFlags = 1; //absolute

    static int lastX = 0, lastY = 0;

    if (flags & CET_MOUSE_WHEEL)
    {
        LOGV("Mouse wheel data: %d", x);
        // mouse wheel data goes in mButtonData.
        xstxevent.mInfo.mMouse.mButtonData = x;
        xstxevent.mInfo.mMouse.mLastX = lastX;
        xstxevent.mInfo.mMouse.mLastY = lastY;
    }
    else
    {
        lastX = xstxevent.mInfo.mMouse.mLastX = (int)((x + xOff) * scale);
```

```
        lastY = xstxevent.mInfo.mMouse.mLastY = (int)((y + yOff) * scale);
//        LOGV("Mouse data: x:%d,y:%d", lastX, lastY);
    }

    sendInput(xstxevent);
}
```

## Terminating Your Client

The client can end the session with the application in the following ways:

- End the session and then confirm the session ended.
- End the session without regard as to when the session ends.

To end the session and then confirm the session ended, first call `XStxClientStop`. This is a non-blocking function call that immediately returns a result. If the call was successful, then call `XStxClientWait` to wait until the session actually ends. When the `XStxClientWait` call is successful, call `XStxClientRecycle` to recycle the client handle.

To end the session without regard as to when the session ends, call `XStxClientRecycle`. The session then ends without further interaction from the client.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `XStxModule::recycle` function in `XStxModule.cpp`.

```
XStxResult XStxModule::recycle()
{
XStxResult result = XStxClientRecycle(mClientHandle);
if (result != XSTX_RESULT_OK)
    return result;
else
    return XStxClientLibraryRecycle(mClientLibraryHandle);
}
```

# Building a Client for Windows

The Amazon AppStream SDK provides client libraries that you build into your client to receive and decode video and audio from the application and send device inputs and raw inputs to the application.

The following object classes are provided by the client libraries in the Amazon AppStream SDK.

| Class | Description |
|---|---|
| XStxClientLibrary | The top level object of the client libraries. Your client uses this object to ensure that it is calling into the version of the libraries it was compiled against and to create the `XStxClient` object. |
| XStxClient | This object manages client sessions and sends events to your application when Amazon AppStream assigns your application a client session or terminates a client session. |

Your client implements and populate the following structures in order to receive session and application events.

| Interface | Description |
|---|---|
| XStxIClientListener | Receives callbacks from a client. |
| XStxIVideoDecoder | Receives and decodes the video frame. |
| XStxIVideoRenderer | Renders the video frame. |
| XStxIRawVideoFrameAllocator | Receives and recycles the video frame. |
| XStxIAudioDecoder | Receives and decodes the audio frame. |
| XStxIAudioRenderer | Renders the audio frame. |
| XStxIRawAudioFrameAllocator | Receives and recycles the audio frame. |

The Amazon AppStream SDK includes source code so that you can build into a example client. The documentation is in the `\doc` directory.

# Lifecycle of a Client for Windows

Your client communicates with the application on Amazon AppStream through libraries of the Amazon AppStream SDK.

The lifecycle of a client is as follows:

1. **Create the `XStxClientLibraryHandle` object by calling the `XStxClientLibraryCreate` function to return a `clientLibraryHandle`.** The object interacts with the client side portion of the API through the `clientLibraryHandle`.
2. **Create the `XStxClient` object by calling the `XStxClientCreate` function to return a `XStxClientHandle` client handle.** The object manages the session between the client and the application.
3. **Instantiate and populate the `XStxIClientListener` structure.** This object responds to the callback functions from the application when a session is established between the client and the application. After populating the listener structure, call the `XStxClientSetListener` function with the client handle and a pointer to the `XStxIClientListener` structure.
4. **Initialize a video module to get, decode, and render the video frames.** To initialize this module, the client has to instantiate the following structures:

   - **`XStxIVideoDecoder`** Gets and decodes the video frame.
   - **`XStxIVideoRenderer`** Renders the video frame.
   - **`XStxIRawVideoFrameAllocator`** Gets and recycles the video frame.

5. **Initialize an audio module to get, decode, and render the audio frames.** To initialize this module, the client has to instantiate the following structures:

   - **`XStxIAudioDecoder`** Gets and decodes the audio frame.
   - **`XStxIAudioRenderer`** Renders the audio frame.
   - **`XStxIRawAudioFrameAllocator`** Gets and recycles the audio frame.

6. **Configure a session between the client and application.** Call the `XStxClientSetEntitlementUrl` function with the endpoint of the entitlement service to configure this session.
7. **Call the `XStxClientStart` function to start the session.** Once the session is established, the `XStxClient` object will invoke the `XStxIClientListenerFcnClientReady` function on the `XStxIClientListener` structure. You can then call `xStxClientLibraryRecycle` to recycle the client library handle.
8. **The client sends user inputs to the application**
9. **The client closes the session.**

# Building the Example Client for Windows

The Amazon AppStream SDK contains source code that you can build into a example client application. To learn how to build the source code, see the `\doc` directory.

# Creating Your Client

To create a client, your client must create an `XStxClientLibraryHandle` object. This is the top level object that you will use to interact with the libraries you will use to connect to sessions, render the audio and video, and send inputs to the application.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `main` function of `XStxExampleClient.cpp`.

```
/** instantiate client library handle */
XStxClientLibraryHandle clientLibraryHandle;
XStxResult createResult = XSTX_RESULT_OK;

if ((createResult = XStxClientLibraryCreate(
XSTX_CLIENT_API_VERSION_MAJOR,
XSTX_CLIENT_API_VERSION_MINOR,
&clientLibraryHandle)) != XSTX_RESULT_OK)
{
    const char * name; const char * desc;
    XStxResultGetInfo(createResult, &name, &desc);
    printf("XStxClientLibraryCreate failed with: %s\n", name);
    return 1;
}
```

The function creates a client library handle (`clientLibraryHandle`) that the client will use to create a client object. The example client creates the client object by instantiating an object from a user-defined class.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `WindowListener` class definition of `XStxExampleClient.cpp`.

```
// setup xstx client and start making the connection
if ((result = XStxClientCreate(mClientLibraryHandle, &mClientHandle))
    != XSTX_RESULT_OK)
{
    mRenderWindow->setErrorText("Failed to create client");
    const char * name; const char * desc;
```

```
    XStxResultGetInfo(result, &name, &desc);
    printf("XStxClientCreate failed with: %s\n", name);
    return;
}
```

The object needs a structure to respond to the callback functions from the application. The client creates and populates a `XStxIClientListener` structure to respond to the callback functions.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `readyForConnection` function of `XStxExampleClient.cpp`.

```
...
XStxIClientListener mStxListener;
...
mStxListener.mClientReadyFcn = &clientReady;
mStxListener.mClientReadyCtx = this;
mStxListener.mClientStoppedFcn = &clientStopped;
mStxListener.mClientStoppedCtx = this;
mStxListener.mMessageReceivedFcn = &messageReceived;
mStxListener.mMessageReceivedCtx = this;
```

After populating the structure, call the `XStxClientSetListener` function to configure a listener that responds to the callback functions.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `readyForConnection` function of `XStxExampleClient.cpp`.

```
if ((result = XStxClientSetListener(mClientHandle, &mStxListener))
!= XSTX_RESULT_OK)
{
    mRenderWindow->setErrorText("Failed to set listener");
    const char * name; const char * desc;
    XStxResultGetInfo(result, &name, &desc);
    printf("XStxClientSetListener failed with: %s\n", name);
    return;
}
```

The client uses structures to get, render, and decode video and audio frames. The next step is to create and populate these structures. The example client instantiates a class to create and populate the structures.

To handle a video frame, the client create and populate the following structures defined in `VideoModule.h`:

- **`XStxIVideoDecoder`**. Used to get and decode a video frame.
- **`XStxIVideoRenderer`**. Used to render a video frame.
- **`XStxIRawVideoFrameAllocator`**. Used to get and recycle a video frame.

The following excerpt from the example client in the Amazon AppStream SDK illustrates populating the structures for video. The excerpt is from `VideoModule.cpp`.

```
/**
* Initialize video module
```

```
* @param[in] clientHandle handle to XStx client
* @param[in] rw renderer
*/
bool VideoModule::initialize(XStxClientHandle mClientHandle,
    RenderWindow & rw)
{
    // initialize frame allocator
    mStxFrameAllocator.mInitFcn = &allocatorInit;
    mStxFrameAllocator.mInitCtx = this;
    mStxFrameAllocator.mGetVideoFrameBufferFcn = &allocatorGetFrame;
    mStxFrameAllocator.mGetVideoFrameBufferCtx = this;
    mStxFrameAllocator.mRecycleVideoFrameBufferFcn = &allocatorRecycleFrame;
    mStxFrameAllocator.mRecycleVideoFrameBufferCtx = this;
    mStxFrameAllocator.mSize = sizeof(mStxFrameAllocator);
    if (XStxClientSetVideoFrameAllocator(mClientHandle, &mStxFrameAllocator)
            != XSTX_RESULT_OK)
    {
        printf("Failed to SetVideoFrameAllocator.\n");
        return false;
    }

    // initialize renderer
    mRenderer = &rw;

    mStxRenderer.mRenderVideoFrameFcn = &renderFrame;
    mStxRenderer.mRenderVideoFrameCtx = mRenderer;
    mStxRenderer.mSetMaxResolutionFcn = &rendererMaxResolution;
    mStxRenderer.mSetMaxResolutionCtx = mRenderer;
    mStxRenderer.mSize = sizeof(mStxRenderer);
    if (XStxClientSetVideoRenderer(mClientHandle, &mStxRenderer)
            != XSTX_RESULT_OK)
    {
        printf("Failed to set SetVideoRenderer.\n");
        return false;
    }

    // initialize decoder
    mDecoder = new H264ToYuv();
    if (!mDecoder)
    {
        return false;
    }
    mStxDecoder.mGetCapabilitiesCtx = mDecoder;
    mStxDecoder.mGetCapabilitiesFcn = &videoDecoderGetCapabilities;
    mStxDecoder.mDecodeVideoFrameFcn = &decodeFrame;
    mStxDecoder.mDecodeVideoFrameCtx = mDecoder;
    mStxDecoder.mStartFcn = &videoDecoderStart;
    mStxDecoder.mStartCtx = mDecoder;
    mStxDecoder.mSize = sizeof(mStxDecoder);
    if (XStxClientSetVideoDecoder(mClientHandle, &mStxDecoder)
            != XSTX_RESULT_OK)
    {
        printf("Failed to set SetVideoDecoder.\n");
        return false;
    }
```

To handle an audio frame, the client create and populate the following structures defined in `AudioModule.h`:

- **`XStxIAudioDecoder`**. Used to get and decode an audio frame.
- **`XStxIAudioRenderer`**. Used to render an audio frame.
- **`XStxIRawAudioFrameAllocator`**. Used to get and recycle an audio frame.

The following excerpt from the example client in the Amazon AppStream SDK illustrates populating the structures for audio. The excerpt is from `AudioModule.cpp`.

```
/**
 * Initialize audio module
 * @param[in] clientHandle handle to XStx client
 */
bool AudioModule::initialize(XStxClientHandle clientHandle)
{
    // initialize frame allocator
    mStxFrameAllocator.mInitFcn = &allocatorInit;
    mStxFrameAllocator.mInitCtx = this;
    mStxFrameAllocator.mGetAudioFrameBufferFcn = &allocatorGetFrame;
    mStxFrameAllocator.mGetAudioFrameBufferCtx = this;
    mStxFrameAllocator.mRecycleAudioFrameBufferFcn = &allocatorRecycleFrame;
    mStxFrameAllocator.mRecycleAudioFrameBufferCtx = this;
    mStxFrameAllocator.mSize = sizeof(mStxFrameAllocator);
    if (XStxClientSetAudioFrameAllocator(clientHandle, &mStxFrameAllocator)
            != XSTX_RESULT_OK)
    {
        printf("Failed to SetAudioFrameAllocator\n");
        return false;
    }

    // initialize decoder
    mDecoder = new (std::nothrow) OpusToPcm();
    if (!mDecoder)
    {
        printf("Failed to create Decoder\n");
        return false;
    }
    mStxDecoder.mDecodeAudioFrameFcn = &decoderDecodeFrame;
    mStxDecoder.mDecodeAudioFrameCtx = mDecoder;
    mStxDecoder.mStartFcn = &decoderStart;
    mStxDecoder.mStartCtx = mDecoder;
    mStxDecoder.mSize = sizeof(mStxDecoder);
    if (XStxClientSetAudioDecoder(clientHandle, &mStxDecoder)
            != XSTX_RESULT_OK)
    {
        printf("Failed to SetAudioDecoder\n");
        return false;
    }

    // initialize renderer
    mRenderer = new (std::nothrow) AudioRenderer(mFramePool,
        clientHandle);
    if (!mRenderer)
    {
        return false;
```

```
    }

    // Set XStx callbacks and contexts on for the XStxIAudioRenderer struct
    mStxRenderer.mStartFcn = &start;
    mStxRenderer.mStartCtx = mRenderer;
    mStxRenderer.mSize = sizeof(mStxRenderer);

    // set the XStxIAudioRenderer to be used with the given clientHandle
    if (XStxClientSetAudioRenderer(clientHandle, &mStxRenderer)
            != XSTX_RESULT_OK)
    {
        printf("Failed to SetAudioRenderer\n");
        return false;
    }
    return true;
}
```

The client is now ready to configure and start a session with the application. The client configures a session by calling the `XStxClientSetEntitlementUrl` function with the endpoint of the entitlement service.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `readyForConnection` function in `XStxExampleClient.cpp`.

```
if ((result = XStxClientSetEntitlementUrl(
        mClientHandle, mEntitlementUrl.c_str())) != XSTX_RESULT_OK)
{
    mRenderWindow->setErrorText("Failed to set entitlement URL");
    const char * name; const char * desc;
    XStxResultGetInfo(result, &name, &desc);
    printf("XStxClientSetEntitlementUrl failed with: %s\n", name);
    return;
}
```

The client starts a session by calling a function.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `readyForConnection` function in `XStxExampleClient.cpp`.

```
// non-blocking!
if ((result = XStxClientStart(mClientHandle)) != XSTX_RESULT_OK)
{
    mRenderWindow->setErrorText("Failed to start client");
    const char * name; const char * desc;
    XStxResultGetInfo(result, &name, &desc);
    printf("XStxClientStart failed with: %s\n", name);
    return;
}

// success !
mRenderWindow->setErrorText("Starting STX");
```

# Choosing a Color Subsampling Rate

Amazon AppStream streams the video at the YUV420 color subsampling rate to client applications. If your streaming application, client application and device support YUV444, you can set your client application to display the video at YUV444. The example client application in Amazon AppStream SDK is configured to accept and render a stream in the YUV444 color subsampling rate.

When the client application connects to the streaming application, Amazon AppStream compares the color subsampling options available on the client application with the options advertised on the streaming application. Amazon AppStream then selects the highest color resolution supported by both the client application and streaming application. Amazon AppStream then calls the `XStxIClientListener2FcnSetConfiguration` callback function that the client application supplied and passes the structure with the *`XStxChromaSampling`* setting.

# Sending Your Client Inputs to the Application

The client can send keyboard, mouse, touch, or raw inputs to the application. The client sends an input by filling out the `XStxInputEvent` structure that describes the input and then calling a function that passes the structure to the application.

The following excerpt from the example client in the Amazon AppStream SDK illustrates a keyboard action that populates the structures and then sending that structure to the application. The excerpt is from the `keyChange` function in `DirectXRenderWindow.cpp`.

```
/** Handles WM_KEYDOWN & WM_KEYUP messages. */
void keyChange(WPARAM wParam, LPARAM lParam, bool isKeyDown)
{
    if (!mListener)
    {
        return;
    }
    XStxInputEvent inputEvent;
    inputEvent.mTimestampUs = mud::TimeVal::mono().toMicroSeconds();
    inputEvent.mDeviceId = 0;
    inputEvent.mUserId = 0;
    inputEvent.mType = INPUT_EVENT_TYPE_KEYBOARD;

    inputEvent.mInfo.mKeyboard.mIsKeyDown = isKeyDown;
    inputEvent.mInfo.mKeyboard.mVirtualKey = wParam;
    inputEvent.mInfo.mKeyboard.mScanCode = lParam;

    inputEvent.mSize = sizeof(inputEvent);

    mListener->sendInput(inputEvent);
}
```

The following excerpt from the example client in the Amazon AppStream SDK illustrates a mouse action that populates the structures and then sending that structure to the application. The excerpt is from the `mouseChange` function in `DirectXRenderWindow.cpp`.

```
void mouseChange(WPARAM wParam, LPARAM lParam, bool isDown, bool isLeft)
{
    if (!mListener)
```

```
    {
        return;
    }
    XStxInputEvent inputEvent;
    inputEvent.mTimestampUs = mud::TimeVal::mono().toMicroSeconds();
    inputEvent.mDeviceId = 0;
    inputEvent.mUserId = 0;
    inputEvent.mType = INPUT_EVENT_TYPE_MOUSE;

    inputEvent.mInfo.mMouse.mFlags = MOUSE_MOVE_ABSOLUTE;

    int32_t leftchange = 0;
    int32_t rightChange = 0;
    if (isLeft) {
        leftchange = isDown ? RI_MOUSE_LEFT_BUTTON_DOWN : RI_MOUSE_LEFT_BUT
TON_UP;
    } else {
        rightChange = isDown ? RI_MOUSE_RIGHT_BUTTON_DOWN :
RI_MOUSE_RIGHT_BUTTON_UP;
    }

    inputEvent.mInfo.mMouse.mButtonFlags = leftchange  | rightChange;
    inputEvent.mInfo.mMouse.mButtons = 0;// not needed
    inputEvent.mInfo.mMouse.mButtonData = 0; // not sending wheel data
    rescaleMouseInput(lParam, inputEvent.mInfo.mMouse.mLastX,
        inputEvent.mInfo.mMouse.mLastY);

    inputEvent.mSize = sizeof(inputEvent);
    mListener->sendInput(inputEvent);
}
```

If the client is a different size from the application, the client rescales the mouse position to adjust for the different sizes.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `rescaleMouseInput` function in `DirectXRenderWindow.cpp`.

```
/** Rescales absolute mouse position. */
void rescaleMouseInput(LPARAM lParam, int32_t & outX, int32_t & outY)
{
    // if we send absolute input, we need to rescale it
    RECT desktopRect;
    GetWindowRect(mWindow, &desktopRect);
    float rescaleHeight =  mLastSetHeight /
        (float) (desktopRect.bottom - desktopRect.top - mWindowBorderAdjus
tHeight);
    float rescaleWidth =  mLastSetWidth /
        (float) (desktopRect.right - desktopRect.left - mWindowBorderAd
justWidth);

    outX = GET_X_LPARAM(lParam) * rescaleWidth + 0.5f; // add 0.5 for rounding

    outY = GET_Y_LPARAM(lParam) * rescaleHeight + 0.5f;
}
```

Touch input is sent as a mouse action.

## Terminating Your Client

The client can end the session with the application in the following ways:

- End the session and then confirm the session ended.
- End the session without regard as to when the session ends.

To end the session and then confirm the session ended, first call the `XStxClientStop` function. This is a non-blocking function call that immediately returns a result. If the call was successful, then call the `XStxClientWait` function to wait until the session actually ends. When the `XStxClientWait` call is successful, call the `XStxClientRecycle` function to recycle the client handle.

To end the session without regard as to when the session ends, call the `XStxClientRecycle` function. The session then ends without further interaction from the client.

The following excerpt from the example client in the Amazon AppStream SDK illustrates this step. The excerpt is from the `windowClosed` function in `XStxExampleClient.cpp`.

```
/** Clean up */
void windowClosed()
{
    // the window was closed, do what I need
    if (!mClientHandle)
    {
        return;
    }
    // ensure the audio stopped pulling
    mAudioModule.stop();

