Microsoft

PlayReady

Developing PlayReady Clients

April 2015

Abstract

Microsoft® PlayReady® is the premier platform for protection and distribution of digital content. This white paper provides an overview of the PlayReady product suite and discusses PlayReady client technologies in terms of key concepts, platform compatibility and support for related technologies, and tools and options for developing PlayReady clients.

Legal Notice

Microsoft

Table of Contents

Introduction

Microsoft PlayReady is a comprehensive content protection and management solution for multi-industry (mainly entertainment) products and services across all platforms and types of devices. With more than 15 years and $2 billion of research and development, a full IP patent portfolio, proven robustness, and backing by a dedicated breach response team, PlayReady has become the industry-leading, digital rights management (DRM) system for protecting media content on certified devices. It provides scalable, secure, user-friendly protection of content for a wide range of distribution and consumption options. PlayReady supports:

- Multiple media content distribution models, including subscription, video on demand, rental, ad-based, and purchase (download to own).
- Multiple media delivery options, including live and on-demand streaming, and basic and progressive download.
- Emerging and established international and industry standards, including MPEG-DASH, HTML5 media extensions, Smooth Streaming, and Apple HTTP Live Streaming (HLS).
- A broad range of consumer devices, including phones, laptops, tablets, set-top boxes, Smart TVs, and connected Blu-ray™ players.
- All major client platforms, including Android, iOS, Windows®, Windows Phone®, and Xbox®.

PlayReady is also approved and adopted by major Hollywood studios, the Digital Entertainment Content Ecosystem, UltraViolet™, Smart TV Alliance, and HbbTV®.

As a content protection system, PlayReady is fundamentally designed to secure the distribution of digital content and to enable rights to be specified and enforced for that content, primarily through the use of digital encryption keys (*content keys*) and licenses. With its cross-functional and cross-platform capabilities, PlayReady is also a versatile component of any larger digital media ecosystem. The following diagram identifies major phases and components of the digital media ecosystem and PlayReady support for them.

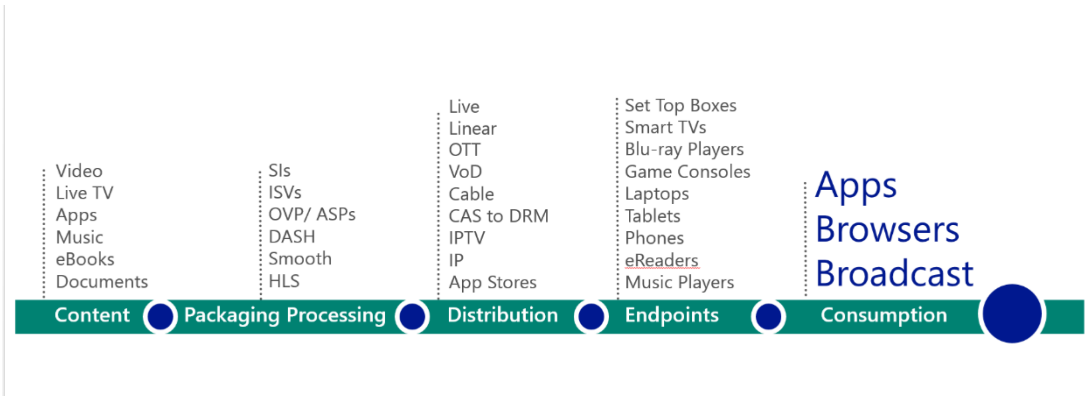


Figure 1 – PlayReady Support for Phases and Components of the Digital Media Ecosystem

In this white paper we'll provide a brief overview of the PlayReady product suite and we'll discuss PlayReady client technologies in terms of key concepts, platform compatibility and support for related technologies, and tools and options for developing, testing, and distributing PlayReady clients. To learn about PlayReady more generally, see Deploying PlayReady Technology on the PlayReady website.

The PlayReady Product Suite

To enable end-to-end content protection across the media ecosystem, the PlayReady product suite includes both client and server technologies. It also includes software development kits (SDKs) and a device porting kit for implementing those technologies on various platforms.

The following diagram identifies the primary components of the PlayReady product suite.

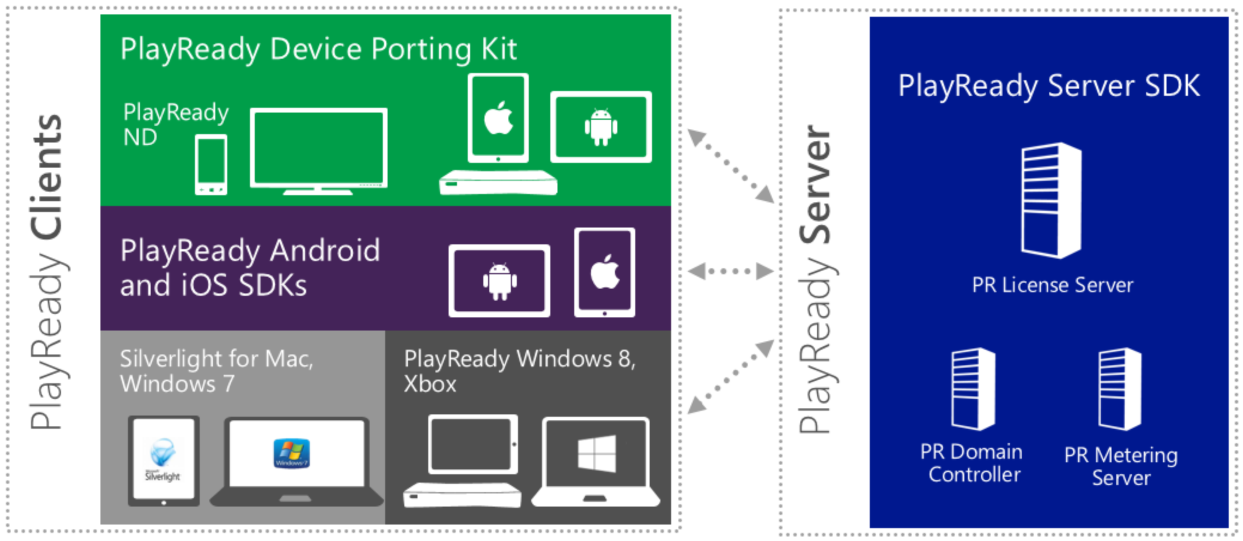
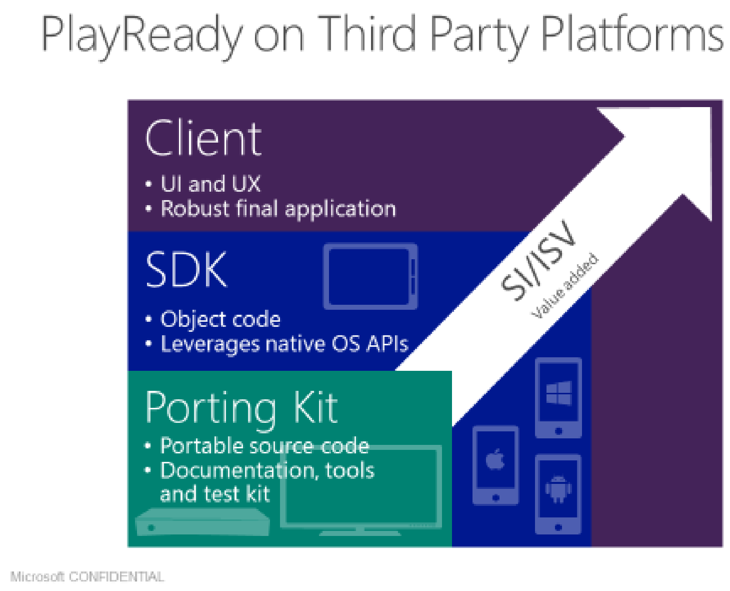


Figure 2 – The PlayReady Product Suite

To support development of PlayReady clients, Microsoft offers several SDKs, each of which is optimized for a specific major platform, and the PlayReady Device Porting Kit, which can be used to implement PlayReady functionality on virtually any platform or type of device. The porting kit is typically used for devices such as set-top boxes (STBs), Smart TVs, kiosks, and mobile devices, and it is the foundation for all the PlayReady client SDKs. To support development and deployment of PlayReady server technologies, PlayReady offers the PlayReady Server SDK, which provides detailed documentation, tools, and APIs for tasks such as generating content keys and issuing licenses.


PlayReady on Third Party Platforms
Client
• UI and UX
• Robust final application
SDK
• Object code
• Leverages native OS APIs
Porting Kit
• Portable source code
• Documentation, tools and test kit
SI/ISV
Value added
Microsoft CONFIDENTIAL

Client Technologies

A PlayReady client is a device or component — for example, an STB, app, media application, or browser plug-in — that can use PlayReady technologies to acquire and interpret licenses, decrypt and play back protected content, and enforce the license policies defined by a content provider. To perform those tasks, PlayReady clients use a combination of the media framework and other APIs provided by the host platform and PlayReady APIs, which add layers of content protection to that framework.

PlayReady provides a multitude of client options spanning all major platforms and virtually any type of device. Those options derive from native support on many platforms, the latest web standards and technologies, PlayReady client SDKs that are optimized for specific platforms, and a device porting kit that can be used to implement PlayReady functionality on virtually any platform or type of device.

To support PlayReady client development directly, Microsoft offers the following kits:

- **Silverlight SDK for Windows and MacOS X**   Used to develop PlayReady-enabled apps that are working in browser plugins. This SDK can be used to implement PlayReady protection for live and on-demand playback of MPEG-DASH and Smooth Streaming content (in combination with the Smoot Streaming Client SDK) and various PlayReady features.