    XStxClientRecycle(mClientHandle);
    mClientHandle = NULL;
}
```

# Codec and Open Source Licensing

## What audio and video formats does Amazon AppStream use?

AppStream utilizes the H.264/AVC video format for encoding streamed video, and the open-source Opus audio format for encoding streamed audio.

## How does my client decode video from Amazon AppStream?

You may need to include an H.264/AVC decoder with your client application. In our experience, the built-in decoder included with Windows 7 provides sufficient functionality, but the decoders provided with the iOS and Android platforms do not. The example client we provide for developer education and testing uses FFmpeg, an LGPL2.1-licensed open-source decoder. We have also found that CoreAVC, a proprietary

decoder available from CoreCodec, Inc., is a good option as well. You are responsible for complying with the license terms which apply to the decoder you use in your client application.

## Does use of Amazon AppStream require proprietary licenses?

You may have to obtain a license from MPEG-LA for use of a video decoder with your client application and transmission of video of your hosted application, depending on the nature of your application and the video content transmitted to your end users. You should reach out to the provider of your decoder and/or MPEG-LA for guidance.

Content that you or any End User run on, cause to interface with, or upload to Amazon AppStream is Your Content. You are responsible for determining whether your content or technology used in your Amazon AppStream hosted applications, entitlement service, or clients require any additional licensing.

## Are there any open source considerations?

Amazon AppStream components, including client binary components, utilize certain open source packages, which are noted in the `notice.txt` files provided with the Amazon AppStream SDK. When you distribute your Amazon AppStream client, you should include the attributions for the Amazon AppStream client binary component, which are listed in the applicable `notice.txt` file for the particular client in the Amazon AppStream SDK. In addition, Amazon AppStream sample code provided in the SDK also includes several open source packages, which carry their own licenses; if you use this sample code you need to comply with the applicable license terms.

# Amazon AppStream Client Application License and Notices

**Updated: November 17, 2014**

THIS IS AN AGREEMENT BETWEEN YOU AND AWS MOBILE LLC (WITH ITS AFFILIATES, **"AWS MOBILE"** OR **"WE"**) THAT GOVERNS YOUR USE OF THE AMAZON APPSTREAM CLIENT SOFTWARE (TOGETHER WITH ANY UPDATES AND ENHANCEMENTS TO IT, AND ACCOMPANYING DOCUMENTATION, THE **"APPLICATION"**) THAT WE MAKE AVAILABLE TO YOU ON YOUR CONNECTED DEVICE (EACH, A **"DEVICE"**). IF YOU INSTALL OR USE THE APPLICATION, YOU WILL BE BOUND BY THIS AGREEMENT.

## 1. Use of the Application.

We hereby grant you a personal, limited, non-exclusive, non-transferable, non-sublicenseable license to install and use the Application on your Device. The Application enables you to connect to content and services you or third parties make available via the Amazon Appstream service (**“Streamed Content”**). Streamed Content may be made available under separate terms and conditions, including separate fees and charges. Because we may not have tested or screened the Streamed Content, your use of any Streamed Content is at your sole risk. You should carefully review the terms and privacy statements for any Streamed Content you use.

## 2. Limitations.

You may not, and you will not encourage, assist or authorize any other person to, (a) incorporate any portion of the Application into your own programs or compile any portion of it in combination with your

own programs; (b) sell, rent, lease, lend, loan, distribute, act as a service bureau, publicly communicate, transform, or sub-license the Application or otherwise assign any rights to the Application in whole or in part; (c) modify, alter, tamper with, repair, or otherwise create derivative works of the Application, or (d) reverse engineer, disassemble, or decompile the Application or apply any other process or procedure to derive the source code of any software included in the Application. All rights granted to you are conditioned on your continued compliance with this Agreement, and will immediately and automatically terminate if you do not comply.

# 3. Reservation of Rights.

You may not use the Application for any illegal purpose. The Application is the intellectual property of Amazon.com, Inc. or its affiliates, and its licensors. The structure, organization, and code of the Application are valuable trade secrets and confidential information of Amazon.com, Inc. or its affiliates. The Application is protected by law, including without limitation copyright laws and international treaty provisions. Except for the rights explicitly granted to you in this Agreement, all right, title and interest in the Application are reserved and retained by us and our licensors.

# 4. Updates.

In order to keep the Application up-to-date, we may offer automatic or manual updates at any time and without notice to you. If we elect to provide maintenance or support of any kind, we may terminate that maintenance or support at any time without notice to you.

# 5. Termination.

You may terminate this Agreement at any time by uninstalling or destroying all copies of the Application that are in your possession or control. In the case of termination, you must cease all use and destroy all copies of the Application. We may also terminate your right to use the Application at any time and if we do so, we may modify the Application to make it inoperable. Our failure to insist upon or enforce your strict compliance with this Agreement will not constitute a waiver of any of our rights.

# 6. Disclaimer of Warranties and Limitation of Liability.

a. YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT INSTALLATION AND USE OF, AND ANY OTHER ACCESS TO, THE APPLICATION AND ANY STREAMED CONTENT IS AT YOUR SOLE RISK. THE APPLICATION IS DELIVERED TO YOU **"AS IS"** WITH ALL FAULTS AND WITHOUT WARRANTY OF ANY KIND, AND AWS MOBILE, ITS LICENSORS AND DISTRIBUTORS, WIRELESS CARRIERS OVER WHOSE NETWORK THE APPLICATION IS DISTRIBUTED, AND EACH OF THEIR RESPECTIVE AFFILIATES AND SUPPLIERS (COLLECTIVELY, THE **"RELEASED PARTIES"**) DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, QUIET ENJOYMENT, AND NON-INFRINGEMENT. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY A RELEASED PARTY OR AN AUTHORIZED REPRESENTATIVE OF A RELEASED PARTY WILL CREATE A WARRANTY. THE LAWS OF CERTAIN JURISDICTIONS DO NOT ALLOW THE DISCLAIMER OF IMPLIED WARRANTIES. IF THESE LAWS APPLY TO YOU, SOME OR ALL OF THE ABOVE DISCLAIMERS, EXCLUSIONS, OR LIMITATIONS MAY NOT APPLY TO YOU, AND YOU MAY HAVE ADDITIONAL RIGHTS.

b. TO THE EXTENT NOT PROHIBITED BY LAW, NO RELEASED PARTY WILL BE LIABLE TO YOU FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, BREACH OF CONTRACT, NEGLIGENCE, STRICT LIABILITY, OR ANY OTHER LEGAL THEORY RELATED TO THE APPLICATION OR THE STREAMED CONTENT, INCLUDING WITHOUT LIMITATION ANY DAMAGES ARISING OUT OF LOSS OF PROFITS, REVENUE, DATA,

OR USE OF THE APPLICATION, EVEN IF A RELEASED PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, ANY RELEASED PARTY'S AGGREGATE LIABILITY UNDER THE AGREEMENT WILL BE LIMITED TO $50.00. THE LAWS OF CERTAIN JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. IF THESE LAWS APPLY TO YOU, SOME OR ALL OF THE ABOVE EXCLUSIONS OR LIMITATIONS MAY NOT APPLY TO YOU, AND YOU MAY HAVE ADDITIONAL RIGHTS.

# 7. Indemnification.

You are liable for and will defend, indemnify, and hold harmless the Released Parties and their officers, directors, agents, and employees, from and against any liability, loss, damage, cost, or expense (including reasonable attorneys' fees) arising out of your use of the Application or Streamed Content, violation of this Agreement, violation of applicable law, or violation of any right of any person or entity, including without limitation intellectual property rights.

# 8. Export Regulations.

You will comply with all export and re-export restrictions and regulations of the United States Department of Commerce and other United States and foreign agencies and authorities that may apply to the Application, and not to transfer, or encourage, assist, or authorize the transfer of the Application to a prohibited country or otherwise in violation of any applicable restrictions or regulations.

# 9. U.S. Government End Users.

The Application is a **"Commercial Item"** as that term is defined in 48 C.F.R. § 2.101, consisting of **"Commercial Computer Software"** and **"Commercial Computer Software Documentation,"** as the terms are used in 48 C.F.R. § 12.212 or 48 C.F.R. § 227.7202, as applicable. Consistent with these provisions, the Application is licensed to U.S. Government end users: (a) only as a Commercial Item; and (b) with only those rights as are granted to all other end users pursuant to these Terms of Use. Unpublished-rights are reserved under U.S. copyright laws.

# 10. Amendment.

We may amend this Agreement at our sole discretion by posting the revised terms on the Amazon Web Services website (located at: http://aws.amazon.com) or within the Application. Your continued use of the Application after any amendment's effective date evidences your agreement to be bound by it.

# 11. Conflicts.

The terms of this Agreement govern the Application and any updates or upgrades to the Application that we may provide that replace or supplement the original Application, unless the update or upgrade is accompanied by a separate license, in which case the terms of that license will govern.

# 12. Contact Information.

For communications concerning this Agreement, please write to AWS Mobile LLC, Attn: Legal Department, 410 Terry Avenue North, Seattle, WA 98109-5210.

Notices for AppStream Demonstration Windows Client
---
Notices for component(s):
JRtpLib

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation files
(the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

---

Notices for component(s):
reed_solomon.c and reed_solomon.h

// Reed-Solomon encoding/erasure decoding

// Implementation of the algorithms described in
// Efficient erasure decoding of Reed-Solomon codes
// http://arxiv.org/abs/0901.1886v1

// (c) 2009 Frederic Didier.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
// disclaimer in the documentation and/or other materials
// provided with the distribution.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND
// ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
// THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
// PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS
// BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
// OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
// OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
// TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
// OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
// OF SUCH DAMAGE.

---
Notices for component(s):
Opus

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,
Jean-Marc Valin, Timothy B. Terriberry,
CSIRO, Gregory Maxwell, Mark Borgerding,
Erik de Castro Lopo

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Skype Limited:
https://datatracker.ietf.org/ipr/1602/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

---
Notices for component(s):
FFmpeg

This software uses code of FFmpeg licensed under the LGPLv2.1 and its source can be downloaded from www.ffmpeg.o

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999



[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU Ger

This license, the Lesser General Public License, applies to some specially designated software packages--typically librari

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surren

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights tha

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives y

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is m

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a compar

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GN

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally sp

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordin

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a la

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a prog

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference bet
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyrigh

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application p

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work ba

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete sourc

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, pr

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in ex

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and co

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any ch
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Lib

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the inten

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Libra

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of th

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License appl

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable for

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access t

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being com

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (be

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the wor

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small i

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Sec

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to p

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whate
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified i
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally a

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other lib

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other libra
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explainin

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission t

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a licens

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to pa

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interface

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from tim

Each version is given a distinguishing version number. If the Library specifies a version number of this License which appl

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible w

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO TH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT H
END OF TERMS AND CONDITIONS

---
Notices for component(s):
Port Audio

Copyright (c) 1999-2011 Ross Bencina and Phil Burk

PortAudio Portable Real-Time Audio Library
Copyright (c) 1999-2011 Ross Bencina and Phil Burk

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentati

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Softw

THE SOFTWARE IS PROVIDED ""AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHE

---
Notices for component(s):
msinttypes

Copyright (c) 1998, Regents of the University of California
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following c

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclai

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS

---
Notices for component(s):
ProtoBuf

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

---
Notices for component(s):
OpenSSL

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSD-style
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License
---------

```
/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

Original SSLeay License
-----------------------
```

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

---
Notices for component(s):
libcURL

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2013, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

Notices for AppStream Demonstration Client for Android
---
Notices for component(s):
JRtpLib

© 2006-2013 Jori Liesenborgs
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation files
(the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge,
publish, distribute, sublicense, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

---
Notices for component(s):
reed_solomon.c and reed_solomon.h

```
// Reed-Solomon encoding/erasure decoding

// Implementation of the algorithms described in
// Efficient erasure decoding of Reed-Solomon codes
// http://arxiv.org/abs/0901.1886v1

// (c) 2009 Frederic Didier.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
// disclaimer in the documentation and/or other materials
// provided with the distribution.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND
// ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
// THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
// PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS
// BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
// OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
// OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
// TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
// OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
// OF SUCH DAMAGE.



-------------------------
Notices for component(s):
Opus

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,
                    Jean-Marc Valin, Timothy B. Terriberry,
                    CSIRO, Gregory Maxwell, Mark Borgerding,
                    Erik de Castro Lopo

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the
names of specific contributors, may be used to endorse or promote
products derived from this software without specific prior written
permission.
```

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are
specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Skype Limited:
https://datatracker.ietf.org/ipr/1602/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

---
Notices for component(s):
FFmpeg

This software uses code of FFmpeg licensed under the LGPLv2.1 and its source can be downloaded from www.ffmpeg.or

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU Ger

This license, the Lesser General Public License, applies to some specially designated software packages--typically librari

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surren

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights tha

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives y

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is r

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a compa

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the G

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally s

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordi

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a la

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a pro

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference bet
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyrigh

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application p

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work ba

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete sourc

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, pr

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in ex

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and co

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any ch
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Lib

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the inten

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Libra

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of th

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License appl

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable fo

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access t

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being cor

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (be

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the wor

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small i

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Sec

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to p

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whate
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified i
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally a

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other lib

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other libra
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explainin

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission t

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a licens

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to pat

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interface

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from tim

Each version is given a distinguishing version number. If the Library specifies a version number of this License which appl

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible w

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO TH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT H

END OF TERMS AND CONDITIONS

---
Notices for component(s):
ProtoBuf



Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

---
Notices for component(s):
STLPort



This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk.
Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are reta

The Licensee may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrict

The Licensee may distribute original or modified STLport sources, provided that:

- The conditions indicated in the above permission notice are met;
- The following copyright notices are retained when present, and conditions provided in accompanying permission notice

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

---
Notices for component(s):
OpenSSL

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License
---

```
/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
```

```
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/
```

Original SSLeay License

---

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
```

```
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

---

Notices for component(s):
libcURL

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2013, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Notices for Appstream Demonstration Client for iOS

---

Notices for component(s):
JRtpLib

© 2006-2013 Jori Liesenborgs
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,

subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

---

Notices for component(s):
reed_solomon.c and reed_solomon.h

// Reed-Solomon encoding/erasure decoding

// Implementation of the algorithms described in
// Efficient erasure decoding of Reed-Solomon codes
// http://arxiv.org/abs/0901.1886v1

// (c) 2009 Frederic Didier.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
// disclaimer in the documentation and/or other materials
// provided with the distribution.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND
// ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
// THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
// PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS
// BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
// OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
// OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
// TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
// OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
// OF SUCH DAMAGE.

---

Notices for component(s):
Opus

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,

Jean-Marc Valin, Timothy B. Terriberry,
CSIRO, Gregory Maxwell, Mark Borgerding,
Erik de Castro Lopo

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Skype Limited:
https://datatracker.ietf.org/ipr/1602/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

---

Notices for component(s):
FFmpeg

This software uses code of FFmpeg licensed under the LGPLv2.1 and its source can be downloaded from www.ffmpeg.or

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU Ger

This license, the Lesser General Public License, applies to some specially designated software packages--typically librari

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surren

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights tha

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives y

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is m

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a compan

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GN

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally sp

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordin

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a la

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a prog

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference bet
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyrigh

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application p

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work ba

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete sourc

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, pr

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in ex

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and co

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any ch

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Lib

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the inten

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Libra

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of th

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License appl

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable for

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access t

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being com

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (be

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the wor

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small i

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Sec

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to p

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whate
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified i
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally a

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other lib

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other libra
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explainin

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission t

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a licens

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to pat

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interface

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from tim

Each version is given a distinguishing version number. If the Library specifies a version number of this License which appl

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible w

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO TH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT H
END OF TERMS AND CONDITIONS
---
Notices for component(s):
ProtoBuf

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

---
Notices for component(s):
OpenSSL

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License
---

```
/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
```

```
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
```

```
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/


-------------------------
Notices for component(s):
libcURL

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2013, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

Notices for Appstream Demonstration Client for Mac OS
-------------------------
Notices for component:
JRtpLib

© 2006-2013 Jori Liesenborgs
Permission is hereby granted, free of charge, to any person
 obtaining a copy of this software and associated documentation files
 (the "Software"), to deal in the Software without restriction,
 including without limitation the rights to use, copy, modify, merge,
 publish, distribute, sublicense, and/or sell copies of the Software,
 and to permit persons to whom the Software is furnished to do so,
 subject to the following conditions:

 The above copyright notice and this permission notice shall be
 included in all copies or substantial portions of the Software.
```

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

---

Notices for component:
reed_solomon.c and reed_solomon.h

// Reed-Solomon encoding/erasure decoding

// Implementation of the algorithms described in
// Efficient erasure decoding of Reed-Solomon codes
// http://arxiv.org/abs/0901.1886v1

// (c) 2009 Frederic Didier.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
// disclaimer in the documentation and/or other materials
// provided with the distribution.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND
// ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
// THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
// PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS
// BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
// OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
// OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
// TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
// OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
// OF SUCH DAMAGE.

---

Notices for component:
Opus

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,
Jean-Marc Valin, Timothy B. Terriberry,
CSIRO, Gregory Maxwell, Mark Borgerding,
Erik de Castro Lopo

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the
names of specific contributors, may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are
specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Skype Limited:
https://datatracker.ietf.org/ipr/1602/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

---
Notices for component:
FFmpeg

This software uses code of FFmpeg licensed under the LGPLv2.1 and its source can be downloaded from www.ffmpeg.or
GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence

the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU Ger

This license, the Lesser General Public License, applies to some specially designated software packages--typically librari

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surren

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights tha

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives y

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is m

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a compar

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GN

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally sp

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordin

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a la

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a pro

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference bet
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyrigh

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application p

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work ba

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete sourc

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, pr

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in ex

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and co

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any ch
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Lib

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the inten

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Libra

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of th

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License appl

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable for

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access t

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being com

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (be

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the wor

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small i

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Sec

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to p

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whate
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified i
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally a

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other lib

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other libra
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explainin

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission t

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a licens

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to pat

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interface

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from tim

Each version is given a distinguishing version number. If the Library specifies a version number of this License which appl

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible w

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO TH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT H
END OF TERMS AND CONDITIONS

---
Notices for component:
ProtoBuf

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

---
Notices for component:
OpenSSL

```
  LICENSE ISSUES
  ==============

  The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
  the OpenSSL License and the original SSLeay license apply to the toolkit.
  See below for the actual license texts. Actually both licenses are BSD-style
  Open Source licenses. In case of any license issues related to OpenSSL
  please contact openssl-core@openssl.org.

  OpenSSL License
  ---------------

 /* ====================================================================
  * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
  *
  * Redistribution and use in source and binary forms, with or without
  * modification, are permitted provided that the following conditions
  * are met:
  *
  * 1. Redistributions of source code must retain the above copyright
  *    notice, this list of conditions and the following disclaimer.
  *
  * 2. Redistributions in binary form must reproduce the above copyright
  *    notice, this list of conditions and the following disclaimer in
  *    the documentation and/or other materials provided with the
  *    distribution.
  *
  * 3. All advertising materials mentioning features or use of this
  *    software must display the following acknowledgment:
  *    "This product includes software developed by the OpenSSL Project
  *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
  *
  * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
  *    endorse or promote products derived from this software without
  *    prior written permission. For written permission, please contact
  *    openssl-core@openssl.org.
  *
  * 5. Products derived from this software may not be called "OpenSSL"
  *    nor may "OpenSSL" appear in their names without prior written
  *    permission of the OpenSSL Project.
  *
  * 6. Redistributions of any form whatsoever must retain the following
  *    acknowledgment:
  *    "This product includes software developed by the OpenSSL Project
  *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
  *
  * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
  * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
  * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
  * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
  * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
  * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
  * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
  * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
  * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
```

```
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
```

```
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */


-------------------------
Notices for component:
libcURL

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2013, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.

Notices for AppStream Demonstration Client for Chrome
-------------------------
Notices for component(s):
JRtpLib

© 2006-2013 Jori Liesenborgs
Permission is hereby granted, free of charge, to any person
 obtaining a copy of this software and associated documentation files
 (the "Software"), to deal in the Software without restriction,
 including without limitation the rights to use, copy, modify, merge,
 publish, distribute, sublicense, and/or sell copies of the Software,
 and to permit persons to whom the Software is furnished to do so,
 subject to the following conditions:

 The above copyright notice and this permission notice shall be
 included in all copies or substantial portions of the Software.

 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
 KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
 WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
```

NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

Notices for component(s):
reed_solomon.c and reed_solomon.h

// Reed-Solomon encoding/erasure decoding

// Implementation of the algorithms described in
// Efficient erasure decoding of Reed-Solomon codes
// http://arxiv.org/abs/0901.1886v1

// (c) 2009 Frederic Didier.

// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
//
// 1. Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following
// disclaimer in the documentation and/or other materials
// provided with the distribution.
//
// THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND
// ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
// THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
// PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS
// BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
// OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
// PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
// OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
// TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
// OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
// OF SUCH DAMAGE.

---

Notices for component(s):
Opus

Copyright 2001-2011 Xiph.Org, Skype Limited, Octasic,
Jean-Marc Valin, Timothy B. Terriberry,
CSIRO, Gregory Maxwell, Mark Borgerding,
Erik de Castro Lopo

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

- Neither the name of Internet Society, IETF or IETF Trust, nor the
names of specific contributors, may be used to endorse or promote
products derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Opus is subject to the royalty-free patent licenses which are
specified at:

Xiph.Org Foundation:
https://datatracker.ietf.org/ipr/1524/

Skype Limited:
https://datatracker.ietf.org/ipr/1602/

Broadcom Corporation:
https://datatracker.ietf.org/ipr/1526/

---
Notices for component(s):
FFmpeg

This software uses code of FFmpeg licensed under the LGPLv2.1 and its source can be downloaded from www.ffmpeg.or

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU Ger

This license, the Lesser General Public License, applies to some specially designated software packages--typically librari

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surren

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights tha

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives y

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is m

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a compan

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GN

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally sp

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordin

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a la

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a prog

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference bet
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyrigh

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application p

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work ba

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete sourc

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, pr

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in ex

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and co

a) The modified work must itself be a software library.
b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any ch
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Lib

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the inten

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Libra

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of th

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License appl

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable fo

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access t

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being co

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (be

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the wor

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small i

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Sec

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to p

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whate
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified i
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally a

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other lib

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other libra
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explainin

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission t

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a licens

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to pat

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interface

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from tim

Each version is given a distinguishing version number. If the Library specifies a version number of this License which appl

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible v

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO TH

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT H
END OF TERMS AND CONDITIONS

---
Notices for component(s):
ProtoBuf

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

---
Notices for component(s):
STLPort

Copyright 1999, 2000 Boris Fomitchev

This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk.

Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are reta

The Licensee may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrict

The Licensee may distribute original or modified STLport sources, provided that:

The conditions indicated in the above permission notice are met;
The following copyright notices are retained when present, and conditions provided in accompanying permission notice

Copyright 1994 Hewlett-Packard Company

Copyright 1996,97 Silicon Graphics Computer Systems, Inc.

Copyright 1997 Moscow Center for SPARC Technology.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby grante

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby grante

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby grante

---
Notices for component(s):
Boost
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.

---
Notices for component(s):
OpenSSL

LICENSE ISSUES

==========

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSD-style
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License
---

```
/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
```

```
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
```

```
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

---

Notices for component(s):
libcURL

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2013, Daniel Stenberg, <daniel@haxx.se>.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

# Advanced Topics

**Topics**


- Tools for Amazon AppStream (p. 166)
- Building an Amazon AppStream Streaming Application (p. 167)
- Build an Application Installer (p. 192)
- Building an Entitlement Service (p. 193)
- Amazon AppStream Opaque Data Specification (p. 205)
- Using Environment Variables (p. 205)


## Tools for Amazon AppStream

Amazon AppStream provides several tools to simplify the process of developing application solutions.

### Amazon AppStream SDK

The Amazon AppStream SDK simplifies the process of adding streaming to your application and makes it easier to build clients for Windows devices. It provides C header files and libraries with the functionality needed to stream your application from Amazon AppStream; as well as receive the streamed content in a client. The Amazon AppStream SDK; includes the source code for a sample application and a client as well as a pre-compiled client file that you can use to connect to an application. You can use these sample implementations to test streaming an application from Amazon AppStream without writing any code. You can obtain this SDK from Downloads (p. 11).

### Amazon AppStream SDK for Java

The Amazon AppStream SDK for Java includes functions you can call to interact with the Amazon AppStream service. These functions wrap the Amazon AppStream REST API (p. 216) and handle details like signing the requests sent to the Amazon AppStream service. The most common use of the Amazon AppStream SDK for Java is to write an entitlement service that authorizes user access to your applications. Your entitlement service calls into Amazon AppStream to create new client sessions. You can also use the Amazon AppStream SDK for Java functions to automate Amazon AppStream. For example, you could write an application to bulk-add applications.

## Amazon AppStream Console

The Amazon AppStream console is a graphical interface that you can use to add and manage applications. With it, you fill out web forms to specify the details of streaming applications and view the details of existing applications. The console is available online at https://console.aws.amazon.com/appstream/.

# Building an Amazon AppStream Streaming Application

An application to stream is the heart of your product. In order to be streamed, your application needs to make initialization calls to let Amazon AppStream know that it's ready to accept client sessions, and to have the proper interfaces implemented that Amazon AppStream can call into to connect to client sessions and stream content. You do this using the header and library files provided in the Amazon AppStream SDK. The following sections describe the modifications necessary for streaming and how to add this functionality to an application.

Throughout the discussion, we'll reference code excerpts from a sample application provided in the `<SDK_dir>\examples_src` directory of the Amazon AppStream SDK. You can download the SDK from the links in Downloads (p. 11).

**Topics**


- Design Considerations for Your Streaming Application (p. 167)
- Add Streaming to Your Application (p. 168)
- Testing Your Streaming Application (p. 181)
- Security considerations (p. 189)


## Design Considerations for Your Streaming Application

Streaming your application from the cloud offers several advantages over running it natively on consumer devices: you can run a complex application on simple devices, support new consumer devices without updating your application, seamlessly provide new versions of your application to clients, and improve the security of your code.

In exchange for these advantages, however, you have some new requirements to take into consideration when building your application:

- **Continuous network connection**—A application requires a continuous network connection. What user experience will you offer your customers when the network is unavailable? You might show them an error message, or—in the case of a hybrid application—provide them access to the portion of your application that can function offline, running entirely on the consumer device.
- **Managing latency**—Streaming your application from Amazon AppStream adds sources of latency, a small amount of latency from Amazon AppStream overhead as well as variable latency from network conditions. While in many cases, the added latency will be imperceptible to your users, your application needs to be able to tolerate some latency and to handle latency spikes gracefully.
- **Persistent storage**—When you stream your application from Amazon AppStream, it runs on Amazon AppStream hosts in the cloud. When a client session ends, the Amazon AppStream host resources are recovered and any data stored locally on the Amazon AppStream host is lost. If your application needs to persist state between user sessions, you'll need to record that data in a persistent data store

(such as Amazon S3, Amazon RDS, or DynamoDB) before the client session ends and load the data from the persistent store when the next client session begins.