- **PlayReady Client SDK for Windows Store app** Used to develop PlayReady-enabled apps that leverage native WinRT APIs. This SDK can be used to implement PlayReady protection for live and on-demand playback of MPEG-DASH and Smooth Streaming content (in combination with the Smooth Streaming Client SDK) and various PlayReady features.
- **Windows Phone SDK** Used to develop PlayReady-enabled apps that leverage native Phone APIs by using Windows Phone SDK. This SDK can be used to implement PlayReady protection for live and on-demand playback of MPEG-DASH and Smooth Streaming content (in combination with the Smooth Streaming Client SDK for Windows Phone) and various PlayReady features.
- **Xbox ADK (Xbox 360 and Xbox One)** Used to develop PlayReady-enabled apps that leverage native Xbox APIs by using Xbox Application Development Kit. This SDK can be used to implement PlayReady protection for live and on-demand playback of Smooth Streaming content as a native support and various PlayReady features. XboxOne ADK supports also MPEG-DASH content natively.
- **HTML5 Video/Audio web standard platform with Media Extensions on Internet Explorer 11 (on Windows 8.1)** Used to develop PlayReady-enabled JavaScript apps that leverage both W3C Encrypted Media Extensions (EME) APIs and Media Source Extensions (MSE) APIs. This application platform and APIs is interoperable for cross-platform devices, and can be used to implement PlayReady protection for live and on-demand playback of MPEG-DASH content,
- **PlayReady Client SDK for Android** – Used to develop PlayReady-enabled apps that leverage native Android APIs. This SDK can be used to implement PlayReady protection for media formats that Android natively supports, live and on-demand playback of MPEG-DASH and Smooth Streaming content, and various PlayReady features.
- **PlayReady Client SDK for iOS** – Used to develop PlayReady-enabled apps that leverage native iOS APIs. This SDK can be used to implement PlayReady protection for media formats that iOS natively supports such as HLS, live and on-demand playback of MPEG-DASH, Smooth Streaming and HLS content, and various PlayReady features.
- **PlayReady Device Porting Kit** – Provides a comprehensive PlayReady API, portable source code, tools, test resources, and documentation to build PlayReady clients for a wide variety of system architectures, operating system environments, and device classes. The porting kit is frequently used by device manufacturers and providers to implement PlayReady technologies on devices such as STBs, Smart TVs, kiosks, mobile devices, HDMI dongles, and even on Android and iOS devices.

Each kit provides APIs, a sample media application, tools, detailed technical information, and task-based guidance for implementing PlayReady protection mechanisms and features on the target platform or device. Kit users can also access a fully functioning PlayReady server to test a client's ability to acquire licenses and decrypt and play back content. To learn more about the kits, see Development Tools and Options.

Server Technologies

PlayReady server technologies help prepare content for distribution, issue licenses to PlayReady clients, and optionally manage domains and meter content usage. To perform those tasks, PlayReady server technologies provide the following services:

- **License Service** – Stores licenses that specify protection information and rights for using content, and receives and responds to authentication and license requests from PlayReady clients. License services handle tasks such as managing usage rules and policies that are applied to licenses, implementing business logic for authorization and control of usage rights, and building and issuing licenses to PlayReady clients.
- **Domain Controller** – Determines what a given *domain* represents. A domain is a virtual entity that represents a group of users or devices that can share licenses for media content — for example, a game console, tablet, and phone within the same household. The domain controller can store a list of entities that are associated with a domain, handle requests to join or leave a domain, and enforce policies that define how many devices or users can be part of a domain.
- **Metering Service** – Stores information about the number of times that a media file is played, based on playback counts that PlayReady clients upload to the service. The metering service also obtains metering certificates to help ensure the security of metering data. Note that metering does not affect any behavior on a user's system or identify the user. It simply allows content providers to assess royalties accurately.

Of those services, only the license service is integral to implementing PlayReady for content protection. The domain controller and metering service are optional, depending on a content provider's business and distribution model.

At their core, PlayReady server technologies are SOAP-based web services that run in Internet Information Services (IIS) and provide standard operations, messages, and schemas for communicating, accessing, and managing protection data on a server. All the services are configured, deployed, and managed by using the PlayReady Server SDK.

Fundamental Concepts

PlayReady media files are encrypted by AES encryption algorithm, so those can be moved, archived, copied, and distributed but their content cannot be consumed without a *license*. A license contains a *content key*, which is a symmetric cryptographic key that is used to decrypt a media file, and it specifies *policies* that define how and under what conditions a file's content may be used.

In the following sections, we'll provide an overview of the media distribution workflow and the role of PlayReady client and server technologies within that workflow. We'll also explain key concepts that enable PlayReady clients to protect various types of media content. To learn more about these and related concepts, see Deploying PlayReady Technologies on the PlayReady website.

Media Distribution Workflow

In the typical flow of content within a content distribution system, an encoding/packaging server takes unprotected media content and encrypts and packages the content for distribution. After the content is packaged, it's delivered to a distribution network. The distribution network then delivers the content to various endpoints for consumption.

The following diagram illustrates the typical flow of content and licenses within a distribution system.

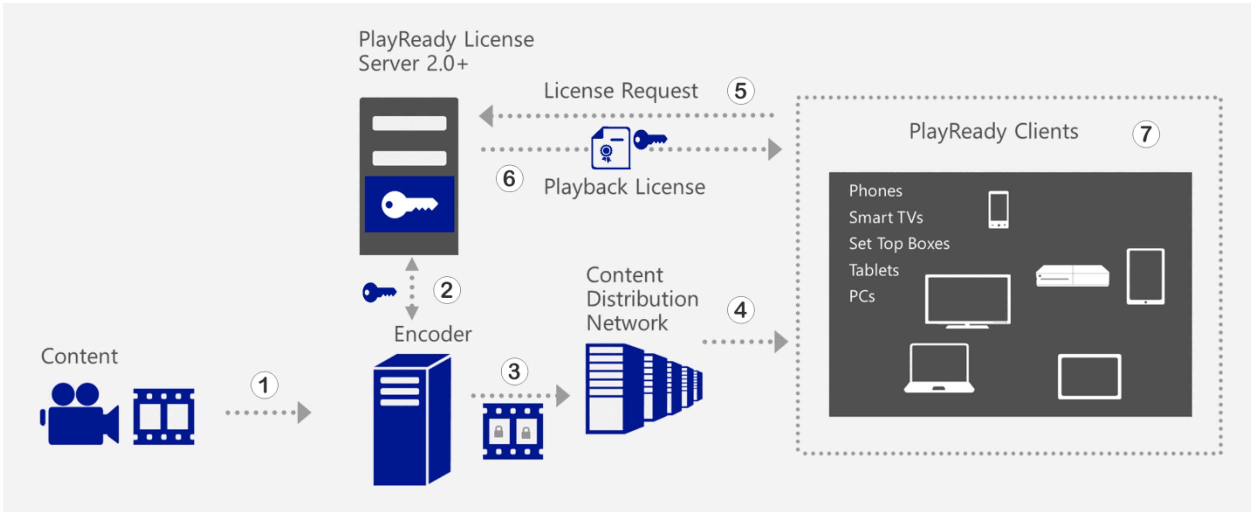


Figure 3 – Typical Flow of Content and Licenses within a Distribution System

In the preceding diagram, the basic steps are:

1. A media asset is sent to an encoder for encoding and encryption.
2. The encoder does the following:
    - Encodes the media asset.
    - Encrypts the resulting media file by using a content key, which it shares with PlayReady license services. PlayReady uses AES-128 CTR encryption.

- Adds a PlayReady header to the media file. The PlayReady header is a rights management header that enables a PlayReady client to decrypt and acquire a license for the file.

3. The encoder packages the media file and sends it to a content distribution network for delivery to PlayReady clients.
4. The content distribution network sends the media file to a PlayReady client in response to a request from the client.
5. The client does the following:
   - Attempts to play back the media file and finds the PlayReady header in the file. Because the file contains this header, the client determines that the file is encrypted and a license must be acquired to play back the file.
   - By using information in the PlayReady header, the client sends a license request to a designated license server. The request includes a content identifier, which uniquely identifies the media file, and the public key from a key pair that is unique to the client device (*device public key*).
6. The license server receives the license request and does the following:
   - Authenticates the device.
   - Executes business logic to ensure that the user and the device are authorized to consume the content in the media file.
   - Builds the license and adds policies that specify usage rules and restrictions for consuming the content in the media file. The license also includes the content key for the file.
   - Encrypts the license by using the client device's public key. Using the device's public key to encrypt the license both protects license data and ensures that the license applies to only that device.
   - Sends the resulting license to the client.
7. The client receives the license and decrypts it by using the private key from the key pair that is unique to the client device (*device private key*). The client then uses the license to securely decrypt and play back the media file in accordance with the policies specified in the license.

The preceding steps represent simple packaging and consumption processes. Processes vary between systems due to many factors, such as distribution and delivery model, and PlayReady supports a wide range of models.

The PlayReady Header Object

The PlayReady header object is a placeholder to store PlayReady rights management header that enables PlayReady clients to acquire a license for and decrypt the content in a media file. It can also store an embedded license directly in a media file. The header is added to and subsequently stored in a media file or, for streaming content, a media manifest file when the file is packaged for distribution.

Each PlayReady rights management header contains a standard set of metadata about a license. This includes a *key ID* that identifies the content key, the type of encryption used to encrypt the file, the URL for the applicable license acquisition service (*LAURL*), and any custom attributes that a content provider chooses to define. PlayReady rights management headers are encoded using UTF-16.

In a basic workflow, a PlayReady client finds and extracts the PlayReady rights management header from a media or media manifest file when it begins parsing the file. The client then processes the header data to acquire a license for and ultimately decrypt the content with the content key in the acquired license. In most cases, this includes sending the header to the appropriate license service, which in turn processes the header data, verifies that the license request is from a valid client (*client authentication*), and then issues a license to the client.

For detailed technical information about the PlayReady header, see the PlayReady Header Object specification on the PlayReady website.

Licenses and Policies

A license contains two fundamental protection components, a *content key* and *policies*. A content key is a symmetric cryptographic key that is used to decrypt a media file. Policies are usage rules that specify how, when, and where a file's content can be consumed. When a license is built for a media file, the content key and appropriate policies are added to the license for the file. When a license is issued to a PlayReady client, the client uses the content key and policies to decrypt and play back the file according to the rules specified in the license.

In PlayReady, policies support several types of usage rules, most commonly: time-based restrictions, which specify a time frame that a license is valid for; output-protection levels, which indicate whether playback is restricted to specific types of output ports on devices; and, allowable-export restrictions, which specify restrictions for moving or exporting content to a different content protection scheme. Both output-protection and allowable-export policies are governed by the PlayReady compliance and robustness rules.

To support a variety of distribution scenarios, PlayReady provides several enhanced license model. The following table identifies and describes those models.