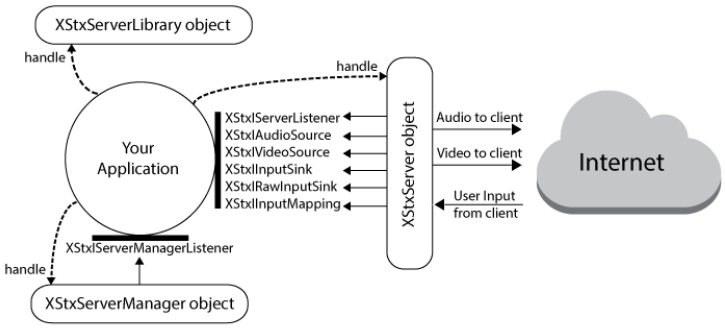
- **Redirecting video and audio output and user input**—In order to stream your application across Amazon AppStream, you'll need to redirect the output to application libraries provided in the Amazon AppStream SDK, and implement interfaces that listen to events from those libraries.
- **Hybrid applications**—Will you stream your entire application from Amazon AppStream or perform some processing on the consumer device? Hybrid applications can offer solutions for both loss of network as well as handling latency spikes. One way to use Amazon AppStream is to run basic application functionality on the consumer device and enhance the experience when a network is available, such as a game with basic character animations rendered on the device with enhanced graphics and detailed backgrounds available through streaming.

> **Note**
> Applications with zero tolerance for latency are not recommended for streaming, such as first-person shooter games or player vs. player fighting games. For these types of applications, consider building a hybrid experience where the latency intolerant aspects of the experience are run locally on the consumer device, while other aspects of the experience requiring cloud resources are streamed.

# Add Streaming to Your Application

The Amazon AppStream SDK provides server libraries that you build into your application and call in order to receive client sessions from Amazon AppStream and to stream input and output between your application and clients.

The following image illustrates how your application interacts with the libraries of the Amazon AppStream SDK. Your application has a handle to each of the objects that it can use to call the functions of that object. It also implements callback functions and registers them with the objects in order to receive client and session events.




The following object classes are provided by the server libraries in the Amazon AppStream SDK. Documentation for these classes are available in the Amazon AppStream SDK.

| Class | Description |
|---|---|
| `XStxServerLibrary` | The top level object of the server libraries. Your application uses this object to ensure that it is calling into the version of the libraries it was compiled against and to create the `XStxServerManager` object. |

| Class | Description |
|---|---|
| `XStxServerManager` | This object manages client sessions and sends events to your application when Amazon AppStream assigns your application a client session or terminates a client session. When it fires the `XStxIServerManagerListenerFcnServerInitialize` event, it returns a handle to the `XStxServer` object your application uses to stream data to the client. |
| `XStxServer` | Your application uses this object to stream content to clients and to receive user input from clients. |

Your application implements callback functions that adhere to the following interfaces in order to receive session and client events. Documentation for these interfaces is available in the Amazon AppStream SDK. The callback functions are invoked from different threads, so your application needs to be cautious when modifying data in the callback functions and to use thread synchronization mechanisms.

| Interface | Description |
|---|---|
| `XStxIAudioSource` | Called by the `XStxServer` object to get audio frames from your application. If a callback function implementing this interface is not set with the `XStxServerSetAudioSource` function of the `XStxServer` object, the `XStxServer` object captures system audio and transmits that to clients instead. |
| `XStxIInputMapping` | Maps user input from a format returned by the client to a format the application can use. |
| `XStxIInputSink` | Receives user data from the `XStxServer` object. This can be keyboard, mouse, or touch data. |
| `XStxIRawInputSink` | Receives raw user data from the `XStxServer` object. This is sent as a stream of bytes. |
| `XStxIServerListener` | receives event messages from the `XStxServer` object when it establishes or loses connection to a client. |
| `XStxIServerManagerListener` | receives event messages from the `XStxServerManager` object when Amazon AppStream creates or terminates a client session. |
| `XStxIVideoSource` | Called by the `XStxServer` object to configure video streaming from the application. If the video mode is set to `XSTX_VIDEO_MODE_PULL`, the `XStxServer` object also uses this interface to get video frames from the application. |

## Lifecycle of a Streaming Application

Your application communicates with Amazon AppStream and clients through the libraries of the Amazon AppStream SDK.

The lifecycle of an application is as follows:

1. **Create the `XStxServerLibrary` object.** This ensures that the loaded libraries match the version your application was compiled with. It also gives you a handle to an `XStxServerLibrary` object, which is required for the next step.

2. **Create the `XStxServerManager` object.** This object communicates with Amazon AppStream to manage the assignment of client sessions to your application.
3. **Register an `XStxIServerManagerListener` event sink.** Pass the location of a callback function to receive events from the `XStxServerManager` object.
4. **Start the `XStxServerManager` object.** Your application calls `XStxServerManagerStart` to cause the `XStxServerManager` object to start accepting client sessions from Amazon AppStream.
5. **Wait for the `XStxServerManager` object to end the session.** During this phase, your application receives events to its `XStxIServerManagerListener` callback function when the session manager accepts a new client session, prompts your application to persist state before a session ends, or terminates a client session.

   If the event is `XStxIServerManagerListenerFcnServerInitialize`, a new client session has begun, and your application is passed a handle to the `XStxServer` object it will use to stream content to the client. In this case, perform the following steps:

   a. **Initialize the `XStxServer` object.**
   b. **Register event sinks for `XStxIAudioSource`, `XStxIInputMapping`, `XStxIInputSink`, `XStxIRawInputSink`, `XStxIServerListener`, and `XStxIVideoSource` events.** These callback functions are how your application is notified about streaming events.
   c. **Call methods of the `XStxServer` object to send audio and video frames.** Repeat this step for the duration of the streaming session.
   d. **Terminate the `XStxServer` object.** When the client is disconnected or the application ends the client session, your application should call `XStxServerTerminate` to release resources allocated to the `XStxServer` object.

6. **Recycle the `XStxServerLibrary` and `XStxServerManager` objects.** After the client session ends, release the resources allocated to the `XStxServerLibrary` and `XStxServerManager` objects.

## Sample Streaming Application

The file `main.cpp` is the source code for a sample implementation of a streaming application that can be streamed from Amazon AppStream. This sample is in the *<SDK_dir>*`\example_src\server\windows\SimpleDirectXServer` directory of the Amazon AppStream SDK.

The sample application uses DirectX for video rendering and XAudio2 for audio rendering. Walking through the sample code is useful in understanding how to construct an application.

## Initializing a Streaming Application

Before your application can start streaming content to a client, it must create the objects that it will use to connect to client sessions and stream content as well as register callback functions to receive event notifications about sessions and clients. The following explanation covers steps 1–5 of Lifecycle of a Streaming Application (p. 169).

First, your application should create an `XStxServerLibrary` object. This is the top level object that you use to interact with the libraries you'll use to connect to sessions and stream content to clients.

The following excerpt from the sample streaming application illustrates this step. The excerpt is from the `initializeAppStream` function of the file `main.cpp`. The file is in the *<SDK_dir>*`\example_src\server\windows\SimpleDirectXServer` directory.

```
int runAsAppStreamGame(int argc, const char* argv[])
{
    // Main AppStream initialization
    XStxResult result;

    ...

    // Create our Server Library handle
    if ((result = XStxServerLibraryCreate(
        XSTX_SERVER_API_VERSION_MAJOR,
        XSTX_SERVER_API_VERSION_MINOR,
        &g_serverLibraryHandle)) != XSTX_RESULT_OK)
        return result;

 ...
}
```

Next, your application creates an `XStxServerManager` object. To do so, you'll pass in the handle to the `XStxServerLibrary` object you created previously. Your application uses the `XStxServerManager` object to receive session assignments from Amazon AppStream.

The following excerpt from the sample streaming application illustrates this step. The excerpt is from the `initializeAppStream` function of the file `main.cpp`. The file is in the *<SDK_dir>*`\example_src\server\windows\SimpleDirectXServer` directory.

```
static XStxResult initializeAppStream()
{
    // Main AppStream initialization
    XStxResult result;

    ...

    // Create our Server Manager
    if ((result = XStxServerLibraryCreateXStxServerManager(
        g_serverLibraryHandle, &g_serverManagerHandle)) != XSTX_RESULT_OK)
        return result;

    ...

}
```

After the `XStxServerManager` object is created, your application calls `XStxServerManagerSetListener` to register an event sink.

The following excerpt from the sample streaming application illustrates this step. The excerpt is from the `initializeAppStream` function of the file `main.cpp`. The file is in the *<SDK_dir>*`\example_src\server\windows\SimpleDirectXServer` directory.

The code that implements the `XStxIServerManagerListener` interface and handles the events sent by the `XStxServerManager` object is located in the file `ServerManagerListener.cpp`. The file is in the *<SDK_dir>*`\example_src\server\common` directory.

```
int runAsAppStreamGame(int argc, const char* argv[])
{
    ...
```

```
    // Set up callbacks so we know when a server is initialized or terminated
    ZeroMemory(&g_serverManagerListener, sizeof(XStxIServerManagerListener));
    g_serverManagerListener.mSize = sizeof(XStxIServerManagerListener);
    g_serverManagerListener.mServerInitializeFcn = serverInitialize;
    g_serverManagerListener.mServerSaveStateFcn = serverSaveState;
    g_serverManagerListener.mServerTerminateFcn = serverTerminate;

    // Supply our callbacks to AppStream
    if ((result = XStxServerManagerSetListener(
        g_serverManagerHandle,
        &g_serverManagerListener)) != XSTX_RESULT_OK)
        return result;

 ...

}
```

After you've registered the event sink, your application is ready to start the server manager to notify Amazon AppStream that it's ready to receive client sessions. To do so, call the `XStxServerManagerStart` function.

The following excerpt from the sample streaming application illustrates this step. The excerpt is from the `initializeAppStream` function of the file `main.cpp`. The file is in the *<SDK_dir>*`\example_src\server\windows\SimpleDirectXServer` directory.

```
static XStxResult initializeAppStream()
{
    ...

    // Tell the server to start up
    if ((result = XStxServerManagerStart(
        g_serverManagerHandle)) != XSTX_RESULT_OK)
        return result;

    ...

}
```

At this point, your application's `main` function should wait until the `XStxServerManager` terminates the session. This is done by calling `XStxServerManagerWait`.

The following excerpt from the sample streaming application illustrates this step. The excerpt is from the `runAsAppStreamGame` function of the file `XStxExampleServer.cpp`. The file is in the *<SDK_dir>*`\example_src\server\common` directory.

```
int runAsAppStreamGame(int argc, const char* argv[])
{
    ...

    /** Wait for the session manager to exit */

    XStxResult result = XStxServerManagerWait(serverManagerHandle);

    ...

}
```

While your application is waiting for the session to end, your `XStxIServerManagerListener` event sink is receiving session events. When the `XStxIServerManagerListenerFcnServerInitialize` event fires, it indicates that Amazon AppStream has assigned your application a client. For more information, see Initializing a Client Session (p. 173)

## Initializing a Client Session

When the `XStxIServerManagerListenerFcnServerInitialize` event fires it returns a handle to an `XStxServer` object. This is the object that your application uses to communicate with the client: streaming audio and video output and receiving user input and client messages. Your implementation of a `XStxIServerManagerListenerFcnServerInitialize` callback function should initialize the `XStxServer` object, register event sinks for the `IServerListener` events, and set the audio source by calling the `XStxServerSetVideoSource` function. Setting the audio source by calling the `XStxServerSetAudioSource` function is optional; for more information, see Streaming Audio to a Client (p. 177).

This excerpt from `ServerManagerListener.cpp` shows how to initialize the `XStxServer` object. The file is in the *<SDK_dir>*`\example_src\server\common` directory.

```
XStxResult ServerManagerListenerImp::XStxIServerManagerListenerServerInitialize(

    XStxServerHandle server,
    uint32_t timeout,
    const char* applicationContext)
{

    if (mServerToInfoMap.find(server) != mServerToInfoMap.end())
    {
        return XSTX_RESULT_ALREADY_CREATED;
    }

    XStxResult result = XSTX_RESULT_OK;

    /** Create an object to hold on to server specific info */

    ServerInfo* info = new ServerInfo();
    XStxIServerListener* listener = NULL;
    info->mApp = NULL;
    info->mServer = server;

    /** Instantiate the hosted application */

    result = HostedApplication::createHostedApplication(
        server,
        applicationContext,
        mServerContext,
        info->mApp);

    if (result != XSTX_RESULT_OK)
    {
        goto exit;
    }

    /** Point the app to the server and start the app */

    result = info->mApp->setServer(info->mServer);
```

```
    if (result != XSTX_RESULT_OK)
    {
        goto exit;
    }

    result = info->mApp->start();

    if (result != XSTX_RESULT_OK)
    {
        goto exit;
    }

    /** Point the server to the app and start the server */

    listener = info->mApp->getServerListener();
    if ( NULL != listener )
    {
        XSTX_CALLBACK_NOT_NULL_OR_ERROR(listener, ServerReady);
        XSTX_CALLBACK_NOT_NULL_OR_ERROR(listener, ServerStopped);
        XSTX_CALLBACK_NOT_NULL_OR_ERROR(listener, MessageReceived);
    }
    result = XStxServerSetListener(
        info->mServer,
        listener);

    if (result != XSTX_RESULT_OK)
    {
        goto exit;
    }

    result = XStxServerSetInputSink(
        info->mServer,
        info->mApp->getInputSink());

    if (result != XSTX_RESULT_OK)
    {
        goto exit;
    }
#ifdef APPLICATION_CAPTURES_AUDIO
    //The example audio source does provides timestamps
    result = XStxServerSetAudioSource(
        info->mServer,
        info->mApp->getAudioSource(), true);

    if (result != XSTX_RESULT_OK)
    {
        goto exit;
    }
#endif
    //need to manually change the isProvidingTimestamp flag here to false
    //if it doesn't provide timestamp
    result = XStxServerSetVideoSource(
        info->mServer,
        info->mApp->getVideoSource(), true);

    if (result != XSTX_RESULT_OK)
    {
        goto exit;
```

```
    }

    mServerToInfoMap[server] = info;

    return XSTX_RESULT_OK;

exit:

    if (info != NULL)
    {
        delete info->mApp;
        XStxServerRecycle(info->mServer);
        delete info;
    }

    return result;
}
```

# Streaming Video to a Client

In order to stream video frames to the client, your application calls the `XStxServerSetVideoSource` function of the `XStxServer` object to specify the source of video frames. For more information, see Initializing a Client Session (p. 173). You must also define the mode that your application will use to stream frames to the client.

## Choosing a Video Mode

To support a variety of applications, Amazon AppStream provides three strategies for streaming video:

- **Push with frame-rate blocking**—Calls to `XStxServerPushVideoFrame()` will block until enough time has elapsed to gate the video frame rate of the application to that specified by `XStxIVideoSourceFcnSetFrameRate()`. The amount of time blocked varies dynamically with changes to the target frame rate and the time it takes to generate frames. To use this strategy, set the video mode to `XSTX_VIDEO_MODE_PUSH_BLOCKING`.
- **Push immediately**—Calls to `XStxServerPushVideoFrame()` will return immediately. If calls are received faster than the frame rate indicated by the last call to `XStxIVideoSourceFcnSetFrameRate()` then frames will be dropped from the video stream to achieve the target frame rate. To use this strategy, set the video mode to `XSTX_VIDEO_MODE_PUSH_IMMEDIATE`.
- **Pull**—Video frames will be pulled from the application by calls to `XStxIVideoSourceFcnGetFrame()`. The thread making those calls will delay as necessary to limit the frame rate. The application is free to perform operations on this thread as long as it returns before the next frame is due. To use this strategy, set the video mode to `XSTX_VIDEO_MODE_PULL`.

The sample application uses the immediate push mode, as shown in the following excerpt from `main.cpp`. This file is in the *<SDK_dir>*`\example_src\server\windows\SimpleDirectXServer` directory.

```
static XStxResult getVideoMode(void* context, XStxVideoMode* mode)
{
    // We push frames to AppStream when we want to.
    // AppStream will adapt by dropping frames if we're too fast
    *mode = XSTX_VIDEO_MODE_PUSH_IMMEDIATE;
    return XSTX_RESULT_OK;
}
```

## Choosing a Color Subsampling Rate

Amazon AppStream streams the video at the YUV420 color subsampling rate to client applications. You can set your streaming application to stream video at the YUV444 color subsampling rate to a client application running on supported devices. The streaming application calls the `XStxServerAddChromaSamplingOption()` function to notify Amazon AppStream that the streaming application supports the YUV444 color subsampling option. The client application calls a similar function to also notify Amazon AppStream that the client application supports the YUV444 color subsampling option.

When the client application connects to the streaming application, Amazon AppStream compares the color subsampling options available on the client application and the streaming application. Amazon AppStream then selects the highest color resolution supported by both applications and then informs both applications which color sampling option will be used for the session. Amazon AppStream then calls the `XStxIServerListener2FcnServerConfigurationSettingsReceived` callback function that the streaming application supplied and passes a structure with the *`XStxChromaSampling`* setting.

> **Note**
> Streaming at the YUV444 subsampling rate requires higher bandwidth availability than the YUV420 rate.

The following code excerpt demonstrates how to use the YUV444 color subsampling option.

```
/**
 * Inform the server of server application's chroma sampling capability.
 * You can call this as many times as you want for each chroma sampling
 * listed in XStxChromaSampling. If this method is never called, then
 * XSTX_CHROMA_SAMPLING_YUV420 will be used by default chroma sampling.
 * Register a callback function at XStxIVideoSourceFcnSetChromaSampling
 * so that STX server can notify server application which chroma sampling
 * will be used for streaming.
 * @param[in] serverHandle The handle of the server.
 * @param[in] chromaSampling The chroma sampling scheme. Look at XStxAPI.h
 * @return This function will return one of these values.
 * Return code                  | Description
 * ---------------------------- | -------------------------------------------

 * XSTX_RESULT_OK               | The operation is successful.
 * XSTX_RESULT_INVALID_HANDLE   | serverHandle is invalid.
 * XSTX_RESULT_INVALID_ARGUMENTS | chromaSampling is not recognized.
 */
XSTX_API_EXTERN XStxResult XSTX_API XStxServerAddChromaSamplingOption(
    XStxServerHandle serverHandle,
    XStxChromaSampling chromaSampling);
```

## Sending Frames to the Client

If you are converting an existing application to work with Amazon AppStream streaming, you will need to redirect the video output to streaming clients instead of displaying video on the local machine. The following excerpt is from the `render` function in `main.cpp` and shows the changes you would make to render the video to clients instead of a local machine.

```
...

EnterCriticalSection(&g_frameCriticalSection); // Don't want to be interrupted
 now
```

```
// Copy back buffer data. Can also use D3DXLoadSurfaceFromSurface if we need
to resize/change pixel format
g_D3DDevice->GetRenderTargetData(g_backBuffer, g_memBuffer);
D3DLOCKED_RECT lockedRect;
g_memBuffer->LockRect(&lockedRect, NULL, D3DLOCK_READONLY);
// Convert to YUV so we can supply it to AppStream
switch (g_chromaSamplingType)
{
    case XSTX_CHROMA_SAMPLING_YUV420:
        convertToYUV420((unsigned char*)lockedRect.pBits, WINDOW_WIDTH, WIN
DOW_HEIGHT, 2, 1, 0, 4, lockedRect.Pitch, g_videoFrame.mPlanes);
        break;
    case XSTX_CHROMA_SAMPLING_YUV444:
        convertToYUV444((unsigned char*)lockedRect.pBits, WINDOW_WIDTH, WIN
DOW_HEIGHT, 2, 1, 0, 4, lockedRect.Pitch, g_videoFrame.mPlanes);
        break;
    default:
        assert(!"Unknown chroma sampling type"); // Make sure we don't get an
unknown chroma sampling type
}
g_memBuffer->UnlockRect();

XStxServerPushVideoFrame(g_serverHandle, &g_videoFrame); // Push the video frame

LeaveCriticalSection(&g_frameCriticalSection);

...
```

# Streaming Audio to a Client

There are two ways your application can transmit audio to clients:

- **Explicitly send audio frames to the client**—Explicitly sending audio frame-by-frame is best for situations in which you need tight integration between the video and audio frames, for example, if you are streaming speech synchronized with video of a person talking. To do so, call the `XStxServerSetAudioSource` function of the `XStxServer` object and set an audio source, you can then explicitly stream audio frames to the client by implementing the `XStxIAudioSourceFcnGetFrame` function of the `XStxIAudioSource` interface.
- **Automatically capture system audio and send that to the client** —This is the easiest way to stream audio and works best in cases where you do not need tight integration between audio and video, for example if you are streaming background music during a puzzle game. To automatically stream audio, do not set an audio source by calling the `XStxServerSetAudioSource` function. When no audio source is set, the `XStxServer` object automatically captures system audio and streams that to the client.

## Audio Timestamps

Timestamps are important for keeping the audio and video frames in synchronization. If your application uses automatic streaming of the system audio, the `XStxServer` object timestamps the audio frames using a built-in timestamp manager. You can also have the `XStxServer` object automatically timestamp your video frames by setting the `willProvideTimestamps` parameter to `false` when you call the `XStxServerSetVideoSource` function to set the video source of your application.

If you want to set the timestamps explicitly for your video frames and your application is using the automatic streaming of system audio, you can do so by having your application call the `XStxServerGetTimestampUs` function. This function returns the current mono timestamp from the `XStxServer` object, in microseconds.

# Receiving Content from a Client

To provide a fully interactive experience for users, the client collects data from user input sources (such as keyboard, mouse, or touch inputs) and sends that data to the application so the application can respond to the user action. There are three types of content that a client may send to your application:

- **Formatted user input**—such as keyboard, mouse or touch input.
- **Raw user input**—a stream of bytes from the device. This enables your application to support new or device-specific types of input such as a data stream from an accelerometer.
- **Client messages**—messages from the client to the application that are independent of user actions. These can be status messages, additional metadata, or other content as negotiated between the client and application developers.

Your application receives this content from the client as events sent by the `XStxServer` object to your callback functions. The following sections describe how to implement the callback functions to handle these events.

**Topics**

- Accepting Keyboard, Mouse, and Touch User Input (p. 178)
- Receiving Raw User Input from a Client (p. 179)
- Receiving Client Messages (p. 179)

## Accepting Keyboard, Mouse, and Touch User Input

When a client sends keyboard, mouse, or touch user input to your application, the `XStxServer` object fires an `XStxIInputSinkFcnOnInput` event. Your callback function to handle this event uses the `mType` member of the `XStxInputEvent` structure it receives to determine the input source (keyboard, mouse, or touch) and handle it appropriately.