Enhanced License Model	Description
Simple license	A license contains two fundamental protection components, a content key and policies. A content key is a symmetric cryptographic key that is used to decrypt a media file. Policies are usage rules that specify how, when, and where a file's content can be consumed.
Chained license	A series of associated licenses that collectively specify content keys and policies for one or more media files. Each license chain includes a root license and one or more leaf licenses that are associated with and dependent on the root license: • Leaf license – Contains the content key (*leaf key*) for a specific media file, defines any policies that are specific to the file, and references and inherits policies from the root license that it is associated with. • Root license – Contains an uplink key (*root key*) that is used to encrypt content keys (*leaf keys*) in all the leaf licenses that are associated with it. It also stores policies that apply to all the content that is licensed by those leaf licenses. Chained licenses are frequently used in subscription-based distribution models because they enable providers and users to renew licenses for a large number of media files more efficiently.
Scalable chained license	An extension of the chained license model that allows a provider to issue only a subset of content keys instead of all required leaf keys for all content. Designed to support Live TV scenarios, this model enables a provider to create a master key set that specifies all the channels and regions for a service. The master key set is then used when generating scalable root and scalable leaf licenses for channel content. The scalable root license defines the channels and regions for a user's account and a client requests that license before streaming any content. Scalable leaf licenses are generated and delivered in-stream for each live channel.

Enhanced License Model	Description
Embedded license	A license that is acquired and subsequently stored in the PlayReady header of a media file. Because the license is stored or "embedded" in the media file, instead of being issued and stored separately, it's immediately available. This can facilitate user scenarios such as moving content from one device to another, backing up content, and playing back content with a device that is offline. In addition, an embedded license can be bound to a domain certificate, which allows the media file to be played by other devices that are members of the same PlayReady domain.

License Acquisition Models

PlayReady supports two license acquisition models, proactive and reactive. In the proactive model, a client acquires and stores a license locally before playback is initiated, and the acquired license typically persists in local storage. This model is typically used in purchase (download to own), rental, and subscription-based scenarios with/without time-based restriction policies.

In the reactive model, a client acquires a license on demand when playback is initiated. This model is typically used in streaming and progressive playback scenarios. The reactive model also supports use of *non-persistent licenses*, which is a license that is stored temporarily in a client's local memory, used only once, and expires automatically when playback ends. This type of license works well in scenarios that don't support or require offline playback.

PlayReady clients can use a mixture of proactive and reactive licenses. The PlayReady client SDKs and the PlayReady Device Porting Kit can be used to implement both acquisition models.

Client Authentication

In the context of content protection, client authentication may occur at two levels, the client device and the user. PlayReady handles device authentication automatically through its use of *device keys* when issuing licenses to PlayReady clients, as illustrated in the following diagram.

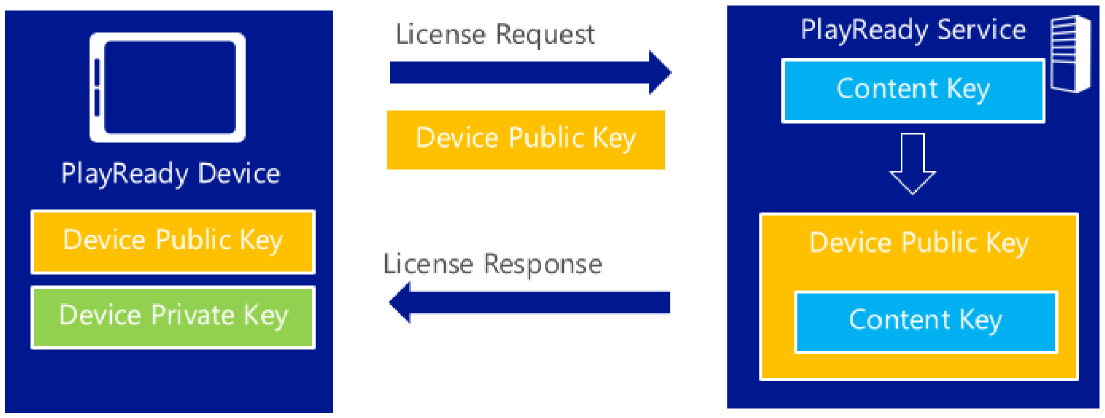


Figure 4 – Role of Device Keys in the Licensing Process

Each PlayReady client device has a unique public-private key pair. When a client sends a license request to a license service, it includes its device public key. The license service uses that public key to verify that the client device is a valid PlayReady client before it issues the license. If the device is a valid PlayReady client, the license service uses the device's public key to encrypt the content key for the media file and then sends the resulting license to the client. The client must then use its device private key to decrypt and play back the content. This authentication model ensures that the issued license is valid and usable only on that device (in the case of the device binding).

For user authentication, PlayReady license services are designed to integrate with the authentication model that a content provider chooses to implement. In practice, user authentication is integrated with PlayReady in either of two ways:

- Clients send license requests to a license service indirectly through a proxy server. In this model, each license request is first sent to a proxy server. The proxy server then authenticates the user and passes license requests from valid users to the license service.
- Clients send license requests to the license service directly and include an authentication token. In this model, an authentication token is passed to a server with a license request and the license service can validate the token with a separate authentication service.

The PlayReady SDKs and Device Porting Kit can be used to integrate a PlayReady implementation with either authentication model.

Device Activation and Revocation

Activation is a process that allows licenses to be issued and bound to a PlayReady client device. For this reason, a client device must be activated before most PlayReady protection mechanisms will work. By default, activation occurs automatically the first time a PlayReady client attempts to play back content that is protected by PlayReady technologies. To avoid any playback lag during activation, you can implement activation proactively in a PlayReady client. The PlayReady client SDKs and Device Porting Kit explain how.

Revocation is a process that identifies a PlayReady client device or a model of client devices whose security or compliance is compromised and prevents the device from accessing additional licenses for protected content. For example, if a specific device model was compromised, the corresponding model certificate can be revoked to prevent devices of that model from playing protected content until the issue is addressed through a firmware or other type of update. When identified, the device or the model of device may be added to a revocation list. Revocation lists are managed by Microsoft and used by license servers to deny license requests from devices that have been revoked. A software and/or firmware in that client device should be upgraded with a new software and/or firmware which address the security or compliance issue.

Device and Platform Compatibility

To support a broad spectrum of media distribution models, PlayReady client technologies are compatible with all major platforms and virtually any type of device — they work on a variety of device classes, architectures, and system environments. This means that you can implement a PlayReady client on any type of in- or out-of-home device, including mobile phones, tablets, laptops, game consoles, STBs, Smart TVs, kiosks, and embedded displays. You can also implement PlayReady on network device (ND) transmitters and receivers, which enable users to share media content between devices that are connected to the same Internet Protocol (IP) network. Overall, any device that is capable of receiving and playing media content can host PlayReady client technologies.

In addition, a PlayReady client can use any application model that is supported by the target platform. For example, a PlayReady client might be a desktop application for a Windows 7 or Linux computer, a universal app for a Windows 8 tablet and Windows Phone 8 device, an Xbox One app, or an app for an Android phone or iPad. It might also be a Silverlight® or HTML5 media application.

The breadth of client options derives from a combination of factors — native PlayReady support on many platforms, the PlayReady client SDKs and Device Porting Kit for other platforms, and PlayReady support for the latest web standards and technologies. Another significant factor is the PlayReady architecture and development model. PlayReady makes extensive use of the media framework, interfaces, and APIs of the host platform and combines them with PlayReady APIs that add layers of content protection. Consequently, you can build a client that uses the application model, features, and APIs that you want for your specific business needs.

The following Microsoft platforms provide native support for PlayReady:

- Windows 7 and later
- Windows Phone, all versions
- Xbox One and Xbox 360®

For other platforms, you can build a PlayReady client by using a PlayReady SDK that is designed specifically for that platform (Android or iOS) or the PlayReady Device Porting Kit. The porting kit provides portable, platform-independent source code and a comprehensive API for calling PlayReady functions. To help you optimize a PlayReady client for a specific platform, both the source code and functions are designed to accommodate differing device capabilities and configurations. To learn more about the PlayReady kits, see Development Tools and Options.

Moving away from specific platforms, you also have the option of leveraging the latest web standards and technologies to build a PlayReady client that works on multiple platforms. Both Silverlight and MPEG-DASH (MPEG Dynamic Adaptive Streaming over HTTP) provide cross-platform solutions for delivering protected media content. Silverlight provides built-in PlayReady support for multiple media formats and it works on Windows, Mac OS X, and Windows Phone devices. On Windows and Mac OS X devices, Silverlight can be used if a free specialized plug-in is installed and it can be used for both in- and out-of-browser scenarios. PlayReady supports both types of scenarios.

MPEG-DASH, when used with HTML5 Media Source Extensions (MSE) and Encrypted Media Extensions (EME), also provides an end-to-end framework for delivering protected content to multiple platforms. Like Silverlight, it can be used for both in- and out-of-browser scenarios. Unlike Silverlight however, it doesn’t require any specialized plug-ins and it consists of only ISO and W3C standards. PlayReady fully supports MPEG-DASH, MSE, and EME. In addition, Windows 8.1 introduced built-in support for EME APIs.

Supported Media Formats

PlayReady supports a plethora of standard encoding and delivery formats for essentially any type of media content, including movies, premium live content, music, games, and images. PlayReady can be bound to any type of file format by defining the binding specification to the file format. PlayReady isn’t tied to a specific media format. Instead, PlayReady support is determined by whether a client understands the PlayReady binding for a media file. In addition, PlayReady uses only industry-standard encryption technologies to protect content. Bolstering that support, Microsoft continuously publishes specifications about using PlayReady to protect content in commonly used formats, and most commercial encoders and packagers support PlayReady technologies.

PlayReady supports nearly all standard and platform-specific media formats. This includes the new Common File Format (CFF), which the Digital Entertainment Content Ecosystem (DECE) designed to work with UltraViolet players on all platforms and with all protection systems approved by the DECE. PlayReady is approved by the DECE. CFF can be used by other distribution systems and is expected to be used more broadly, partly because MPEG is currently updating the MPEG-4 container format to include aspects of CFF. Here is major media formats which can bind PlayReady.