The excerpt from `HostedApplicatonDirectX.cpp` shows an implementation of `XStxIInputSink`. The file is in the *<SDK_dir>*`\server\windows\XStxDirectXServer` directory. The sample streaming application only accepts keyboard input from client application so this implementation ignores mouse and touch input. Code comments show where you would add that functionality.

```
XStxResult HostedApplicationImp::XStxIInputSinkOnInput(const XStxInputEvent*
event)
{
    if (NULL == event)
        return XSTX_RESULT_INVALID_ARGUMENTS;

    if ((event->mType == XSTX_INPUT_EVENT_TYPE_KEYBOARD) &&
        (event->mInfo.mKeyboard.mIsKeyDown) &&
        (event->mInfo.mKeyboard.mVirtualKey == 78))
    {
        printf("Received N-key\n");
        std::string msg("Server received N-keyboard input!");
        if (XSTX_RESULT_OK == XStxServerSendMessage(
           mServer, (const unsigned char*) msg.c_str(), (uint32_t)msg.size()))
```

```
            printf("Successfully sent message to client\n");
        else
            printf("Failed to send message to client\n");
    }
    else
        if (NULL != mGame)
        {
            mGame->handleInput(event); // All other input is delegated to the
game
        }
    return XSTX_RESULT_OK;
}
```

### Receiving Raw User Input from a Client

When a client sends keyboard, mouse, or touch user input to your application, the `XStxServer` object fires an `XStxIRawInputSinkFcnOnRawInput` event. The data is transmitted as a raw stream of bytes. Your callback function to handle this event is responsible for interpreting the byte stream data. This requires close integration between your application and its client applications.

### Receiving Client Messages

The client may send messages to your application. These are not related to user input; the content and purpose of these messages is specific to the client. During application development, you should research the messages that may be sent by the client and implement code to handle those messages. Sending messages from the client is optional, and some clients may send no messages at all.

To receive and handle client messages, your application implements a callback function to handle `XStxIServerListenerMessageReceived` events. When a `XStxIServerListenerMessageReceived` event fires, it passes the message to your callback function as a byte array.

## Storing Persistent Data

When you host your application on Amazon AppStream, your application runs on an Amazon AppStream host in the cloud. When the session ends, the Amazon AppStream host is recycled, and any data stored on the Amazon AppStream host is lost. If your application needs to persist data between sessions, it should record the data in a persistent data store before terminating the session. You can record the data to a physical server or store the data on AWS using a service such as Amazon S3, Amazon RDS, or DynamoDB.

The `XStxServerManager` object fires an `XStxIServerManagerListenerFcnServerSaveState` event before terminating a session to give your application a chance to record data before the session ends. To do this, provide an implementation of `XStxIServerManagerListenerServerSaveState` that records data to a persistent data store.

The following excerpt from the sample streaming application shows an implementation of `XStxIServerManagerListenerServerSaveState`. It is from `ServerManagerListener.cpp`. This file is in the *<SDK_dir>*`\example_src\server\common` directory. This implementation does not persist data, but the comment shows where you would add that functionality.

```
XStxResult ServerManagerListenerImp::XStxIServerManagerListenerServerSaveState(
XStxServerHandle session,
uint32_t timeout,
XStxStopReason reason)
{
```

```
    // Add code here to record data to a persistent data store

    return XSTX_RESULT_OK;

}
```

## Terminating a Client Session

When a client session ends, the `XStxServerManager` object fires an `XStxIServerManagerListenerFcnServerTerminate` event. This gives your application a chance to gracefully release resources associated with the client session.

The following excerpt from the sample streaming application illustrates this step. It is from `ServerManagerListener.cpp`. This file is in the *<SDK_dir>*`\example_src\server\common` directory.

```
XStxResult ServerManagerListenerImp::XStxIServerManagerListenerServerTerminate(
XStxServerHandle session,
uint32_t timeout,
XStxStopReason reason)
{

    ...

    if (info != NULL)
    {
        XStxServerRecycle(info->mServer);
        delete info->mApp;
        delete info;
    }
    return XSTX_RESULT_OK;
}
```

## Terminating a Streaming Application

The `XStxServerManagerWait` function returns when a client session ends and Amazon AppStream terminates the session. Your application should release the resources allocated to the `XStxServerManager` and `XStxServerLibrary` objects.

The following excerpt from the sample streaming application illustrates this step. It is from the `runAsAppStreamGame` function of `XStxExampleServer.cpp`. This file is in the `<install_dir>\example_src\server\common` directory.

```
exit:

    XStxResult cleanUpResult = XStxServerManagerRecycle(serverManagerHandle);
    if (cleanUpResult != XSTX_RESULT_OK) {
        printf("Failed to recycle ServerManager\n");
    }

    cleanUpResult = XStxServerLibraryRecycle(serverLibraryHandle);
    if (cleanUpResult != XSTX_RESULT_OK) {
        printf("Failed to recycle ServerLibrary\n");
    }

    delete serverManagerListener;
```

```
    printf("Exit! %s\n", XStxResultGetName(result));
    fflush(stdout);

    return (result != XSTX_RESULT_OK) ? -1 : 0;
```

# Testing Your Streaming Application

Although you plan to stream the production version of your streaming application from the cloud using Amazon AppStream, you will likely want to run your streaming application locally or by using Amazon AppStream standalone mode during testing. This lets you test changes you make to the code quickly, without having to build an installer, upload your streaming application to Amazon S3, generated a presigned URL, and the other steps required to deploy an application on Amazon AppStream.

Developing a streaming application typically has the following stages:

1. **Test your application using Amazon AppStream standalone mode**—in this stage, the core functionality of your application is done, and you're adding streaming. By testing streaming on a standalone Amazon EC2 instance that is not managed by Amazon AppStream, you can connect to the Amazon AppStream host using a remote management service to modify your application code, add dependency files, or change the server configuration. For more information, see Streaming Your Application Using Amazon AppStream Standalone Mode (p. 181).
2. **Deploy your production-ready application on Amazon AppStream**—in this stage, development and testing of your application is done, and you are ready to deploy it on Amazon AppStream and have the service manage client sessions. For more information, see Deploying Your Streaming Application to Amazon AppStream (p. 38).

## Streaming Your Application Using Amazon AppStream Standalone Mode

Testing your application using Amazon AppStream standalone mode provides a way to test that streaming works without having the overhead of a full deployment to Amazon AppStream. It also gives you a way to test your application without having to write an entitlement service or a custom client by using a AWS CloudFormation template to create your own Amazon EC2 instance with the Amazon AppStream SDK libraries installed.

**You may incur charges when you use Amazon AppStream standalone mode.** In standalone mode, you are using your own EC2 instance rather than Amazon AppStream. Using standalone mode does not apply to the first 20 hours of streaming from Amazon AppStream.

> **Note**
> Do not use a standalone streaming server to deploy your streaming application for client access. Doing so will prevent you from taking advantage of the session management and automatic scaling provided by Amazon AppStream.

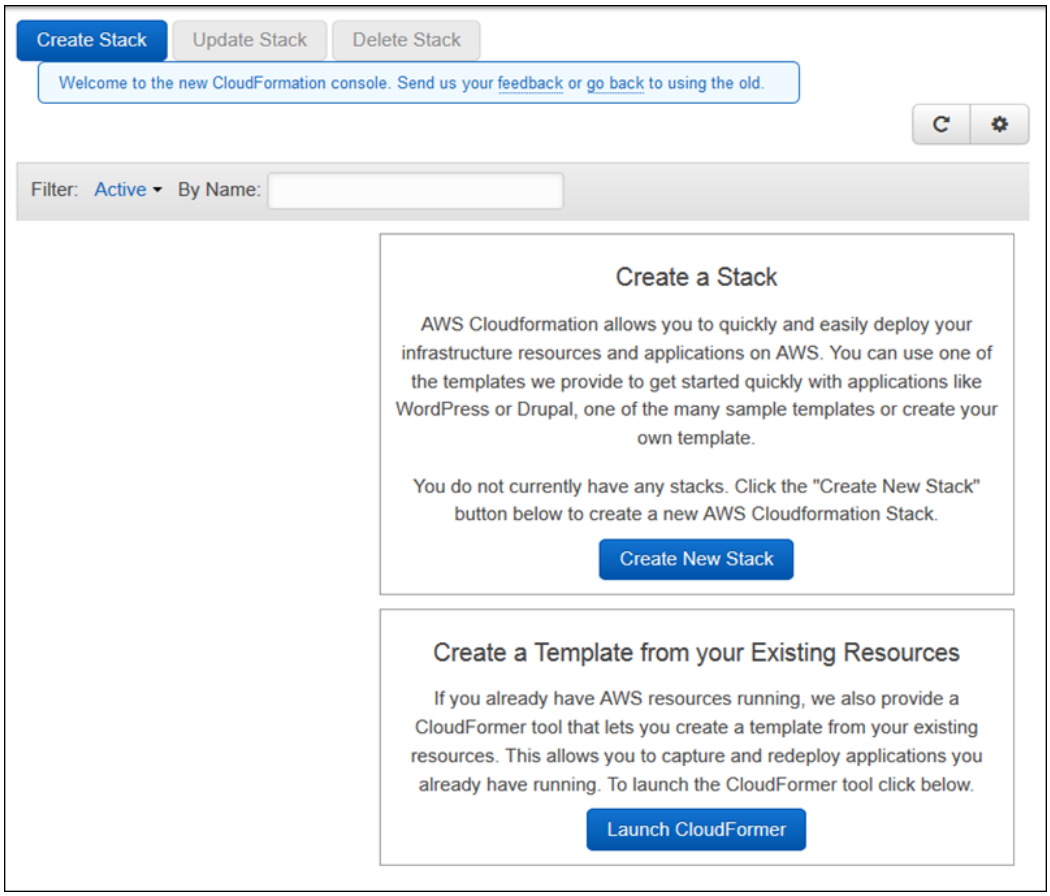
To use the AWS CloudFormation template described below, you must have:

- An active AWS account. You will need the access keys for the account or for an IAM user in the account. If you are using the access keys of an IAM user, that user must have permissions to perform Amazon AppStream actions. For more information, see AWS Security Credentials.
- An EC2 key pair. You can use this to connect to the instance that hosts the entitlement service with SSH. For more information, see Amazon EC2 Key Pairs.

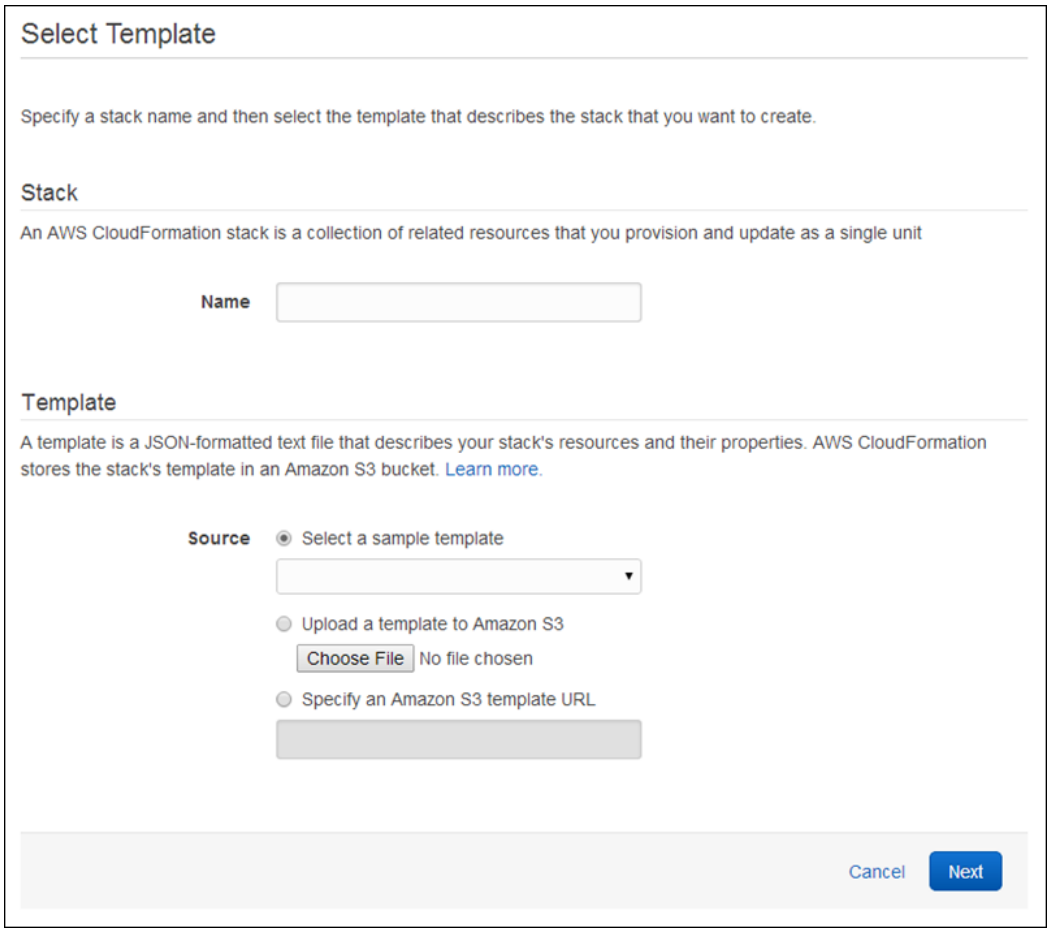
### To create the Amazon AppStream standalone mode

1. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. From the navigation bar, select the same region where you created a new key pair or are using an existing key pair.

   **Important**
   Do not select a region other than the one where your key pair was created.
3. In the AWS CloudFormation console, select **Create New Stack**.




4. For **Name**, type a name to identify the stack. For example, **`StandaloneStack`**.
5. Select **Specify an Amazon S3 template URL** and then enter **`https://s3.amazonaws.com/appstream-public/AppStreamDeveloper.template`** and then click **Next**.



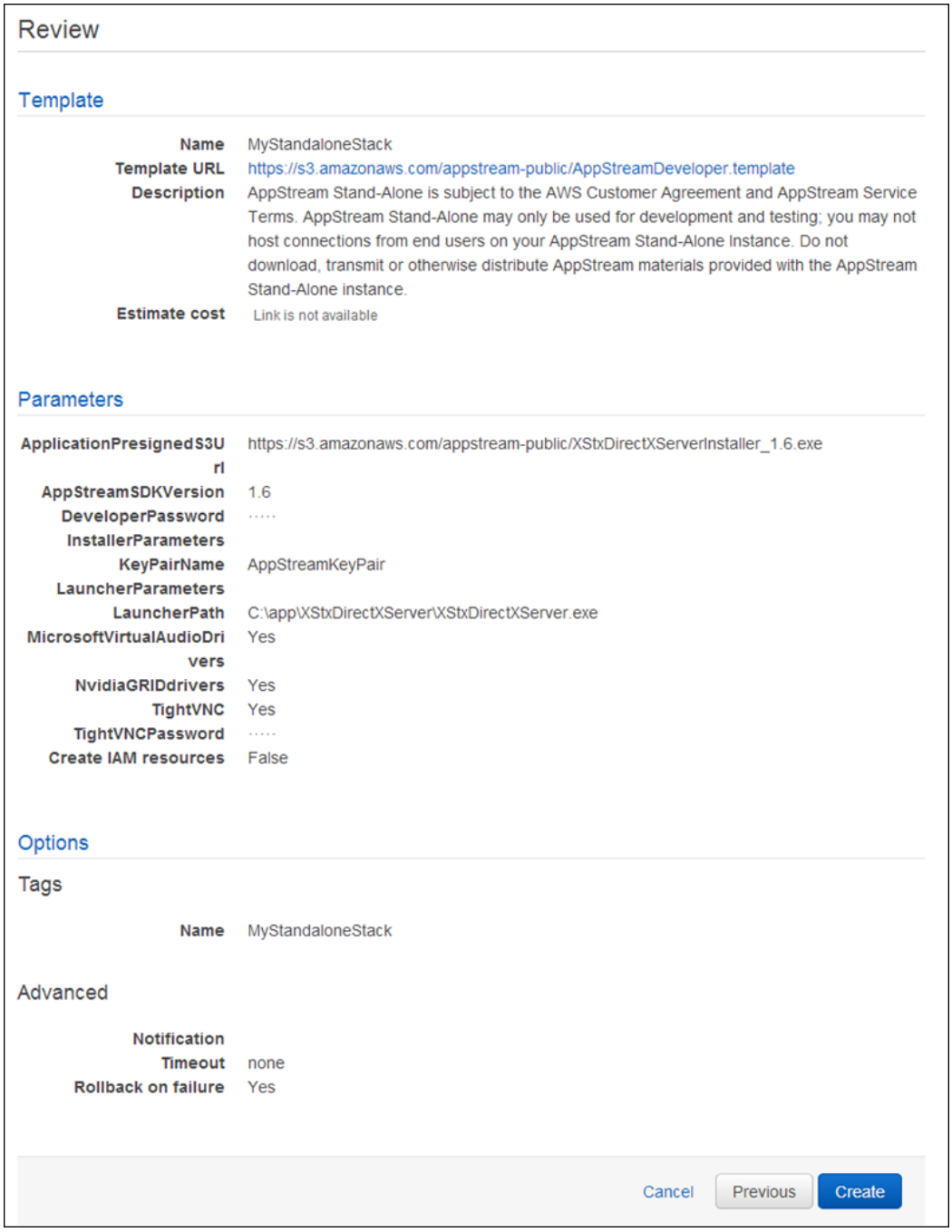

6. In **Specify Parameters**, do the following:

   - For **DeveloperPassword**, type a password that is between 12-20 characters that meets the following requirements:
     - Contains at least 2 numbers
     - Contains two lower case and two upper case letters
     - Contains two special characters (@, #, $, %, ^, &, +, or =)
     - Does not contain spaces
   - For **KeyPairName**, type the name of your existing key pair in the same region.
   - For **MicrosoftVirtualAudioDrivers**, type **Yes**.
   - For **NvidiaGRIDDrivers**, type **Yes**.
   - For **TightVNC**, type **Yes**.
   - For **TightVNC**, type an eight character password that meets the following requirements:
     - Must be a different password from the password used for **DeveloperPassword**.
     - Contain one number, one letter, one special character.
     - Contain one special characters (@, #, $ %, ^, &, +, or =)
     - Must not contain any spaces.
   - Click **Next**.

7. In **Options**, do the following:

- For **Key**, type **`Name`**.
- For **Value**, type the name that you created earlier.
- Click **Next**.

8. In **Review**, check that the parameters you entered are correct and then click **Create**.



When the status of your stack changes to **CREATE_COMPLETE**, the Amazon AppStream standalone mode is deployed and ready to use. You can view the properties of your new server in the Amazon EC2 console.

### To find the IP Address of your standalone host

1. In the AWS CloudFormation console, click on the name of your standalone host to display details about the stack at the bottom of the browser window.
2. In the details pane, click **Outputs**. This displays the following values:

| Key | Description |
|---|---|
| InstanceId | The identifier of the EC2 instance. You can use this value to locate the standalone host in the EC2 console. |
| PublicIp | The public IP address of the EC2 instance. You can use this value to connect the sample clients to your application without using an entitlement service. |
| PublicDnsName | The public DNS name of the EC2 instance. |

## Connecting to Your Standalone Host

After AWS CloudFormation finished deploying the stack for your standalone streaming server, you can use Remote Desktop Connection (RDC) to connect to the EC2 instance, upload your application, and start your application. For more information, see Connecting to Your Windows Instance in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.

To log into your EC2 instance, you will need the administrator password and public IP address to your EC2 instance. While retrieving the administrator password and public IP address, you can download a Remote Desktop file that contains the connection information to your EC2 instance.

### To get the administrator password, public IP address, and Remote Desktop File of your instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, select **Instances**. Select your instance, and click **Connect**.
3. In the **Connect To Your Instance** dialog box, record the public IP address and user name. You'll need the public IP address to stream to your client application. Click **Get Password** (it will take a few minutes after the instance is launched before the password is available).
4. Click **Browse** and navigate to the private key file that you saved when you created the key pair. Select the file and click **Open** to copy the entire contents of the file into contents box.
5. Click **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.
6. Record the default administrator password. You need this password to connect to the instance and install your application.
7. Click **Download Remote Desktop File**. Your browser prompts you to either open or save the `.rdp` file. Save the file as you might need this file to administer the instance. When you have finished, you can click **Close** to dismiss the **Connect To Your Instance** dialog box.

## Streaming Your Application

After you deploy your application in standalone mode, you can stream your application by connecting a client application to your EC2 instance. This client application does not call an entitlement service to connect to a streaming application, instead it uses the IP address of the hosted streaming application to connect. Because it does not use an entitlement service, the client application can only be used to connect to pre-release applications hosted locally or on a standalone GPU instance that you manually launched and configured in Amazon EC2. It cannot be used to connect to applications that are hosted on Amazon AppStream.

Despite this limitation, the client application is useful for testing your application during development because it gives you a known working client to test your streaming application against and you can develop and test your streaming application before creating a custom client application.

To stream your application to a client application, follow the instructions for your version:

- To stream to the Chrome App (p. 186)
- To stream to the Android client application (p. 186)
- To stream to the **Basic AppStream Client** (p. 186)
- To stream to the OS X client application (p. 187)
- To stream to the Windows client application (p. 187)

### To stream to the Chrome App

1. From the Chrome browser, go to https://chrome.google.com/webstore/detail/amazon-appstream-example/jooibckpjiaaoaffacacomklfdodlmah.
2. Click **FREE** and then click **LAUNCH APP**.
3. Click **Connect to Standalone Instance**.
4. In **Standalone instance URL or IP address**, type the IP address of your EC2 instance.
5. Click **Connect**.

### To stream to the Android client application

1. Install a file manager app on your device.
2. Allow your device to install application from unknown sources other than the Google Play store.
3. Download the client application from *http:// …* and install it on your device.
4. Start the **Amazon AppStream Example Client**.
5. Click **Connect to Standalone Instance**.
6. In **Standalone instance URL or IP address**, type the IP address of your EC2 instance.
7. Click **Connect**.

### To stream to the Basic AppStream Client

1. Download the client application from *http:// …* and install it on your device.
2. Start the **Basic AppStream Client**.
3. Click **Connect to Standalone Instance**.
4. In **Standalone instance URL or IP address**, type the IP address of your EC2 instance.
5. Click **Connect**.

### To stream to the OS X client application

1. Download the client application from *http:// …* and install it on your device.
2. Start the **Amazon AppStream Example Client**.
3. Click **Connect to Standalone Instance**.
4. In **Standalone instance URL or IP address**, type the IP address of your EC2 instance.
5. Click **Connect**.

### To stream to the Windows client application

1. Download the client application from *http:// …* and install it on your device.
2. Start the **Amazon AppStream Example Client**.
3. Click **Connect to Standalone Instance**.
4. In **Standalone instance URL or IP address**, type the IP address of your EC2 instance.
5. Click **Connect**.

## Cleaning Up Resources

When you are done testing your application, you should release the AWS resources you used to deploy the standalone streaming server to prevent further charges from accruing. To do so, you delete the AWS CloudFormation stack you created to deploy the standalone server, see Deleting a Stack on the AWS CloudFormation Console in the **AWS CloudFormation User Guide**.

# Debugging Your Streaming Application in Amazon AppStream Standalone Mode

You can debug your streaming application in Amazon AppStream standalone mode by installing the Remote Tools for Visual Studio on your Amazon EC2 instance. Remote Tools for Visual Studio allows you to debug your streaming application on the EC2 instance from another computer running Visual Studio.

### To install Remote Tools for Visual Studio

1. From your computer, start Remote Desktop Connection (RDC) and connect to the EC2 instance. For more information, see Connecting to Your Windows Instance in the *Amazon Elastic Compute Cloud Microsoft Windows Guide*.
2. From the EC2 instance, download and install the version of Remote Tools for Visual Studio that matches the version of Visual Studio on your computer.

   - Remote Tools for Visual Studio 2013 at http://www.microsoft.com/en-us/download/details.aspx?id=40781.
   - Remote Tools for Visual Studio 2012 at http://www.microsoft.com/en-us/download/details.aspx?id=38184.

3. Restart the EC2 instance and connect to it using RDC.

   **Note**
   Restarting the EC2 instance may change its Public IP address.

### To configure Remote Tools for Visual Studio

1. From the **Start** menu of the EC2 instance, click **Remote Debugger**.

2. From **Visual Studio Remote Debugging Monitor**, click **Tools > Options**.
3. In the **Options** dialog box, do the following:

   a. For **TCP/IP port number**, type a port number of your choice. In this section, the port number is designated by **`N`**.
   b. For **Authentication mode**, select **Windows Authentication**.
   c. Click **OK**.

### To configure your EC2 instance

In this procedure, you will configure the security group for your EC2 instance to accept inbound and outbound traffic from a specific TCP port. By limiting the traffic to a specific port during testing, you are reducing the chances of a malicious attack on your EC2 instance.

1. From your computer, open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, click **Instances**. Select your EC2 instance and click the link to the security group in the **Description** tab.
3. In the details pane, on the **Inbound** tab, click **Edit**.
4. In the dialog, click **Add Rule**, and then select **Custom TCP Rule** from the **Type** list. For **Port Range**, type the port number you entered in the earlier step. In the **Source** field, select **Anywhere**.
5. Click **Save**.

### To start debugging

1. From your computer, start Visual Studio.
2. From the **Tools** menu, click **Attach to Process**.
3. In the **Attach to Process** dialog box, do the following:

   a. For **Transport**, select **Default**.
   b. For Qualifier, type the Public IP address of the EC2 instance using the form, ***`xxx.xxx.xxx.xxx`*****`:`*****`N`***.

      Where

      - *`xxx.xxx.xxx.xxx`* is the Public IP address of your EC2 instance.
      - *`N`* is the port number you specified in an earlier step.

   c. Select **Show process from all users**.
   d. In the **Available Process** list, select the process of your streaming application and then click **Attach**.

# Security considerations

## Using IAM to Control Access to Amazon AppStream Resources

Amazon AppStream integrates with AWS Identity and Access Management (IAM), which allows you to control access to Amazon AppStream.

For general information about IAM, go to:

- Identity and Access Management (IAM)
- IAM User Guide

You can give IAM users of your AWS account access to all Amazon AppStream operations or to a subset of them. The following is the list of Amazon AppStream operations that can be made available to IAM users.

```
appstream:GetApiRoot
appstream:GetApplications
appstream:GetApplication
appstream:GetApplicationStatus
appstream:GetApplicationErrors
appstream:GetApplicationError
appstream:CreateApplication
appstream:UpdateApplication
appstream:DeleteApplication
appstream:UpdateApplicationState
appstream:GetSessions
appstream:GetSession
appstream:GetSessionStatus
appstream:CreateSession
appstream:UpdateSessionState
```

### Example IAM User Policies for Amazon AppStream

By default, IAM users have no access to Amazon AppStream or to the resources that it uses. If you want IAM users to be able to work with Amazon AppStream, for example, in the AWS Management Console, you must grant them permissions.

This section shows simple policies for controlling access to Amazon AppStream. To use these policies, you create an IAM user and attach one of these policies to the user or to the IAM group that the user belongs to.

#### Giving users access to DynamoDB and a specific set of AppStream API calls

The following policy lets users access DynamoDB.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1394569913000",
      "Effect": "Allow",
      "Action": [
```

```
        "dynamodb:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1394569933000",
      "Effect": "Allow",
      "Action": [
        "appstream:CreateSession",
        "appstream:GetApiRoot",
        "appstream:GetApplication",
        "appstream:GetApplications",
        "appstream:GetApplicationStatus",
        "appstream:GetSession",
        "appstream:GetSessions",
        "appstream:GetSessionStatus",
        "appstream:UpdateSessionState"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
                }
```

### Giving IAM users broad access to Amazon AppStream

The following policy lets users perform any Amazon AppStream action.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "appstream:*"
      ],
      "Resource": ["*"]
    }
  ]
}
```

### Giving users permission to modify applications and sessions

The following policy grants users the permission to create, update, and delete applications and sessions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "appstream:Create*",
        "appstream:Update*",
```

```
        "appstream:Delete*"
      ],
      "Resource": ["*"]
    }
  ]
}
```

### Giving users permission to modify applications

The following policy grants users the permission to deploy applications on Amazon AppStream and to update or delete those applications.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "appstream:CreateApplication",
        "appstream:UpdateApplication*",
        "appstream:DeleteApplication"
      ],
      "Resource": ["*"]
    }
  ]
}
```

### Giving users read-only access to Amazon AppStream

The following policy grants users read-only access to Amazon AppStream.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "appstream:Get*"
      ],
      "Resource": ["*"]
    }
  ]
}
```

# Security Best Practices

AWS has several features to help you keep your assets secure.

## Versioning

Versioning offers an additional level of protection by providing a means of recovery when customers accidentally overwrite or delete objects. This allows you to easily recover from unintended user actions and application failures. You can also use versioning for data retention and archiving. For more information, see Amazon Simple Storage Service FAQs and the Amazon Simple Storage Service Developer Guide.

### Multi-Factor Authentication

AWS multi-factor authentication (MFA) is an additional layer of security that offers enhanced control over your AWS account settings and the management of the AWS resources to which the account has subscribed. When you enable this opt-in feature, you need to provide a six-digit single-use code in addition to your user name and password before access is granted. You get this single use code from an authentication device or a special application on a mobile phone that you keep in your physical possession.

This feature is called multi-factor authentication because two factors are checked before access is granted to your account: you need to provide both your AWS email ID and password (the first factor: something you know) and the particular code from your authentication device (the second factor: something you have). You can enable multi-factor authentication for your AWS account as well as for the users you have created under your AWS account using IAM.

It's easy to obtain an authentication device from a participating third-party provider. You can also download and install appropriate software on your mobile phone, then set it up for use via the AWS website. For more information, see AWS Multi-Factor Authentication.

### Key Rotation

You should keep your AWS passwords and access keys safe for the same reasons it is important to change your password frequently. AWS recommends that you rotate your access keys and certificates on a regular basis. To let you do this without potential impact to the availability of your applications, AWS supports multiple concurrent access keys and certificates. With this feature, you can regularly rotate keys and certificates into and out of operation without any downtime to your application. This can help to mitigate risk from lost or compromised access keys or certificates. You can use the IAM APIs to rotate the access keys of your AWS account as well as for users created under your AWS account. For more information, see AWS Security Credentials.

### Use A Strong Password For Remote Management

Use a strong password with remote management services such as SSH and VNC to restrict access to your instances. If you do not configure a strong password with these remote management services, malicious users could access your instances.

### Restrict Access to Your Streaming Application

Restrict your security groups to allow connections only from ports required to support the necessary services. The following are suggested ports to allow incoming connections:

- **SSH**—port 22
- **STX**—port 80
- **STX TCP**—port 5900
- **STX UDP**—ports 9070-9080

For additional protection, restrict access to incoming traffic to a group of IP addresses.

# Build an Application Installer

To install your streaming application on Amazon AppStream, you will need a streaming application installer that does the following:

- Runs silently without any user interaction.

- Exists as a single executable file (`*.exe`). Do not use Microsoft Windows Installer files (`*.msi`), batch files (`*.bat`), or self-extracting zip (`*.zip`) files.
- Installs the streaming application and any required dependency files.
- Sets the necessary file and folder permissions so that a Windows standard user account can run the streaming application. for more information on Windows accounts, see "What is a standard user account?" at Microsoft.com.

# Building an Entitlement Service

An entitlement service authenticates and authorizes users. It is the gatekeeper between clients and your application, ensuring that only those clients entitled to access your application do so. Your entitlement service can authenticate users in a variety of ways: by comparing user login credentials to a list of subscribers in a database, by using an external login service such as Login with Amazon, or by simply authenticating all clients.

An entitlement service:

1. Processes requests from clients to connect to your application.
2. Authenticates user credentials.
3. Checks whether the user is authorized to access your application.
4. Calls into Amazon AppStream to create new client sessions for authorized users.
5. Returns an entitlement URL to authorized clients that the client uses to access your application.

You can download a sample entitlement service from Sample Entitlement Service (p. 11). To deploy the sample entitlement service on your own Amazon EC2 instance, see Design Considerations for Your Entitlement Service (p. 193).

For more information about the lifecycle of a connection request, see How a Client Application Connects to the Streaming Application (p. 6).

> **Note**
> While you could implement the entitlement logic directly in the client, doing so is strongly discouraged because of the requirement to call into the Amazon AppStream service to create new sessions. It is more secure to have your AWS credentials built into a web service running on a server you control than compiled into client code running locally on end-user devices.

**Topics**

- Design Considerations for Your Entitlement Service (p. 193)
- Building the Entitlement Web Service (p. 194)
- Deploying Your Entitlement Service (p. 199)
- Example Entitlement Request and Response (p. 199)
- Amazon AppStream Sample Entitlement Service (p. 200)

# Design Considerations for Your Entitlement Service

When you design the entitlement service for your application, there are several factors to consider:

- **Authentication mechanism** — How will you authenticate user credentials? Will you use an external service such as Login with Amazon (LWA), or an internal data store? If you have an existing log on

process defined for your customers, you can integrate it with your entitlement service for a seamless customer experience when you add applications to your offerings.

- **Web service hosting**—Your entitlement service is a web service and must be continuously available to clients. If your entitlement service becomes unavailable, clients will not be able to create new connections. Because of this, you should host your entitlement service on a reliable platform. You can choose to host it on a physical server or in the cloud, on AWS infrastructure.
- **Service health monitoring**—Because your entitlement service is a crucial part of your product, you should monitor the health of your entitlement service and set alarms that trigger if it becomes slow to respond or unavailable. If you are hosting your entitlement service on AWS, you can use CloudWatch to monitor your entitlement service.

# Building the Entitlement Web Service

Your entitlement service is a web service, handling incoming HTTP requests from clients and returning an HTTP response. The following topics walk you through the process of creating an entitlement service.

To simplify the process of writing an entitlement service, Amazon AppStream provides the Amazon AppStream SDK for Java, which contains Java wrappers for the REST API of the Amazon AppStream service. These wrapper classes handle the overhead of signing your requests to the REST API and provide functions your entitlement service can call in order to create new client sessions. You can download the Amazon AppStream SDK for Java from the links in Downloads (p. 11).

To build an entitlement service in a language other than Java, you can send HTTP requests directly to the Amazon AppStream REST API (p. 216).

**Topics**

- Authenticating the Client (p. 194)
- Checking Client Authorization for the Application (p. 195)
- Requesting a New Session from Amazon AppStream (p. 196)
- Returning an Entitlement URL to the Client (p. 197)
- Sample Entitlement Service Source Code (p. 198)

## Authenticating the Client

The sample entitlement service stores user credentials and entitlement mappings of users to applications in DynamoDB, a fast, fully managed NoSQL database service. For more information, see the DynamoDB Developer Guide.

The following excerpt from `/rs/JaxRsEntitlementService.java` illustrates how the `JaxRsEntitlementService.requestEntitlement` method authenticates a user. This code resides in a `try-catch` block and throws an exception if a method call fails.

```
//look up the user credentials in DynamoDB
User user = entitlementService.getUserFromAuthorization(authorization);
```

The following excerpt from `/services/EntitlementService.java` shows the implementation of the `EntitlementService.getUserFromAuthorization` method.

The `getUserFromAuthorization` method looks up user credentials in an DynamoDB data store. If the user is not found, and the `createUserWhenNew` flag is not set, the method throws an exception. If the `createUserWhenNew` flag is set, the method creates a new user in the DynamoDB data store and populates it with the user credentials passed into the method. By default, `createUserWhenNew` is set to `true`.

```
public User getUserFromAuthorization(String authorization)
        throws AuthorizationException {
    if (authorization == null) {
        throw new AuthorizationException("Missing Authorization header.");
    }

   Identity identity = authorizationHandler.processAuthorization(authorization);


    User user = dynamoDBMapper.load(User.class, identity.getId());

    if (user == null) {
        if (!createUserWhenNew) {
            log.warn("No such user: " + identity.getId());

            throw new UserNotFoundException();
        }

        user = new User();

        user.setId(identity.getId());
        user.setName(identity.getName());   // May be null
        user.setEmail(identity.getEmail()); // May be null
        user.setEntitleAll(entitleAllWhenNew);
        user.setSessionCount(0);

        dynamoDBMapper.save(user);
    }

    return user;
}
```

## Checking Client Authorization for the Application

After your entitlement service has verified that the user credentials are valid, it should check whether the user holding those credentials is authorized to connect to the application.

The following excerpt from `/rs/JaxRsEntitlementService.java` illustrates how the `JaxRsEntitlementService.requestEntitlement` method checks whether a user is authorized to access an application. The `User` object passed into the `checkIfEntitled` method was created in the previous step, Authenticating the Client (p. 194).

```
//call Amazon AppStream to look up the application from its ID
Application application = entitlementService.getApplication(applicationId);

//check to see whether the user is entitled to access the application by looking
 up entitlement mappings in DynamoDB
entitlementService.checkIfEntitled(user, application);
```

The following excerpts from `/services/EntitlementService.java` shows the implementation of the `EntitlementService.getApplication` and `EntitlementService.checkIfEntitled` methods.

The `getApplication` method calls Amazon AppStream to retrieve information about an application based on an application identifier. If the application is found, the method returns an Application object populated with metadata about the application. If the application does not exist in Amazon AppStream, the method throws an exception.

```
public Application getApplication(String applicationId)
        throws ApplicationNotFoundException {
    try {
        return appstream.getApplications().getById(applicationId);
    } catch (Exception e) {
        log.error(e);

        throw new ApplicationNotFoundException("The application identified by
" + applicationId + " was not found.");
    }
}
```

The `checkIfEntitled` method checks the DynamoDB data store to see whether the user is entitled to access the application. Adding user-application mappings in the sample entitlement service is handled through a web-based interface. .

```
public void checkIfEntitled(User user, Application application)
        throws UserNotEntitledException {

    if (user == null || !user.isEntitled(application.getId())) {
        throw new UserNotEntitledException("You are not currently allowed to
use this application.  Please ask the developer for access.");
    }
}
```

The implementation of the `isEntitled` method of the `User` class is located in `/model/User.java`.

## Requesting a New Session from Amazon AppStream

After your entitlement service has verified that the user is authorized to access the application, it calls the `entitleSession` function of the current `Application` object to create a new client session for that application. If successful, the function returns the newly created `Session` object.

The following excerpt from `/services/EntitlementService.java` illustrates how to make this call.

When you call `entitleSession`, you have the option of passing in binary data using the `opaqueData` member of the `EntitleSessionInput` object. If supplied, this data is passed to the application when the new session is created. The format and content of this data is specific to the application. You might use `opaqueData`, for example, to initialize the client session with state information saved the last time this user connected to your application.

```
public Session entitleSession(User user, Application application) {
    Session session;

    try {
        EntitleSessionInput entitleSessionInput = new EntitleSessionInput();

       // Could optionally pass opaque data to the application that will service
 the session.
       // entitleSessionInput.setOpaqueData("some application-specific data");


        session = application.entitleSession(entitleSessionInput);
    } catch (AmazonServiceException e) {
        // Make the exception just a bit more obvious with an informational
```

```
message.
        // It may not be obvious where the error occurred with just a stack
trace.
        if (e.getErrorType().equals(AmazonServiceException.ErrorType.Service))
 {
          log.error("An error occurred in the AppStream service while entitling
 a session for app: " + application.getId(), e);
        }

        throw e;
    }

    user.storeSessionId(application.getId(), session.getId());
    user.setSessionCount(user.getSessionCount() + 1);

    dynamoDBMapper.save(user);

    return session;
}
```

## Returning an Entitlement URL to the Client

The following excerpt from `/rs/JaxRsEntitlementService.java` uses JAX-RS, a Java API for interacting with REST-ful web services. It processes incoming HTTP requests from clients to establish a connection to an application. If the client is entitled to access the application, this method returns an entitlement URL that the client can use to connect to the application. If the request is denied, it returns an error.

The logic exposed by this function is the entitlement service in a nutshell.

```
@POST
@Path("/{applicationId}")
@Produces(MediaType.TEXT_PLAIN)
public Response requestEntitlement(@HeaderParam("Authorization") String author
ization,
                                   @PathParam("applicationId") String applica
tionId,
                                   @FormParam("terminatePrevious") @Default
Value("false") Boolean terminatePrevious) {
    try {
       User user = entitlementService.getUserFromAuthorization(authorization);

        Application application = entitlementService.getApplication(applica
tionId);

        if (!entitlementService.shouldAlwaysEntitle()) {
            entitlementService.checkIfEntitled(user, application);
        }

        if (checkPreviousSession) {
            entitlementService.checkForPreviousSession(user, application, ter
minatePrevious);
        }

       Session session = entitlementService.entitleSession(user, application);
```

```
        return response(Status.CREATED, session.getEntitlementUrl());
    } catch (AuthorizationException e) {
        String authenticateHeader = e.getAuthenticateHeader();

        if (authenticateHeader == null) {
            return response(Status.UNAUTHORIZED, e.getMessage());
        } else {
            return response(Status.UNAUTHORIZED, e.getMessage(), Collec
tions.singletonMap("WWW-Authenticate", authenticateHeader));
        }
    } catch (UserNotEntitledException e) {
        return response(Status.FORBIDDEN, e.getMessage());
    } catch (ApplicationNotFoundException e) {
        return response(Status.NOT_FOUND, e.getMessage());
    } catch (SessionActiveException e) {
        return response(Status.CONFLICT, e.getMessage());
    }
}
```

## Sample Entitlement Service Source Code

You can download the source code for the sample entitlement from Sample Entitlement Service (p. 11). The compressed file extracts to the `sample-entitlement-service-src/sample-entitlement-service-1.1.0-SNAPSHOT/src/main/java/com/amazonaws/sample/entitlement` directory. The contents of that directory are organized as follows:

| Directory | Description |
| --- | --- |
| `/authorization` | Classes to authenticate user credentials and to check whether the user is authorized to access an application. This directory has the following authorization implementations:<br>• Amazon<br>• Facebook<br>• Github<br>• Policy-based authorization<br>• Twitter<br>• Username authorization |
| `/exceptions` | Classes that return authorization exceptions to the client. |
| `/https` | Classes that handle the HTTP response. |
| `/model` | Classes that interact with Amazon DynamoDB to persist and query user data. For more information, see Java: Object Persistence Model. in the *Amazon DynamoDB Developer Guide*. |
| `/rs` | Classes that use JAX-RS to handle HTTP requests and responses. |
| `/services` | Classes providing the entitlement service logic. |

| Directory | Description |
| --- | --- |
| `/ui` | Implementation of the web-based user interface used to add users and entitle them to access applications. |

# Deploying Your Entitlement Service

After you develop your entitlement service, you need to host it on a server where clients can send it HTTP requests. You host your entitlement service as you would any other web service. You can host it on a physical server or in the cloud.

The sample entitlement service includes a AWS CloudFormation template to automate the process of deploying it on the AWS cloud and instructions for how to use that template to deploy the sample entitlement service. For more information, see Amazon AppStream Sample Entitlement Service (p. 200).

If you host your entitlement service on AWS, there are several services that can help:

- **Amazon Elastic Compute Cloud (Amazon EC2)**—Launches an Amazon AppStream host that runs your web server and performs server-side processing. For more information, see the Amazon EC2 documentation.
- **AWS Elastic Beanstalk**—Automatically creates, deploys, and manages the IT infrastructure needed to run a custom application. For more information, see the Elastic Beanstalk documentation.
- **AWS CloudFormation**—Automatically deploys IT infrastructure on AWS using templates. For more information, see the AWS CloudFormation documentation.
- **Amazon CloudWatch**— Collects and reports metrics on your AWS resources. For more information, see the CloudWatch documentation.

In addition to the content above, AWS provides two guides that walk you through the process of hosting an web application. For more information, see Getting Started with AWS: Web Application Hosting for Microsoft Windows and Getting Started with AWS: Web Application Hosting for Linux.

# Example Entitlement Request and Response

When a client attempts to connect to your application, it sends an HTTP request to your entitlement service. In the request, the client transmits an application identifier and a set of user credentials. Your entitlement service attempts to authenticate the user credentials and to authorize those credentials for the application. If the credentials are successfully authorized, your entitlement service returns a response indicating success that contains the entitlement URL that the client will use to connect to the application.

If authorization is not successful, your entitlement service returns a response that should include some indication of the reason authorization was not granted. Clients should include logic to gracefully handle failed authorizations.

### Example Request from the Client

The following shows an example of the HTTP headers and JSON body of a request sent from the client to an entitlement service. It includes user credentials and an application ID.

```
POST /api/entitlements/5565ba3a-7e75-4bce-baad-436843ad209e HTTP/1.1
User-Agent: curl/7.24.0 (x86_64-apple-darwin12.0) libcurl/7.24.0 OpenSSL/0.9.8y
 zlib/1.2.5
Host: localhost:8080
Accept: */*
Authorization:Username myUserId
Content-Length: 0
Content-Type: application/x-www-form-urlencoded
```

### Example Response from the Entitlement Service

The following shows an example of the HTTP headers and JSON body of a response sent from the entitlement service to a client. This illustrates an successful client authorization and returns an entitlement URL.