Name	Container	Codec	Description
PlayReady Media Format File (.PYV or .PYA)	ASF	Video: WMV Audio: WMA	This is the binding of PlayReady to the Advanced Systems Format (ASF).
PlayReady Envelope File	Arbitrary	Arbitrary	This is the binding of PlayReady to any type of format.
Protected Interoperable File Format (PIFF)	ISOBMFF	Video: WMV, H.264 Audio: WMA, AAC	This file format is and ISO Base Media File brand with the extension of the binding of PlayReady. This format is a single encoding format appropriate for download, broadcast, streaming and multi-bitrate adaptive streaming.
DECE Common File Format	ISOBMFF	Codecs capable to ISOBMFF container	This file format is and ISO Base Media File brand. PlayReady is bound by using ISO Common Encryption. The format is used for UltraViolet DTO content.
DECE Common Streaming Format	ISOBMFF	Codecs capable to ISOBMFF container	This file format is and ISO Base Media File brand. PlayReady is bound by using ISO Common Encryption. This supports late binding for DTO content using SMPTE packaging format, late binding for streaming using MPEG DASH Media Presentation Descriptor file.

Name	Container	Codec	Description
PlayReady MPEG-2 TS	MPEG2-TS	Codecs capable to MPEG-2 TS container	This file format can be used for PlayReady protected HLS. The binding specification is disclosed only to the PlayReady licensees. This MPEG-2 TS format is compatible with Azure Media Services.

Of the various encoding options for video content, some of the more commonly used formats that PlayReady supports are H.263, H.264, and H.265 codecs. For audio content, supported formats include AAC, AAC+, and WMA codecs. In addition, PlayReady supports UHD (ultra high definition), HD (high definition), SD (standard definition), and PD (portable definition) profiles. PlayReady support with the codecs is also determined by whether a client understands the PlayReady binding for a codec.

For content distribution, PlayReady also supports multiple distribution options, of which there are two primary methods:

- Download, basic and progressive – Allows a client to begin playing content only after all or some of a media file is downloaded to the client device. In a basic download, playback cannot begin until the entire media file is downloaded. In a progressive download, playback begins as soon as the initial portion of the file is downloaded and the rest of the file is downloaded in parallel with playback. The most commonly used formats for downloadable content are Moving Picture Experts Group (MPEG), Windows Media® Video (WMV), and Apple QuickTime Movie (MOV). PlayReady supports all these formats.
- Stream, live and on demand – Allows a client to begin playing content immediately instead of waiting for all or some of a media file to be downloaded. If the content is encoded at multiple bit rates, a client can optimize playback quality in real time by requesting segments from various encoded bit rates of the media asset. This process, called *adaptive bit rate (ABR) streaming*, begins immediately when playback begins and continues as playback proceeds. The client determines which bit rate to request based on factors such as available network bandwidth, buffer status, and CPU capacity. The most commonly used formats for streaming content are MPEG-DASH, Smooth Streaming, and HTTP Live Streaming (HLS). PlayReady supports all these formats.

The primary difference between the two methods is how content is received and stored by a client. A client should render media files the same way regardless of whether they are downloaded and played from local storage or streamed.

All of the common streaming formats — MPEG-DASH, Smooth Streaming, and HLS — support both live and on-demand ABR streaming. However, only MPEG-DASH is an international standard. Other signal differences between the formats are inherent support for multiple content protection systems and supported client platforms. Although Smooth Streaming and HLS are commonly used at this time, MPEG-DASH is quickly becoming the most cross-platform interoperable format.

MPEG-DASH

Developed by the DASH Industry Forum, MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is the first open ISO standard for ABR streaming technology. The standard specifies formats for delivering live or on-demand media content to a full spectrum of clients and devices via HTTP. The standard (ISO/IEC 23009) was first published in April 2012.

In addition to defining standard file formats, MPEG-DASH is the foundation for a larger media framework that enables providers to package, protect, distribute, and deliver only one set of media files to almost any platform. The MPEG-DASH framework is designed to:

- Provide users with the highest quality video experience – MPEG-DASH combines premium video features with the best ABR streaming technologies developed thus far. Supported features include trick modes (seek, fast forward, rewind), closed captions, multiscreen capability, and multitrack capabilities for audio and subtitles. MPEG-DASH also supports business features such as ad insertion.

- Eliminate dependencies on proprietary and platform-specific formats – Media file formats are based on ISO MPEG formats, which are the most commonly used media formats. In addition, MPEG-DASH isn't tied to only one specific codec.

- Reuse existing content, devices, and delivery infrastructure – Specialized media servers aren't required for distribution. Instead, HTTP web servers and protocols can be used, which means content can be delivered through existing Internet infrastructures. In addition, content can be streamed to various types of applications without requiring specialized plug-ins. If used with HTML5 Media Source Extensions (MSE) and HTML5 Encrypted Media Extensions (EME), providers can protect and deliver the same media file to multiple platforms.

- Work with multiple content protection systems – MPEG-DASH doesn't use a DRM-specific encoding format. Instead, protection is applied by using the format and methods defined by the ISO Common Encryption Scheme (CENC). Consequently, a media file can be encoded and encrypted once and used with multiple content protection systems, including PlayReady.

The following diagram illustrates the key components of the framework.