```
HTTP/1.1 201 Created
Content-Type: text/plain
Date: Fri, 01 Nov 2013 19:23:43 GMT
Content-Length: 97
Server: Jetty(9.0.6.v20130930)
https://appstream.us-east-1.amazonaws.com/entitlements/e018add2-242e-4396-8e47-
ca5cd1a6060b
```

# Amazon AppStream Sample Entitlement Service

The Amazon AppStream SDK includes a sample entitlement service with source code that you can use with your streaming application. The source code is available from the links in Downloads (p. 11).

You can deploy the sample entitlement service by doing the following steps:

**Topics**


- Step 1: Create a Key Pair (p. 200)
- Step 2: Create a Custom Policy with the Required Permissions (p. 201)
- Step 3: Create a Group and a User to Deploy the Entitlement Service (p. 202)
- Step 4: Deploy the Sample Entitlement Service (p. 203)
- Step 5: Configure the Sample Entitlement Service (p. 204)


## Step 1: Create a Key Pair

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

To deploy the sample entitlement service, you need a key pair. You can use an existing key pair that is stored in the same region where you will deploy the sample entitlement service. If you want to use an existing key pair, go to Step 2: Create a Custom Policy with the Required Permissions (p. 201).

If you do not have a key pair, use the following procedure to create one.

### To create a key pair

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From region selector in the top navigation bar, select the region where you want to deploy your streaming application.
3. In the navigation pane, click **Key Pairs**.
4. Click **Create Key Pair**.
5. For **Key pair name** enter a name and then click **Create**.
6. The private key file (*KeyPairName*.pem) is automatically downloaded by your browser. Save the private key file in a safe place.

   **Important**
   This is the only chance for you to save the private key file. You will need this file to connect to the instance you will create in the next step.

Your new key pair appears in the Amazon EC2 console. In the next step, you will create a group or user with the correct permissions that will deploy the sample entitlement service.

For more information about key pairs, see Creating Your Key Pair Using Amazon EC2 in the *Amazon Elastic Compute Cloud User Guide*.

## Step 2: Create a Custom Policy with the Required Permissions

After you create a key pair, you will need to create a policy with the appropriate permissions that a group or user will use to deploy the entitlement service. You create this policy in the AWS Identity and Access Management service. IAM is a web service that enables AWS customers to manage group or user permission in AWS.

### To create a policy to deploy the entitlement service.

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Policies** and then click **Create Policy**.
3. In **Create Your Own Policy**, click **Select**.
4. In **Review Policy**, do the following:

   a. For **Policy Name**, type a name to identify this policy. The name will appear in the AWS CloudFormation console.
   b. Leave **Description** blank.
   c. For **Policy Document**, copy and paste the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1403040000000",
      "Effect": "Allow",
      "Action": [
        "appstream:CreateSession",
        "appstream:GetApiRoot",
        "appstream:GetApplication",
        "appstream:GetApplications",
        "appstream:GetApplicationStatus",
        "appstream:GetSession",
```

```
        "appstream:GetSessions",
        "appstream:GetSessionStatus",
        "appstream:UpdateSessionState"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1403040053000",
      "Effect": "Allow",
      "Action": [
        "dynamodb:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "Stmt1403040077000",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

d. Click **Create Policy**.

## Step 3: Create a Group and a User to Deploy the Entitlement Service

After creating the policy, you will need a user or group with that policy. You can use an existing group or user to deploy the sample entitlement service by adding the appropriate permissions.

### To create a group with the required permissions

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Groups** and then click **Create New Group**.
3. In the **Group Name** box, type the name of this new group and click **Next Step**.
4. In **Filter** of **Attach Policy**, select **Customer Managed Policies**, select your policy, and then click **Next Step**.
5. In **Review**, check that you selected the correct policy and then click **Create Group**.

### To create a user and add that user to the group

1. In the navigation pane, click **Users** and then click **Create New Users**.

2. For **Enter User Names**, type a name for your user. Select **Generate an access key for each user** and then click **Create**.
3. In the **Create User** dialog box, click **Download Credentials** and save the file to a safe place. The comma separated value file contains the access key and secret key that you will need to deploy the sample entitlement service.

   **Important**
   Store this information in a secure place. This is the only time you will be able to get the secret key. If you lose this information, you will need to create a new access key and secret key.
4. Click **Close**.
5. In the navigation pane, click **Groups**, and then click the name of the group.
6. In **Summary**, click **Add Users to the Group**.
7. Click **Add Users to Group**. Select the user you created in the previous step and then click **Add Users**.

Verify that your user is a member of this group.

## Step 4: Deploy the Sample Entitlement Service

You will use AWS CloudFormation to deploy the sample entitlement service. When deployment is complete, you will get the URL for the entitlement service. You will need the URL to configure the entitlement service and connect the client application to the streaming application.

### To deploy the sample entitlement service on AWS CloudFormation

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Click **Actions** and then click **Create Stack** or **Create New Stack**.
3. On the **Select Template** page, do the following:

   a. For **Name** under **Stack**, type a name for the entitlement service.
   b. For **Source**, select **Specify an Amazon S3 template URL** and then type `https://s3.amazonaws.com/appstream-sdk/appstreamEntitlementService.template`.
   c. Click **Next**.

4. In **Specify Parameters**, do the following:

   a. For **KeyName**, type the name of your key pair.
   b. For **Password**, type a password with at least eight characters that meets the following requirements:

      - Contain one number, one letter, one special character.
      - Contain one special characters (@, #, $ %, ^, &, +, or =).
      - Must not contain any spaces.

   c. For **UserAccessKey**, type the Access Key Id of the user you created in the previous step. This value is in the credential file that you downloaded and stored in a safe place when you created the user.
   d. For **UserSecretKey**, type the Secret Access Key of the user you created in the previous step.
   e. Click **Next**.

5. On the **Options** page, click **Next**.
6. On the **Review** page, review the values for the configuration of the stack to and then click **Create** to launch the stack.

This process may take several minutes to complete. While the stack is launching, its status is set to **CREATE_IN_PROGRESS**.

When the status of your stack changes to **CREATE_COMPLETE**, your entitlement service is deployed and ready to use. You will need to get the URL of the entitlement service to connect your client application to your streaming application.

### To locate the URL of the sample entitlement service

1. In the AWS CloudFormation console, select the sample entitlement service you just deployed.
2. Click the down arrow in the **Events** tab and click **Outputs** to display details about the stack at the bottom of the browser window. The **PublicDNS** key displays the URL of the sample entitlement service.

## Step 5: Configure the Sample Entitlement Service

After you successfully deployed the sample entitlement service, you need to configure this service to allow specific users to connect to your streaming application. The sample entitlement service is on a host with port 8080 open. To configure the entitlement service, go to `http://publicDNS:8080/web/` to open the configuration page.

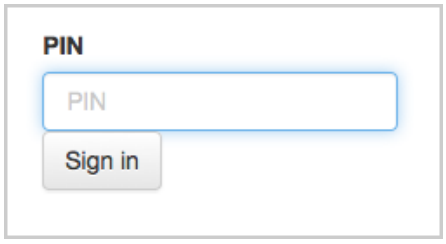
> **Important**
> Always include a trailing forward slash ("/") at the end of the URL to access the entitlement service.

### To configure the sample entitlement service

1. Make a list with the User IDs and email addresses of the users who will connect to your streaming application.
2. In a browser, open `http://publicDNS:8080/web/`.

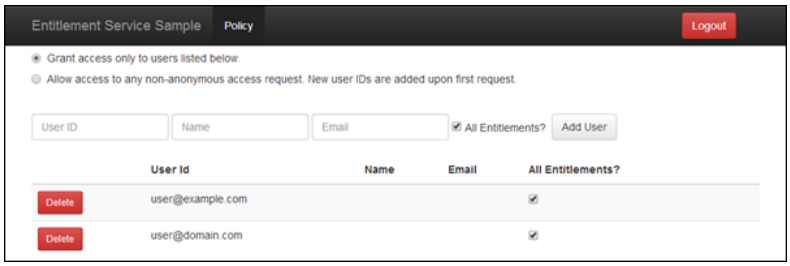
   > **Note**
   > If you cannot open the configuration page, see Sample Entitlement Service Problems (p. 209) for more information.

3. For **PIN**, type the entitlement service password and click **Sign In**.




4. In the **Entitlement Service Sample** page, click **Grant access only to users listed below**.
5. For each user that you want to connect to the streaming application, type their User ID, Name, and email address, select **All Entitlements**, and click **Add User**. The users you add to this page can connect to your streaming application. Other users will receive an error message.

   By default, the email address of the AWS account that created the entitlement service and the email address user@domain.com have can connect to the streaming application.




After you configure the sample entitlement service, send the Public DNS of the entitlement service to the users that will connect to the streaming application.

# Amazon AppStream Opaque Data Specification

Opaque data sent from the entitlement service is stored in the `AppStream_OpaqueData.xml` file located in the `C:\Users\Public\AppStream` folder of the streaming application. The entitlement service generates the XML file.

This XML file uses the following format:

```
<AppStream>
    <Timestamp>yyyy-mm-dd.hh:mm:ss</Timestamp>

    <OpaqueData>
        <![CDATA[OpaqueData]]>
    </OpaqueData>

</AppStream>
```

Note the following:

- Amazon AppStream generates this XML file and then sends the file to the streaming application when a client application connects to the streaming application.
- If your streaming application relies on this file, make sure that the file exists before executing commands that require this file.
- The timestamp indicates when Amazon AppStream creates the file on the streaming application.
- The *OpaqueData* data block cannot contain two right square brackets (]]).

# Using Environment Variables

Environment variables override configuration settings and can be useful for scripting or temporarily setting a condition as the default.

The following variable is supported by Amazon AppStream.

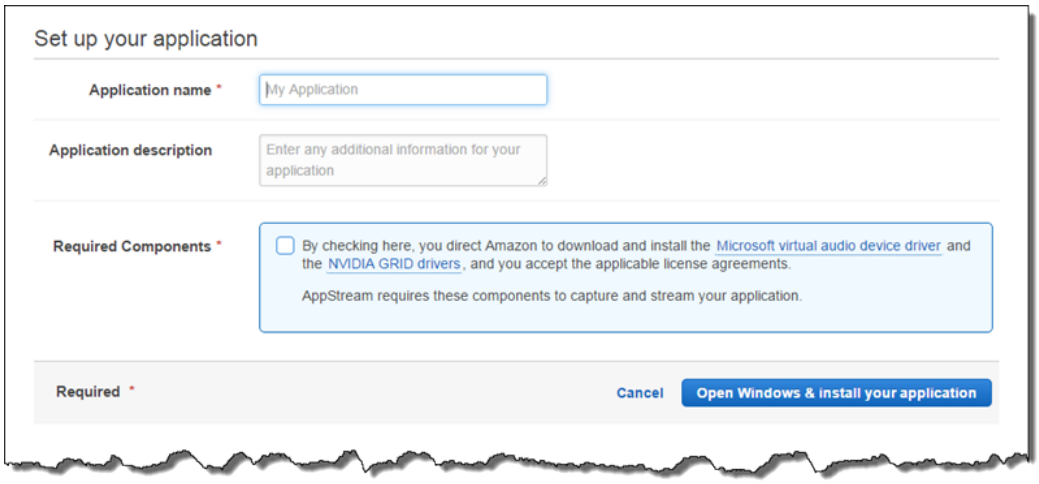
**VDI_CAPTUREMOUSECURSOR –** When set to true, Amazon AppStream displays the remote mouse pointer of the streaming application. By default, this variable is false, which disables the remote mouse pointer. For more information, see Displaying Mouse Pointers (p. 53).

### To set an environment variable

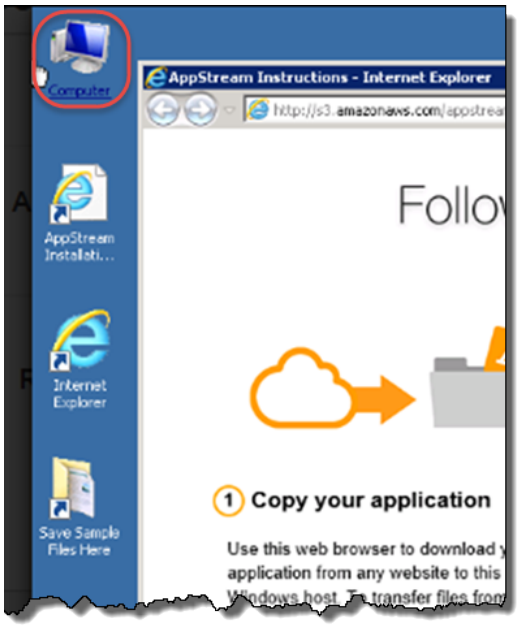
You set an environment variable as part of deploying a streaming application.

1. Sign in to the AWS Management Console and open the Amazon AppStream console at https://console.aws.amazon.com/appstream/.
2. Click **Deploy an application**.
3. In **Set up your application**, do the following:

   - For **Application name**, type the name of your streaming application. This text identifies your streaming application in the console.
   - For **Application description**, type a short description of the streaming application. This text appears in the application summary page.
   - Select **Required Components**.
   - Click **Open Windows & install your application** to display the Windows live host desktop.




4. In the live Windows host desktop, click **Computer**.




5. Navigate to the `C:\Windows\System32` folder.
6. Click `cmd.exe`.
7. In the command prompt, type **`setx`** ***`<environment variable>`*** where *`<environment variable>`* is the environment variable and setting. For example, type the following to configure Amazon AppStream to use the remote mouse pointer:

   ```
   setx VDI_CAPTUREMOUSECURSOR true /m
   ```

8. Press Enter.
9. Type **`exit`** and press Enter to close the command prompt.
10. Continue with the deployment.

# Troubleshooting Amazon AppStream

If you encounter a problem with your product, the following topics offer information about possible causes of your issue as well as proposed solutions.

If you are still having a problem, check the Amazon AppStream forum or contact Amazon AppStream support.

**Topics**


- Deployment Problems (p. 208)
- Sample Entitlement Service Problems (p. 209)
- Streaming Problems (p. 210)
- Error Codes (p. 212)


## Deployment Problems

If your deployment failed, try the following:

- Install all dependency files and drivers required by your application.
- Modify your streaming application to not require elevated/administrator privileges to run.
- Modify your application to only use a single top-level process. Amazon AppStream monitors and streams your application based on the top-level process. Amazon AppStream also captures child processes directly created by the top-level process. Any other processes created by your application will not be streamed.
- Check that you are using the correct launch path and parameters for your streaming application. To change the launch path or parameters, see Editing an Application (p. 60).

If you still cannot deploy your application after trying these items, contact Amazon AppStream support.

**Topics**


- Cursor and Screen Elements Move Very Slowly in Windows (p. 209)
- Error: Installer Cannot Find a GPU, Graphics Card, or Sound Card (p. 209)

## Cursor and Screen Elements Move Very Slowly in Windows

When you see the Windows live host desktop during your streaming application deployment, you are connecting to the instance using technology based on the Windows Remote Desktop Protocol (RDP) instead of the Amazon AppStream STX protocol. The technology based RDP is slower that the STX protocol and does not use the GPU in the instance. When you use a client application to connect to your streaming application, you will be using the STX protocol along with the GPU, which is faster than using the Windows RDP.

## Error: Installer Cannot Find a GPU, Graphics Card, or Sound Card

When you install a streaming application, the Windows Remote Desktop Protocol (RDP) turns off access to GPU, graphics cards, and sound cards. As long as your application installer installs the dependency files and drivers for these devices, your application will start and run correctly when you stream using Amazon AppStream.

# Sample Entitlement Service Problems

If you encounter an issue with deploying the sample entitlement service, try the following solutions.

**Topics**


- Error: Remote Server Not Responding (p. 209)
- Error: Web Page Not Available (p. 209)
- Error: 503 (p. 209)


## Error: Remote Server Not Responding

Check to see that you added **`:8080/web/`** to the end of the URL. To access the sample entitlement service, use the URL `http://`*`publicDNS`*`:8080/web/`

## Error: Web Page Not Available

Check that your firewall settings allow connections using port 8080.

If your firewall settings are correct, open the Amazon EC2 console at https://console.aws.amazon.com/ec2/ and select your entitlement service. Verify that the **Instance State** of your entitlement service is **running**.

If the instance state is correct, verify that the UDP ports in your network are not blocked. If the UDP ports are blocked, unblock the ports or use a network with open UDP ports.

## Error: 503

Make sure the group or user that deployed the sample entitlement service has the correct permissions. The permissions are in Step 2: Create a Custom Policy with the Required Permissions (p. 201).

If the group has the correct permissions, the user that deployed the sample entitlement service is a member of the group.

Make sure you used the correct key pair name, access key, and secret access key when you deployed the sample entitlement service.

# Streaming Problems

Streaming problems can occur after you successfully deployed your streaming application. If you have not deployed your streaming application, see Deployment Problems (p. 208). If you encounter a problem connecting a client application to your streaming application, try the following solutions.

You can also enable logging to get a log file with the events that have occurred while your streaming application was running on Amazon AppStream. The events may provide detailed information about your problem. Simply write to any log directory you have access to and configure Amazon AppStream to upload those logs to the specified Amazon S3 bucket.

To enable logging, see Enabling Logging on a Streaming Application (p. 66).

**Topics**


- Error: Connection Timed Out (p. 210)
- Error: No Instance Available (p. 210)
- Error: Session Closed (p. 211)
- Error: Session Requested But No Available Instance (p. 211)
- Error: WebGL Not Found when using the Chrome App (p. 212)
- Link to Streaming Application No Longer Works (p. 212)
- Link Sent to User Does Not Work (p. 212)
- Two mouse pointers are visible in the Chrome App, OS X client application, or Windows application (p. 212)


## Error: Connection Timed Out

Your firewall must allow traffic through TCP port 80. Open TCP port 80 and try connecting again.

## Error: No Instance Available

This error appears when the server hosting your streaming application is not running or the streaming application is not ready for connections from client applications. Amazon AppStream stops the server when the streaming application has not been used for more than 90 minutes. If **Active Sessions** in the application summary page shows that zero servers are standing by ready for connections, then the server has been stopped.


For more information about viewing the application summary page, see Viewing Application Summary (p. 58).

Wait at least 10 minutes after you see this error before connecting to your streaming application. Waiting at least 10 minutes allows Amazon AppStream to start a new server with your streaming application. Your server is ready for connections to the client if **Active Sessions** in the application summary page has at least one server standing by ready for a connection.


# Error: Session Closed

Your firewall must allow traffic through UDP ports 9070 to 9097. Open UDP ports 9070 to 9097 and try connecting again.

# Error: Session Requested But No Available Instance

Your streaming application can only stream to a limited number of client applications. You can wait until a connection is available before connecting again or you can request that your streaming application accept more connections.

For more information about allowing more connections to your streaming application, see Increasing Your Amazon AppStream Service Limits (p. 71).

## Error: WebGL Not Found when using the Chrome App

See WebGL and 3D graphics in the Google Chrome Help Center for more information.

## Link to Streaming Application No Longer Works

You can use the link from **Application Summary** only once. In addition, the link is valid for 72 hours only. If you used the link already or you clicked on the link 72 hours after deploying your application, the link is not valid. To connect to your streaming application, you must create a new link. For more information, see Generating a New Test Connect Link to Stream Your Application (p. 56).

## Link Sent to User Does Not Work

The link from **Application Summary** can be used only once and is valid for 72 hours. To allow other users to connect to your streaming application, you have to create a new link for each user. For more information, see Generating a New Test Connect Link to Stream Your Application (p. 56).

## Two mouse pointers are visible in the Chrome App, OS X client application, or Windows application

Your streaming application was deployed with the `VDI_CAPTUREMOUSECURSOR` environment variable set to true. To show a single mouse pointer in the Chrome App, choose **Local only** in **Mouse cursor** of **Settings**. For the OS X and Windows client applications, redeploy the streaming application using the default settings.

For more information, see Displaying Mouse Pointers (p. 53).

# Error Codes

The following are the error messages you may encounter using Amazon AppStream. You can see these error messages through application:errors (p. 237) and Application Error (p. 229).

**Topics**


- APPLICATION_DELETION_FAILED (p. 212)
- APPLICATION_INSTALLATION_FAILED (p. 213)
- APPLICATION_LAUNCH_FAILED (p. 213)
- APPLICATION_INSTALLATION_TIMED_OUT (p. 213)
- APPLICATION_LAUNCH_TIMED_OUT (p. 213)
- APPLICATION_RUNTIME_FAILURE (p. 214)
- INTERNAL_FAILURE (p. 214)
- SDK_VERSION_DETECTION_FAILED (p. 214)
- S3_URL_INVALID (p. 214)


## APPLICATION_DELETION_FAILED

This error occurs when Amazon AppStream attempts to delete your application.

## Cause

Amazon AppStream generated an unknown error while deleting an application.

## Solution

Contact Amazon AppStream support.

# APPLICATION_INSTALLATION_FAILED

This error occurs when Amazon AppStream attempts to install your application.

## Cause

Amazon AppStream generated an unknown error while installing your application.

## Solution

Ensure that your application installer installs your application in a reasonable amount of time. If the application installer works, then contact Amazon AppStream support.

# APPLICATION_LAUNCH_FAILED

This error occurs when Amazon AppStream starts your application.

## Cause

Amazon AppStream generated an unknown error while starting your application.

## Solution

Contact Amazon AppStream support.

# APPLICATION_INSTALLATION_TIMED_OUT

This error occurs when Amazon AppStream runs the application installer.

## Cause

Amazon AppStream timed out waiting for the application installer to finish installing the application.

## Solution

Revise the application installer to complete the installation in a reasonable amount of time. Deploy the updated application installer.

# APPLICATION_LAUNCH_TIMED_OUT

This error occurs when Amazon AppStream attempted to start your application.

## Cause

Amazon AppStream timed out while starting your application.

## Solution

Verify that your application installer installs the files and sets the correct permission.

# APPLICATION_RUNTIME_FAILURE

This error occurs when Amazon AppStream runs your application.

## Cause

Your application has stopped.

## Solution

Contact Amazon AppStream support.

# INTERNAL_FAILURE

This error occurs in Amazon AppStream.

## Cause

An unknown error occurred in Amazon AppStream.

## Solution

Contact Amazon AppStream support.

# SDK_VERSION_DETECTION_FAILED

This error occurs when Amazon AppStream runs the application installer.

## Cause

Your application does not integrate with the libraries in the Amazon AppStream SDK.

## Solution

Revise the application code to integrate the libraries in the Amazon AppStream SDK. Deploy the updated application.

# S3_URL_INVALID

This error occurs when Amazon AppStream runs the application installer.

## Cause

- The specified presigned URL to the Amazon S3 bucket with your application installer is incorrect.

- The application installer in the Amazon S3 bucket is not set to public.
- The expiry time you set when creating the presigned URL is inadequate.
- You used the HTTP protocol to create the presigned URL.

## Solution

- Use the correct presigned Amazon S3 URL.
- Change the application installer in the Amazon S3 bucket to public. In the Amazon S3 bucket, select your application installer, click **Actions**, and then click **Make Public**.
- Create a presigned URL with a longer expiry time.
- Create a presigned URL with the HTTPs protocol.

# API Reference

**Topics**


- Amazon AppStream REST API (p. 216)


# Amazon AppStream REST API

The Amazon AppStream web service provides APIs you can call to manage applications hosted on Amazon AppStream and to manage client sessions connecting to those applications.

You can use this API to:

- Programmatically manage applications hosted on Amazon AppStream. For example, you can add new applications to Amazon AppStream and automate tasks you would otherwise perform through the console.
- Create tools or services to control access to your applications. The example found in Building an Entitlement Service (p. 193) demonstrates one such service.

You can call an API provided by Amazon AppStream by either submitting a REST request, or by calling wrapper functions in the Amazon AppStream SDK. You can download the Amazon AppStream SDK from the links in Downloads (p. 11).

**Topics**


- Hypertext Application Language (p. 216)
- Making HTTP Requests to Amazon AppStream (p. 218)
- Signing Requests (p. 221)
- Handling Errors in Amazon AppStream (p. 222)
- Resources (p. 225)
- Link Relations (p. 233)


## Hypertext Application Language

The Amazon AppStream web service is a resource-based API that uses Hypertext Application Language (HAL). HAL provides a standard way for expressing the resources and relationships of an API as hyperlinks. Using HAL, you use HTTP methods (GET, PUT, POST, DELETE) to submit requests and receive

information about the API in the response. Applications can use the information returned to explore the functionality of the API.

For example, you can inspect the JSON returned in the response from the web service on an Application (p. 226) resource to discover the session:entitle (p. 238) link. By extracting the `href` property of that link, you can programmatically build the request needed to create a new client session for that application. For more information about HAL, see the JSON Hypertext Application Language draft.

# Making HTTP Requests to Amazon AppStream

Amazon AppStream REST requests are HTTP requests as defined in RFC 2616. (For more information, go to http://www.ietf.org/rfc/rfc2616.txt.) This section describes the structure of an Amazon AppStream REST request. For detailed descriptions of the resources and references of the API, see Resources (p. 225).

A typical REST action consists of sending an HTTP request to Amazon AppStream and waiting for the response. Like any HTTP request, a REST request to Amazon AppStream contains a request method, a URI, request headers, and sometimes a query string or request body. The response contains an HTTP status code, response headers, and sometimes a response body.

**Topics**


- Limits on Request Rates (p. 218)
- HTTP Header Contents (p. 218)
- HTTP Request Body (p. 219)
- HTTP Responses (p. 219)


## Limits on Request Rates

Amazon AppStream limits the rate at which you can submit requests:

- You can submit a maximum of two POSTs to the applications resource per second per AWS account.
- You can submit a maximum of four POSTs to an application's sessions resource per second per AWS account.

If you submit requests at a faster rate, Amazon AppStream may respond with HTTP 429 errors as explained in API Error Codes (Client and Server Errors) (p. 222).

## HTTP Header Contents

Amazon AppStream requires the following information in the headers of an HTTP request:

**Host (Required)**
The Amazon AppStream endpoint. This value must be `https://appstream.us-east-1.amazonaws.com`.

**x-amz-date or Date (Required)**
The date used to create the signature contained in the `Authorization` header. Specify the date in ISO 8601 standard format, in UTC time, as in the following example: `X-Amz-Date: 20130613T203622Z`.

You must include either `x-amz-date` or `Date`. (Some HTTP client libraries don't let you set the `Date` header). When an `x-amz-date` header is present, the system ignores any `Date` header when authenticating the request.

The time stamp must be within 15 minutes of the AWS system time when the request is received. If it isn't, the request fails with the `RequestExpired` error code to prevent someone else from replaying your requests.

**Authorization (Required)**
The information required for request authentication. For more information about constructing this header, see Signing Requests (p. 221).

**Content-Type (Conditional)**
Specifies JSON and the version, for example, `Content-Type: application/x-amz-json-1.0`.

Condition: Required for PUT and POST requests.

**Content-Length (Conditional)**
Length of the message (without the headers) according to RFC 2616.

Condition: Required if the request body itself contains information (most toolkits add this header automatically).

The following are example headers for an HTTP request to entitle a new client session.

```
POST /applications/e407fbc8-5c27-48e6-9412-a876167546e8/sessions HTTP/1.1
host: appstream.us-east-1.amazonaws.com
x-amz-date: 20120116T174952Z
Authorization: AWS4-HMAC-SHA256 Credential=AccessKeyID/20120116/us-east-
1/ets/aws4_request,SignedHeaders=host;x-amz-date;x-amz-target,Signa
ture=145b1567ab3c50d929412f28f52c45dbf1e63ec5c66023d232a539a4afd11fd9
content-type: application/x-amz-json-1.0
content-length: 56
{
    "opaqueData": "TYtYS00MaWQ9Y2JmOmM4hhZWNS00MjJ"
}
```

# HTTP Request Body

Many Amazon AppStream API actions require you to include JSON-formatted data in the body of the request. The JSON conforms to the Amazon AppStream schema.

**Note**
JSON values in the request body are strings.

# HTTP Responses

All Amazon AppStream API actions include JSON-formatted data in the response. The JSON conforms to the Amazon AppStream schema.

**Note**
JSON values in the response are strings.

Here are some important headers in the HTTP response and how you should handle them in your application, if applicable:

**HTTP/1.1**
This header is followed by a status code. Status code `200` indicates a successful operation. For information about error codes, see API Error Codes (Client and Server Errors) (p. 222).

Type: String

**x-amzn-RequestId**
A value created by Amazon AppStream that uniquely identifies your request, for example, K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG. If you have a problem with Amazon AppStream, AWS can use this value to troubleshoot the problem. We recommend that you log these values.

Type: String

**Content-Length**
The length of the response body in bytes.

Type: String

**Content-Type**
The type of the message's content. Usually `application/hal+json`.

Type: String

**Date**

The date and time that Amazon AppStream responded, for example, `Sun, 25 Mar 2012 12:00:00 GMT`. The format of the date must be one of the full date formats specified by RFC 2616, section 3.3.

Type: String

# Signing Requests

Amazon AppStream requires that you authenticate every request you send by signing the request. To sign a request, you calculate a digital signature using a cryptographic hash function, which returns a hash value based on the input. The input includes the text of your request and your secret access key. The hash function returns a hash value that you include in the request as your signature. The signature is part of the `Authorization` header of your request.

After receiving your request, Amazon AppStream recalculates the signature using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, Amazon AppStream processes the request. Otherwise, the request is rejected.

Amazon AppStream supports authentication using AWS Signature Version 4. The process for calculating a signature can be broken into three tasks:

- Task 1: Create a Canonical Request

  Create your HTTP request in canonical format as described in Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.
- Task 2: Create a String to Sign

  Create a string that you will use as one of the input values to your cryptographic hash function. The string, called the *string to sign*, is a concatenation of the name of the hash algorithm, the request date, a *credential scope* string, and the canonical request from the previous task. The *credential scope* string itself is a concatenation of date, region, and service information.

  For the `X-Amz-Credential` parameter, specify:
  - The code for the endpoint to which you're sending the request, for example, `us-east-1`. For a list of regions and endpoints for Amazon AppStream, see the Regions and Endpoints chapter of the *Amazon Web Services General Reference*. When specifying the code for the endpoint, include only the part between `appstream.` and `.amazonaws.com`
  - `appstream` for the service abbreviation

  For example:

  ```
  X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20130501/us-east-1/appstream/aws4_request
  ```
- Task 3: Create a Signature

  Create a signature for your request by using a cryptographic hash function that accepts two input strings: your *string to sign* and a *derived key*. The *derived key* is calculated by starting with your secret access key and using the *credential scope* string to create a series of hash-based message authentication codes (HMACs).

# Handling Errors in Amazon AppStream

**Topics**


- API Error Codes (Client and Server Errors) (p. 222)
- Catching Errors (p. 224)
- Error Retries and Exponential Backoff (p. 224)


When you send requests to and get responses from the Amazon AppStream API, you might encounter two types of API errors:

- **Client errors:** Client errors are indicated by a 4xx HTTP response code. Client errors indicate that Amazon AppStream found a problem with the client request, such as an authentication failure or missing required parameters. Fix the issue in the client application before submitting the request again.
- **Server errors:** Server errors are indicated by a 5xx HTTP response code, and need to be resolved by Amazon. You can resubmit/retry the request until it succeeds.

For each API error, Amazon AppStream returns the following values:

- A status code, for example, `400`
- An error code, for example, `ValidationException`
- An error message, for example, `Supplied AttributeValue is empty, must contain exactly one of the supported datatypes`

For a list of error codes that Amazon AppStream returns for client and server errors, see API Error Codes (Client and Server Errors) (p. 222).

## API Error Codes (Client and Server Errors)

HTTP status codes indicate whether an operation is successful or not.

A response code of `2xx` indicates the operation was successful. Other error codes indicate either a client error (4xx) or a server error (5xx).

The following table lists the errors returned by Amazon AppStream. Some errors are resolved if you simply retry the same request. The table indicates which errors are likely to be resolved with successive retries. If the value of the Retry column is:

- **Yes:** Submit the same request again.
- **No:** Fix the problem on the client side before submitting a new request.

For more information about retrying requests, see Error Retries and Exponential Backoff (p. 224).

| HTTP Status Code | Error code | Message | Cause | Retry |
|---|---|---|---|---|
| 400 | Conditional Check Failed Exception | The conditional request failed. | Example: The expected value did not match what was stored in the system. | No |

| HTTP Status Code | Error code | Message | Cause | Retry |
|---|---|---|---|---|
| 400 | Incomplete Signature Exception | The request signature does not conform to AWS standards. | The signature in the request did not include all of the required components. See HTTP Header Contents (p. 218). | No |
| 400 | Missing Authentication Token Exception | The request must contain a valid (registered) AWS Access Key ID. | The request did not include the required `x-amz-security-token`. See Making HTTP Requests to Amazon AppStream (p. 218). | No |
| 400 | Validation Exception | Various. | One or more values in a request were missing or invalid; for example, a value was empty or was greater than the maximum valid value. | No |
| 403 | AccessDenied Exception | • Deleting a system preset is not allowed: account=<accountId>, presetId=<presetId>.<br>• General authentication failure. The client did not correctly sign the request. See Signing Requests (p. 221). | You attempted to delete a system preset, the signature in a call to the Amazon AppStream API was invalid, or the IAM user whose credentials were used for this request is not authorized to perform the operation. | No |
| 404 | ResourceNot Found Exception | • The specified <resource> could not be found: <resourceId>. | Example: The application to which you're trying to create a new session doesn't exist or is still being created. | No |
| 409 | Resource InUse Exception | • The <resource> was already in use: accountId=<accountId>, resourceId=<resourceId>. | Example: You attempted to delete an application that is currently in use. | No |
| 429 | Limit Exceeded Exception | • The account already has the maximum number of sessions allowed: account=<accountId>, maximum number of sessions=<maximum> | The current AWS account has exceeded limits on Amazon AppStream objects. . | |
| 429 | Provisioned Throughput Exceeded Exception | You exceeded your maximum allowed provisioned throughput. | Example: Your request rate is too high. The AWS SDKs for Amazon AppStream automatically retry requests that receive this exception. Your request is eventually successful unless your retry queue is too large to finish. Reduce the frequency of requests. For more information, see Error Retries and Exponential Backoff (p. 224). | Yes |

| HTTP Status Code | Error code | Message | Cause | Retry |
|---|---|---|---|---|
| 429 | Throttling Exception | Rate of requests exceeds the allowed throughput. | You are submitting requests too rapidly; for example, requests to entitle new sessions. | Yes |
| 500 | Internal Failure | The server encountered an internal error trying to fulfill the request. | The server encountered an error while processing your request. | Yes |
| 500 | Internal Server Error | The server encountered an internal error trying to fulfill the request. | The server encountered an error while processing your request. | Yes |
| 500 | Internal Service Exception | | The service encountered an unexpected exception while trying to fulfill the request. | Yes |
| 500 | Service Unavailable Exception | The service is currently unavailable or busy. | There was an unexpected error on the server while processing your request. | Yes |

### Sample Error Response

The following is an HTTP response indicating that the value for `inputBucket` was null, which is not a valid value.

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: b0e91dc8-3807-11e2-83c6-5912bf8ad066
x-amzn-ErrorType: ValidationException
Content-Type: application/json
Content-Length: 124
Date: Mon, 26 Nov 2012 20:27:25 GMT

{"message":"1 validation error detected: Value null at 'InstallS3Bucket' failed
 to satisfy constraint: Member must not be null"}
```

## Catching Errors

For your application to run smoothly, you need to build in logic to catch and respond to errors. One typical approach is to implement your request within a `try` block or `if-then` statement.

The AWS SDKs perform their own retries and error checking. If you encounter an error while using one of the AWS SDKs, you should see the error code and description. You should also see a `Request ID` value. The `Request ID` value can help troubleshoot problems with Amazon AppStream support.

## Error Retries and Exponential Backoff

Numerous components on a network, such as DNS servers, switches, load balancers, and others can generate errors anywhere in the life of a given request.

The usual technique for dealing with these error responses in a networked environment is to implement retries in the client application. This technique increases the reliability of the application and reduces operational costs for the developer.

Each AWS SDK supporting Amazon AppStream implements automatic retry logic. The AWS SDK for Java automatically retries requests, and you can configure the retry settings using the `ClientConfiguration` class. For example, in some cases, such as a web page making a request with minimal latency and no retries, you might want to turn off the retry logic. Use the `ClientConfiguration` class and provide a `maxErrorRetry` value of `0` to turn off the retries.

If you're not using an AWS SDK, you should retry original requests that receive server errors (5xx). However, client errors (4xx, other than a `ThrottlingException` or a `ProvisionedThroughputExceededException`) indicate you need to revise the request itself to correct the problem before trying again.

In addition to simple retries, we recommend using an exponential backoff algorithm for better flow control. The idea behind exponential backoff is to use progressively longer waits between retries for consecutive error responses. For example, you might let one second elapse before the first retry, four seconds before the second retry, 16 seconds before the third retry, and so on. However, if the request has not succeeded after a minute, the problem might be a hard limit and not the request rate. For example, you may have reached the maximum number of pipelines allowed. Set the maximum number of retries to stop around one minute.

# Resources

The Amazon AppStream API includes the following resources.

**Topics**


- AppStream (p. 225)
- Applications (p. 226)
- Application (p. 226)
- Application Errors (p. 229)
- Application Error (p. 229)
- Application Status (p. 230)
- Sessions (p. 231)
- Session (p. 231)
- Session Status (p. 232)


# AppStream

The root of the Amazon AppStream service.

## Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The root resource of Amazon AppStream. | GET | No. |
| appstream:applications | A link to the applications resource for the service. | GET | No. |

## Properties

The AppStream resource has no properties.

# Applications

The Applications resource is a collection resource that contains zero or more references to your existing applications, and links that guide you on ways to interact with your collection. The collection offers a paginated view of the contained applications.

## Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The collection of applications you have hosted on Amazon AppStream. | GET | No. |
| application:by-id (p. 234) | Retrieves an individual Application resource based on the specified identifier. | GET | Yes. Requires the application identifier. |
| application:create (p. 234) | Adds a new application to Amazon AppStream. You must have previously uploaded the application package to Amazon S3 and created a presigned URL. For more information about generating presigned URLs, see Share an Object with Others in the *Amazon Simple Storage Service Developer Guide*. | POST | No. |
| item (p. 239) | An array of links to the current page of Application resources. | GET | No. |
| next (p. 239) | The next page of items in a collection. If there are no further pages of items, this link is not returned in the response. | GET | No. |
| first (p. 239) | The first page of items in a collection. This link is returned only when on pages other than the first one. | GET | No. |

## Properties

The Applications resource has no properties.

# Application

An application you have added to Amazon AppStream. You are only able to access applications added to your AWS account.

## Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | An application hosted on Amazon AppStream. | GET | No. |
| application:status | The current status of this application. | GET | No. |
| application:update (p. 236) | Update selected metadata for this application. | POST | No. |
| application:errors | The collection of errors for this application. | GET | No. |
| application:archive | Archives the application. | PUT | No. |
| application:reactivate | Activates a previously archived application. | PUT | No. |
| application:delete | Deletes an application that is in the `Error` state. | DELETE | No. |
| application:sessions | The collection of sessions associated with this application. | GET | No. |
| session:by-id (p. 237) | The session which has the specified identifier. | GET | Yes. Requires the session identifier. |
| session:entitle (p. 238) | Create a new client session for this application. | POST | No. |
| collection (p. 239) | The collection of applications you have hosted on Amazon AppStream. The collection includes this application. | GET | No. |

## Properties

The following properties of the application are set when you create the application, either by using the REST API or the Amazon AppStream console.

| Name | Description |
|---|---|
| id | The application identifier. This is unique across all your applications in Amazon AppStream. |
| name | The name of the application. |
| description | (Optional) The description of the application. This description is used to describe the application in Amazon AppStream, it is not displayed to end users. |

| Name | Description |
| --- | --- |
| installerUrl | A presigned URL that points to the location in Amazon S3 that contains the installation package for the application. For more information on creating a presigned URL, see Share an Object with Others in the *Amazon Simple Storage Service Developer Guide.* |
| installerParameters | The parameters required by the application installer to install that application on a host managed by Amazon AppStream. |
| launchCommand | The command to run in order to launch the application after it is hosted on Amazon AppStream.<br>**Important**<br>This command should not should include any command-line parameters. |
| logBucket | The name of the Amazon S3 bucket where the `.zip` files contains the logs are uploaded. The bucket must already exist and allow Amazon AppStream to use the `PutObject` method. |
| logPaths | A string array whose elements contain the log filenames or directory path where logs are stored. The filenames and paths must meet the following requirements:<br>• Log files must be located in the `C:\` directory.<br>• Log paths must be fully qualified paths. Do not use relative paths. You can use wildcards.<br>• Log paths must be less than the path length limits. See Naming Files, Paths, and Namespaces.<br>• Log file names cannot contain Windows reserved characters. See Naming Conventions.<br>• Your application must have read access to the paths.<br>**Note**<br>The total size of logs in the `.zip` file is subject to the Amazon S3 key size limits. |
| applicationErrorCount | The total number of errors associated with this application. This does not include errors associated with client sessions, which are enumerated in sessionErrorCount. |
| activeSessions | The total number of client sessions currently connected to this application. |
| availableSessions | The additional capacity available to accept client sessions. |
| sessionErrorCount | The total number of errors associated with client sessions connected to this application. |
| createdDate | The date the application was created, in ISO 8601 format. |
| lastUpdatedDate | The date the application metadata was last updated in Amazon AppStream, in ISO 8601 format. |

# Application Errors

The collection of Application Error resources associated with an application hosted on Amazon AppStream.

## Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The collection of errors associated with an application. | GET | No. |
| item (p. 239) | An array of links to the current page of Application Error resources. | GET | No. |
| next (p. 239) | The next page of items in a collection. If there are no further pages of items, this link is not returned in the response. | GET | No. |
| first (p. 239) | The first page of items in a collection. Only available if on a page other than the first. | GET | No. |

## Properties

The Application Errors resource does not have any properties.

# Application Error

An error associated with the current application.

## Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The error state. | GET | No. |
| collection (p. 239) | The collection of errors associated with the current application. The collection includes this error. | GET | No. |

## Properties

| Name | Description |
|---|---|
| id | The identifier of the error. For more information about the errors returned by the Amazon AppStream service, see Handling Errors in Amazon AppStream (p. 222). |

| Name | Description |
| --- | --- |
| state | The current state of the error. This can be one of the following: `New`, `Read`, or `Deleted`. |
| type | The type of error. This can either be `Application` or `Session`. |
| message | A message that describes the error. |
| errorDate | The date the error occurred, in ISO 8601 format. |

# Application Status

The current status of an application hosted on Amazon AppStream.

## Links

| Relation | Description | Methods | Templated |
| --- | --- | --- | --- |
| self | The status of an application. | GET | No. |
| up (p. 239) | The application to which this status applies. | GET | No. |

## Properties

| Name | Description |
| --- | --- |
| state | The state of the current application. This can be one of the following values:<br>• `Active`—The application is ready to accept client sessions.<br>• `Archived`—The application must be reactivated before accepting client sessions.<br>• `Archiving`—The application is in the process of archiving. No new sessions can be entitled, but existing sessions will continue until these sessions finish. To end the existing sessions, explicitly terminate the existing session.<br>• `Blocked`—The application is waiting on dependencies.<br>• `Building`—Amazon AppStream is allocating resources to host the application.<br>• `Deleting`—The application is in the process of being deleted.<br>• `Error`—The application failed to build or deploy properly.<br>• `New`—The application has just been created.<br>• `Unknown`—The application state cannot be determined. |

| Name | Description |
| --- | --- |
| buildStep | This property is present when the state is either `Building` or `Error`. It indicates either the currently processing step if the state is `Building`, or the step that was processing if the state is `Error`. It can be one of the following values:<br>• `Copying`—Copying the installer data into the hosting environment.<br>• `Installing`—Installing the application into the hosting environment.<br>• `Preparing`—Preparing the host environment for application installation.<br>• `Provisioning`—Provisioning the hosting environment to run your application. |

# Sessions

The list of sessions associated with the current application. You can only access sessions for applications associated with your AWS account.

## Links

| Relation | Description | Methods | Templated |
| --- | --- | --- | --- |
| self | The collection of sessions associated with the current application. | GET | No. |
| session:by-id (p. 237) | The session with the specified identifier. | GET | Yes. Requires the session identifier. |
| item (p. 239) | An array of links to the current page of Session resources. | GET | No. |

## Properties

The sessions resource has no properties.

# Session

A resource representing an individual client session of an application hosted on Amazon AppStream.

## Links

| Relation | Description | Methods | Templated |
| --- | --- | --- | --- |
| self | A session of the specified application | GET | No. |

| Relation | Description | Methods | Templated |
|---|---|---|---|
| session:status (p. 238) | The status of the current session. | GET | No. |
| session:terminate (p. 238) | Terminate the current session. | PUT | No. |
| collection (p. 239) | The list of all client sessions for the specified application. | GET | No. |

### Properties

| Name | Description |
|---|---|
| id | Unique identifier for the session. |
| entitlementUrl | The URL clients use to redeem an entitlement for this session and connect to the application. |
| opaqueData | Data to pass to the application. This data is not used by the client, entitlement service, or Amazon AppStream, it is used by the application. An example of opaqueData would be a user identifier, which the application would then use to load previous state information (such as high score or current map level) for that user from a database. |
| errorCount | The number of errors associated with this session. |
| startDate | The time at which the session began, in ISO 8601 format. |
| endDate | The time at which the session ended, in ISO 8601 format. If this is null, the session is active and the client is currently connected. |

## Session Status

Returns the status of the current session.

### Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The status of the current session. | GET | No. |
| up (p. 239) | The session to which this state applies. | GET | No. |

### Properties

| Name | Description |
| --- | --- |
| state | Current state of the session. This can be one of the following values:<br>• `Unknown`—the session state cannot be determined.<br>• `Entitled`—the session has been entitled, but the client has not yet redeemed the entitlement.<br>• `Reserved`—the server side of the session is ready to receive the client, but the client has not yet connected.<br>• `Active`—the session is actively streaming the application to a client.<br>• `Completed`—the session ended by either client or server action.<br>• `Terminating`—the session was terminated by the developer and is in the process of ending.<br>• `Terminated`—the session was terminated by the developer. |

# Link Relations

The Amazon AppStream API provides the following link relations that you can use to access and modify Amazon AppStream resources.

**Topics**


- appstream:applications (p. 233)
- application:by-id (p. 234)
- application:create (p. 234)
- application:update (p. 236)
- application:archive (p. 236)
- application:reactivate (p. 237)
- application:delete (p. 237)
- application:status (p. 237)
- application:errors (p. 237)
- application:sessions (p. 237)
- session:by-id (p. 237)
- session:entitle (p. 238)
- session:status (p. 238)
- session:terminate (p. 238)
- Common Link Relations (p. 239)


## appstream:applications

You can get the Applications resource for all your applications by performing a GET on the href of this link.

### Output

Returns the Applications resource that represents the collection of your Application resources.

## application:by-id

You can use a Get request and this API to retrieve the application associated with the specified identifier.

### Output

Returns the Application associated with the specified identifier.

## application:create

You can use a POST request on the href of this link to add an application to Amazon AppStream. You must have previously uploaded your application's installation package to Amazon S3 and generated a presigned URL for the package's location in Amazon S3. For more information about generating presigned URLs, see Share an Object with Others in the *Amazon Simple Storage Service Developer Guide*.

### Input

In order to add an application to Amazon AppStream, you must pass in the following fields during your POST request.

| Input Field | Description |
|---|---|
| name | The name of the application. |
| description | (Optional) The description of the application. This description is used to describe the application in Amazon AppStream, it is not displayed to end users. |
| installerUrl | A presigned URL that points to the location in Amazon S3 that contains the installation package for the application. For more information on creating a presigned URL, see Share an Object with Others in the *Amazon Simple Storage Service Developer Guide.* |
| installerParameters | The command to run to install the application on an Amazon AppStream host managed by Amazon AppStream. This should include all necessary command-line parameters. |
| launchCommand | The command to run in order to launch the application after it is hosted on Amazon AppStream. This should include all necessary command-line parameters. |
| logBucket | The name of the Amazon S3 bucket where the `.zip` files contains the logs are uploaded. The bucket must already exist and allow Amazon AppStream to use the `PutObject` method. |

| Input Field | Description |
| --- | --- |
| logPaths | A string array whose elements contain the log filenames or directory path where logs are stored. The filenames and paths must meet the following requirements:<br>• Log files must be located in the `C:\` directory.<br>• Log paths must be fully qualified paths. Do not use relative paths. You can use wildcards.<br>• Log paths must be less than the path length limits. See Naming Files, Paths, and Namespaces.<br>• Log file names cannot contain Windows reserved characters. See Naming Conventions.<br>• Your application must have read access to the paths.<br>**Note**<br>The total size of logs in the `.zip` file is subject to the Amazon S3 key size limits. |

## Output

The newly created Application.

### Example Request

```
POST /applications HTTP/1.1
Host: appstream.us-east-1.amazonaws.com
Accept: application/hal+json
Version=4
X-Amz-Algorithm=AWS4-HMAC-SHA256
X-Amz-Credential=...%2Fus-east-1%2Fappstream%2Faws4_request
X-Amz-Date=2013-11-06T19%3A18%3A42.323Z
X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
X-Amz-Signature=...
{
  "name": "SampleApp",
  "description": "A sample application hosted on Amazon AppStream.",
  "installParameters": " ",
  "installerUrl": "S3_URL",
  "launchCommand": "c:\\app\\SampleApp\\sampleapp.exe"
  "logBucket": "MyS3Bucket",
  "logPaths": [""C:\MyLog1.log","C:\MyLog2.log",C:\myapp\logs\*logs]
}
```

### Example Response

```
HTTP/1.1 200 OK
Content-Type: application/hal+json
{

}
```

# application:update

You can use a POST request on the href of this link to update the metadata fields associated with an application hosted on Amazon AppStream. Updating the state affects the active applications. Updating the state does not affect the active sessions.

## Input

In order to update the state of an application, you must pass in the following field during your POST request.

| Input Field | Description |
|---|---|
| name | The new name of the application. |
| description | The new description of the application. This is used internally by Amazon AppStream and not displayed to end users. |
| launchCommand | The new command to launch the application. This command is run on the streaming server after it finishes allocating. The command starts the application on the server. |
| logBucket | The name of the Amazon S3 bucket where the `.zip` files contains the logs are uploaded. The bucket must already exist and allow Amazon AppStream to use the `PutObject` method. |
| logPaths | A string array whose elements contain the log filenames or directory path where logs are stored. The filenames and paths must meet the following requirements:<br>• Log files must be located in the `C:\` directory.<br>• Log paths must be fully qualified paths. Do not use relative paths. You can use wildcards.<br>• Log paths must be less than the path length limits. See Naming Files, Paths, and Namespaces.<br>• Log file names cannot contain Windows reserved characters. See Naming Conventions.<br>• Your application must have read access to the paths.<br>**Note**<br>The total size of logs in the `.zip` file is subject to the Amazon S3 key size limits. |

## Output

The updated Application.

# application:archive

You can use a PUT request and this API to archive an application and release the AWS resources allocated to host it. Any sessions that are currently active on the application will continue, but the application does not accept new sessions. When all the currently active sessions have concluded, the application reaches the `Archived` state.

An application that is archived can later be restarted with application:reactivate.

### Output

The updated Application Status. This resource will initially have a state of `Archiving`, but the state will eventually be `Archived`. You can periodically poll this resource to see its progress.

## application:reactivate

You can use a PUT request and this API to restore a previously archived application and make it ready to accept client sessions.

### Output

The updated Application Status. This resource will initially have a state of `Building`, but the state will eventually be `Active`. You can periodically poll this resource to see its progress.

## application:delete

To delete an Application that is in the `Error` state, call the href in the link using the DELETE method.

### Output

Returns the Applications resource.

## application:status

You can use a GET request on the href of this link to retrieve the Application Status associated with the current Application.

### Output

The Application Status for this Application.

## application:errors

You can use a GET request on the href of this link to retrieve the Application Errors associated with the current Application.

### Output

The Application Errors for this Application.

## application:sessions

You can GET the href of this link to retrieve the Sessions resource for associated with the current application.

### Output

The associated Sessions resource.

## session:by-id

You can use a GET request with this API to retrieve the Session associated with the specified identifier.

### Output

The Session associated with the specified identifier.

## session:entitle

You can use a POST request and this API to entitle a new session for the current application. Amazon AppStream returns the identifier of the newly created session. The new session contains the entitlementUrl property which your entitlement service should pass back to the client. The client will then use this URL to redeem its entitlement from Amazon AppStream and get the IP address it will use to connect to the application.

### Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The session:entitle link relation. | POST | No. |

### Input

When you entitle a new session, you can pass in the following field during your POST request.

| Input Field | Description |
|---|---|
| opaqueData | (Optional) Data to pass to the application. This data is not used by the client, entitlement service, or Amazon AppStream, it is used by the application. An example of opaqueData would be a user identifier, which the application would then use to load previous state information (such as high score or current map level) for that user from a database. |

### Output

The newly entitled Session (p. 231).

## session:status

You can use a GET request on the href of this link to retrieve the Session Status associated with the current Session.

### Output

The Session Status for this Session.

## session:terminate

You can use a PUT request and this API to terminate a session entitlement and return the session to the pool of available sessions. Any client connected to the session will be disconnected.

### Links

| Relation | Description | Methods | Templated |
|---|---|---|---|
| self | The session:terminate link relation. | GET | Yes. Requires the session identifier. |

### Output

The updated session:status (p. 238). This should be set to `Terminating` or `Terminated`.

## Common Link Relations

The following link relations may be applied to Applications or Sessions depending on the context of the Amazon AppStream request. For more information about common link relations, see http://www.iana.org/assignments/link-relations/link-relations.xhtml.

**Topics**

- collection (p. 239)
- first (p. 239)
- item (p. 239)
- next (p. 239)
- up (p. 239)

### collection

The collection link relation identifies a target resource that represents a collection of which the context resource is a member. For example, an application resource may have a collection link relation that points to the collection of all applications associated with your AWS account.

### first

The first page of items in a collection. This can be a page of applications, sessions, or errors, depending on the context of the request.

### item

An item in a collection. This can be an application, session, or error, depending on the context of the request. For example, the Applications resource has an item relation that is an array of individual Application resources.

### next

The next page of items in a collection. This can be a page of applications or sessions, depending on the context of the request.

### up

The parent of the current context. For example, the up link relation of the Application Status resource is the Application to which the status applies.

# Product Updates

**Topics**


- Release for May 14, 2015 (Latest) (p. 240)
- Release for March 26, 2015 (p. 241)
- Release for February 2, 2015 (p. 241)
- Release for November 20, 2014 (p. 242)
- Release for September 30, 2014 (p. 242)
- Release for September 16, 2014 (p. 243)
- Release for August 22, 2014 (p. 243)
- Release for August 5, 2014 (p. 243)
- Release for June 20, 2014 (p. 244)
- Release for May 30, 2014 (p. 245)
- Release for May 9, 2014 (p. 245)
- Release for April 22, 2014 (p. 246)
- Release for March 28, 2014 (p. 246)
- Release for March 7, 2014 (p. 247)
- Release for February 14, 2014 (p. 248)
- Release for January 24, 2014 (p. 248)


## Release for May 14, 2015 (Latest)

This release includes the following changes:

### New client application for Android and Fire OS devices

You can use a device running Android or Fire OS to connect to a streaming application with the new Amazon AppStream Example Client. This client application enables an Android or Fire OS device to connect to a streaming application in test mode or through an entitlement service.

Learn more at Android and Fire OS (p. 22).

## New client application for iOS devices

You can use a device running iOS to connect to a streaming application with the new **Basic AppStream Client**. This client application enables an iOS device to connect to a streaming application in test mode or through an entitlement service.

Learn more at iPad, iPhone, or iPod Touch (p. 28).

## Connect to a streaming application in one click

You can connect to a streaming application in one click through the Chrome browser or devices that use Android, Fire OS, and iOS. To do this, use a URL in a deep link format.

Learn how to construct this URL at Streaming your Application in One Click (p. 16).

# Release for March 26, 2015

This release includes the following changes:

## Updates to the Amazon AppStream Chrome App SDK and Chrome App

The Amazon AppStream Chrome App SDK was updated to use the Chrome video decoder instead of FFmpeg video decoder. The Chrome App has a **Settings** dialog box that allows you to use the mouse input of the Chrome App, the streaming application or both the Chrome App and streaming application, show verbose debugging information to help you solve a problem, and change the inactivity time.

Download the latest Amazon AppStream Chrome App SDK from https://s3.amazonaws.com/appstream-sdk/AppStream_Chrome_SDK-1.2.1.189.zip. The latest Chrome App is available from the Google Play Store.

## Display the mouse pointers

Because your users interact with your streaming application using a client application and your streaming application runs on Amazon AppStream, you have two mouse pointers to consider. You can control which mouse pointers to display on the client applications.

Learn more at Displaying Mouse Pointers (p. 53).

# Release for February 2, 2015

This release includes the following changes:

## Connect to an entitlement service using a Chrome browser

You can connect to an entitlement service from a Chrome browser by using a special URL called a deep link. When you use this URL, the Chrome browser connects the Chrome app to your streaming application through an entitlement service.

Learn more at Streaming your Application in One Click (p. 16).

### New deployment wizards

You can deploy your streaming application by using the interactive or the advanced deployment wizards. The wizard you use depends on your browser and the type of application installer.

Learn more at Deploying Your Streaming Application to Amazon AppStream (p. 38)

## Release for November 20, 2014

This release includes the following changes:

### Revised deployment from the console

The Amazon AppStream console has been revised to simplify the deployment process. We also removed the requirements for an presigned URL to an S3 bucket and to use a silent installer to deploy your streaming application.

Learn more at Getting Started with Amazon AppStream (p. 13).

### Quick link to test your streaming application

After you deploy your streaming application, you can start streaming by using a quick link to connect a client application to your streaming application.

Learn more at Generating a New Test Connect Link to Stream Your Application (p. 56).

## Release for September 30, 2014

This release includes the following changes:

### Amazon AppStream now available in Asia Pacific (Tokyo) region

Developers who cater to users in Asia Pacific can now offer lower latencies to their users by streaming their applications from the AWS data center in Tokyo. A localized version of the Amazon AppStream Developer Guide is also available.

### Sample entitlement service template updated for use in multiple regions

The sample entitlement service has been updated to access applications from any region. You can download the latest sample entitlement service template from Downloads (p. 11).

# Release for September 16, 2014

This release includes the following change:

## Support for Chromebooks and Chrome browsers

The Amazon AppStream Chrome App SDK is now available to support streaming applications on Chromebooks and Chrome web browsers on Mac, Windows, and Linux desktops. An example Amazon AppStream Chrome App is available from the Chrome Web Store, and can be customized with HTML, JS, and CSS using the SDK available from Downloads (p. 11).

Learn more about this new client application at Building a Chrome App (p. 82).

# Release for August 22, 2014

This release includes the following changes:

## Simplified sample entitlement service deployment

Deploying the sample entitlement service is now simpler. Learn more about deploying the sample entitlement service at Amazon AppStream Sample Entitlement Service (p. 200).

## Password requirements for standalone mode

Standalone mode has new password requirements including specific requirements for the DeveloperPassword and the TightVNC passwords.

# Release for August 5, 2014

This release includes the following changes:

## Android client application updates

- The client application can connect to multiple streaming applications without restarting the client application. In the previous version, you had to restart the client application before connecting to another streaming application.
- The client application adjusts the volume on the device instead of sending a command to the streaming application to adjust the volume.
- We added support for joysticks.
- The client application displays error and connection information in a more unified manner with the other client applications.

## iOS client application updates

- The client application can connect to multiple streaming applications without restarting the client application. In the previous version, you had to restart the client application before connecting to another streaming application.

- The client application displays error and connection information in a more unified manner with the other client applications.

## OS X client application updates

- The client application can connect to multiple streaming applications without restarting the client application. In the previous version, you had to restart the client application before connecting to another streaming application.
- The client application displays error and connection information in a more unified manner with the other client applications.

## Windows client application updates

- We added support for the middle mouse button.

## Other Updates

- We fixed an issue that prevented the source code for the Windows example client application from being compiled.

# Release for June 20, 2014

This release includes the following changes:

## Up to 20% performance improvement on 64-bit iOS devices

We added support for ARM64 which improves the performance on 64-bit iOS devices by up to 20%. If you are updating an existing iOS client application, you should use the new `libXStxClient.a` file and update the *<SDK_dir>*`/3rdparty` folder to the version in the SDK and set Architectures in your project Build Settings to `$(ARCHS_STANDARD)` to include ARM64 support.

## Simplified sample streaming application code

The code for the sample streaming application has been simplified to under 600 lines, down from more than 2,000 lines. The SDK also contains a Visual Studio solution file so you just need Visual Studio 2010 or later to compile the code. In the previous SDK version, you needed both CMake for Windows and Visual Studio to compile the code.

## Remote debugging in standalone mode

You can debug your streaming application running in Amazon AppStream standalone mode by using Remote Tools for Visual Studio on your Amazon EC2 instance. Learn more about remote debugging in standalone mode at Debugging Your Streaming Application in Amazon AppStream Standalone Mode (p. 187).

## Other Updates

- We fixed an issue where an invalid path specified in the log page prevents the next step in the console even if logging was not specified.

# Release for May 30, 2014

This release includes the following changes:

## Utilization Log is in Coordinated Universal Time (UTC)

Amazon AppStream now uses UTC time instead of epoch time in the utilization log. The UTC time allows better coordination and consistency between other AWS services and Amazon AppStream.

## Utilization Log Contains GPU Metrics

You can now see GPU metrics in the utilization log when you select logging for your streaming application. The utilization log contains GPU metrics, such as memory information, power utilization, and temperature to help you monitor and diagnose your streaming application.

Learn more about the GPU metrics at Enabling Logging on a Streaming Application (p. 66).

## Simplified the Application Resource Properties

The applicationManifest property is no longer in the Application (p. 226) resource. The properties within applicationManifest property are now in the Application (p. 226) properties.

This change simplifies the Application (p. 226) resource structure.

## Other Updates

- The latency rate is now equal or better than the previous SDK release. Some customers experienced increased latency rates after using the previous SDK release.
- The example Android client application no longer hangs on exit.
- Bandwidth adaptation is improved when the frame rate varies.

# Release for May 9, 2014

This release includes the following changes:

## Logging for Utilization Metrics

You can now see utilization metrics in the logs when you select logging for your streaming application. This comma-separated value file contains operational metrics, such as CPU utilization, memory information, and disk read and write speeds.

Learn more about the utilization metrics at Enabling Logging on a Streaming Application (p. 66).

## YUV444 Color Subsampling Support

Amazon AppStream now supports streaming video to client applications and devices at the YUV444 color subsampling rate. If your streaming application, client application, and devices support YUV444, you can specify this higher rate in your streaming application.

Learn more about YUV444 at Streaming Video to a Client (p. 175).

## Other Updates

- Amazon AppStream now has less server overhead, giving more CPU resources to streaming applications and improving performance.
- The Amazon AppStream console **Summary** page now has a link to the Amazon S3 bucket that stores your log file.
- Amazon AppStream now creates a log file when the streaming application unexpectedly stops. In the previous version, a log file was not created after the streaming application stops.
- The Amazon AppStream console features improved text.

# Release for April 22, 2014

This release includes the following changes:

## Standard and User-defined Logging

You can configure your streaming application to save standard and user-defined logs to an Amazon S3 bucket. When the streaming application terminates, Amazon AppStream collects the logs into a `.zip` file and then uploads this file to your own bucket or to a bucket that Amazon AppStream creates.

Learn more about the logging feature at Enabling Logging on a Streaming Application (p. 66).

## Updated OpenSSL Version

We updated the OpenSSL cryptography library in Amazon AppStream to a version that fixes the Heartbleed security bug.

## Other Updates

- We now report an error if the application installer fails in the middle of the process. Previously, we did not report an error when this condition occurs.

# Release for March 28, 2014

This release includes the following changes:

## New Billing Practice

Your bill now includes session time up to the point when your streaming application unexpectedly terminates. Previously, your bill did not include any session in which your streaming application unexpectedly terminated.

## SDK Version Detected from the Files instead of the Registry

The version of the Amazon AppStream SDK that you use is based on the files rather than a registry setting. This change ensures that Amazon AppStream implements the correct version for your streaming application.

## Other Updates

- The **Cancel** button on the console now takes you to the **Welcome** page.

# Release for March 7, 2014

This release includes the following changes:

## Amazon AppStream Is Available to Anyone with an AWS Account

During Limited Preview, only a limited number of users had access to Amazon AppStream. Now allow anyone with an AWS account can try Amazon AppStream.

## Improved Deployment Error Messages

Error messages now display the reason why a deployment failed. The error messages are documented in Error Codes (p. 212).

## Example Client Application for Mac OS X

The Amazon AppStream SDK now includes a precompiled client application and the source code to build your own client application for Mac OS X.

Learn more about creating a sample application client in Building a Client for OS X (p. 96).

## Increase Your Service Limits

By default, you can have up to ten simultaneous sessions connect to your streaming application. When you want your streaming application to accept more simultaneous sessions, you can request to increase the service limit.

Learn more about increasing your service limits at Increasing Your Amazon AppStream Service Limits (p. 71).

## Other Updates

- When you delete a streaming application from the **Streaming Application** page, it no longer appears there. Previously, a deleted streaming application remained in the **Streaming Application** page with a **Deleted** status.

- Amazon AppStream displays an improved error message it cannot access the presigned Amazon S3 URL to the application installer. Previously, Amazon AppStream displayed a generic error message that did not define the problem.
- The Amazon AppStream console now correctly displays the launcher command.
- Amazon AppStream now displays an error if the streaming application stops after the client connects. Previously, this condition did not trigger and error message.

# Release for February 14, 2014

This release includes the following changes:

## Improved Deployment Experience

The Amazon AppStream console now displays a review page that shows your settings prior to deployment. On this page you can change the settings before deployment so that your streaming application will start correctly.

## Silent Installer Requirement

To install your streaming application, you need to create an installer that installs your streaming application and dependency files without any user intervention.

## Other Updates

- Amazon AppStream displays an error message if the application installer you uploaded to the Amazon S3 bucket does not comply with the requirements.
- A race condition no longer results from errors thrown by the `GetEntitlement` function.

# Release for January 24, 2014

This release includes the following changes:

## Single Download for the Amazon AppStream SDK

All of the different SDK packages are now available as a single downloadable package that contains the libraries for your Windows streaming application and your Android, iOS, and Windows client applications. Previously, you had to download a package for the Windows sample streaming application and client application, a package for the Android client application, and another package for the iOS client application.

Learn more at Downloads (p. 11).

## Improved Console Load Time

The Amazon AppStream console now loads 45% faster than the previous version. The faster load time means faster deployment for your streaming application.

# Support for Android Hardware Decoding

Amazon AppStream now supports hardware decoding for 2013 or later Android devices that use the Qualcomm processor, such as the Amazon Kindle and Google Nexus. The hardware decoder support means more consistent streaming performance over using the software decoder.

# Other Updates

- Amazon AppStream terminates the Amazon EC2 instances if the application installer did not completely install the streaming application. Previously, the EC2 instance would start and run even if the application installer could not complete the installation.
- Error messages now follow the style of error messages in other AWS services.
- You can now register a streaming application reactivated from an archived state.
- The Amazon AppStream console now features separate text boxes for the launch parameters and the path to the launcher to prevent application installer error. Previously, you entered the launch parameters and path to the launcher in the same text box.
- Amazon AppStream now allows an instance to run even if the instance is in pre-production mode.

# Document History

The following table describes the important changes to the Amazon AppStream Developer Guide.

| Change | Description | Release Date |
| --- | --- | --- |
| How to use the new AppStream example clients | New topics on using the new AppStream example clients have been added for the following operating systems:<br>• Android and Fire OS (p. 22)<br>• iPad, iPhone, or iPod Touch (p. 28) | May 14, 2015 |
| Connect to a streaming application in one click | Streaming your Application in One Click (p. 16) shows how to create a URL in a deep link format so you can connect to a streaming application through the Chrome browser or devices that use Android, Fire OS, and iOS. | May 14, 2015 |
| Updated troubleshooting information for Streaming Problems (p. 210) | The following are new troubleshooting entries.<br>• Error: No Instance Available (p. 210)<br>• Error: WebGL Not Found when using the Chrome App (p. 212)<br>• Link to Streaming Application No Longer Works (p. 212) | April 8, 2015 |
| Updated procedures for deploying the sample entitlement service | Amazon AppStream Sample Entitlement Service (p. 200) was updated because the AWS Identity and Access Management (IAM) console was updated with a new way to create a custom policy for groups and users. | April 2, 2015 |
| Updated troubleshooting information for the sample entitlement service | Error: Web Page Not Available (p. 209) has other steps to try if you get this error when you connect to the sample entitlement service. | April 2, 2015 |

| Change | Description | Release Date |
| --- | --- | --- |
| Updated procedure for using the Chrome App | The procedures for the Chrome App (p. 17) section of Streaming Your Amazon AppStream Application (p. 16) was updated because the Chrome App has a **Settings** dialog box. This dialog box allows you to use the mouse input of the Chrome App, the streaming application or both the Chrome App and streaming application, show verbose debugging information to help you solve a problem, and change the inactivity time. | March 26, 2015 |
| New topic on mouse pointers | Displaying Mouse Pointers (p. 53) explains how to display or hide the mouse pointer in the streaming and client applications. | March 26, 2015 |
| New topic on setting environment variables | Using Environment Variables (p. 205) explains how to set environment variables when you deploy your streaming application. | March 26, 2015 |
| Updated opaque data specification | Amazon AppStream Opaque Data Specification (p. 205) suggests checking if the opaque data file exists before executing commands that require this file. The opaque data block must not contain two right square brackets (]]). | March 13, 2015 |
| Added opaque data specification | Amazon AppStream Opaque Data Specification (p. 205) defines the XML format of the file that stores the opaque data. The entitlement service generates this file and sends this file to the streaming application. | February 9, 2015 |
| New procedure for deploying your streaming application | Deploying Your Streaming Application to Amazon AppStream (p. 38) shows how to deploy your streaming application by using the interactive or advanced deployment wizards. The wizard you use depends on the browser and the application installer. | January 22, 2015 |
| New procedure for using a deep link URL to connect to an entitlement service | Streaming your Application in One Click (p. 16) is a new topic on how to construct a URL called a deep link to connect to an entitlement service from a Chrome browser. When you use this URL in a Chrome browser, the Amazon AppStream Chrome App connects to your streaming application. | January 22, 2015 |
| Updated with a procedure to show a single mouse cursor | Deploying Your Streaming Application to Amazon AppStream (p. 38) has been updated with a procedure to show a single mouse cursor. By default, Amazon AppStream displays both the server side and device side cursor. The procedure has steps to set a an enviroment variable to display only the server-side cursor. | January 14, 2015 |
| Updated for the new console | The Amazon AppStream Developer Guide has been updated for the new deployment process. | November 20, 2014 |
| Initial Release | This is the first release of the Amazon AppStream Developer Guide. | November 13, 2013 |

# AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.