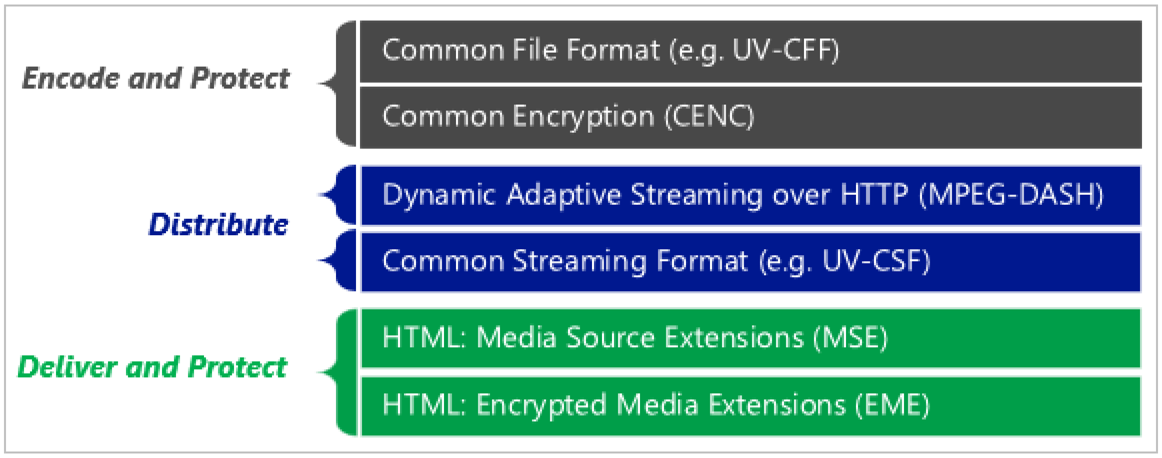

Encode and Protect
Common File Format (e.g. UV-CFF)
Common Encryption (CENC)
Distribute
Dynamic Adaptive Streaming over HTTP (MPEG-DASH)
Common Streaming Format (e.g. UV-CSF)
Deliver and Protect
HTML: Media Source Extensions (MSE)
HTML: Encrypted Media Extensions (EME)

Figure 5 – Key Components of the MPEG-DASH Framework

Delving more deeply into content protection, MPEG-DASH is designed to work with any content protection system that supports the CENC standard (ISO/IEC 23001). CENC defines a common format for capturing the encryption data that a client needs to decrypt a file. All other protection mechanisms — for example, how content keys are acquired and how policies are associated with a media file — remain specific to the protection system that is being used. Consequently, the same media file can be used with multiple content protection systems. PlayReady technologies fully support the CENC standard and can be used to protect MPEG-DASH content. For detailed information about implementing PlayReady support for MPEG-DASH, see DASH Content Protection using Microsoft PlayReady on the PlayReady website.

Client support for the MPEG-DASH framework includes any web browser or application that supports MPEG-DASH formats and the W3C's HTML5 and HTML5 Media Source Extensions (MSE) recommendations. MSE defines a JavaScript API for streaming content and providing ABR and live streaming features without requiring any specialized plug-ins or application customization. It essentially enables you to build streaming media applications by using only HTML and JavaScript. Currently, HTML5 and MSE are supported by the latest versions of Internet Explorer, Chrome, Safari, and Firefox.

To protect content that is delivered by using MSE, providers use HTML5 Encrypted Media Extensions (EME). Also defined by the W3C, EME is an API that enables a JavaScript application to decrypt a media file and work with a content protection system to apply protection mechanisms. The EME architecture uses a *Content Decryption Module (CDM)* that translates generic EME APIs into the requisite data for a specific content protection system. Combined with CENC, EME enables a single JavaScript application to deliver the same protected content to every platform that supports EME and uses the same CDM (protection system).

Starting with Windows 8.1, Windows provides APIs that were built specifically to support EME and it provides a PlayReady framework that uses CDM functionality. Internet Explorer 11 uses those APIs and other Windows browsers and applications can do the same. By using the

PlayReady Device Porting Kit, the PlayReady CDM can be implemented on other platforms spanning a variety of device types. Note that Windows Phone does not currently support EME.

To facilitate browser support, Microsoft published a specification that defines a general interface for mapping EME methods and events to protection systems and platform components. This interface — the *Content Decryption Module interface* (*CDMi)* — enables an application to access a platform CDM and expose the CDM functionality needed to protect content, without requiring browsers to license and incorporate DRM components. Use of the CDMi also provides an additional layer of protection because it uses platform components, instead of browser components, for critical protection functions such as decryption. Although it was developed by Microsoft, the CDMi is an open interface that can be used by any browser with any content protection system. For detailed technical information about CDMi, see the Content Decryption Module Interface specification on the PlayReady website. For an in-depth look at how the various HTML5 extensions and the CDMi work together, see Interoperability, Digital Rights Management and the Web on the PlayReady website.

Smooth Streaming

Smooth Streaming, also known as HTTP Smooth Streaming or Microsoft Smooth Streaming, is developed by Microsoft as an IIS Media Services extension and a component of Microsoft Media Foundation, which is the Windows multimedia platform. It can be used for live or on-demand ABR streaming. Streams can be served by a standard HTTP web server if the IIS Media Services extension is installed, or a dedicated media server.

Smooth Streaming is natively supported by the Windows client operating system (Windows Vista® and later), the Windows Server® operating system (2003 and later), Xbox 360, and Xbox One™. It's also supported by Silverlight 4 and later, which means that content can be delivered to devices running Windows Phone 7 and later, and web browsers and Silverlight applications on Windows and Mac OS X devices that have the Silverlight plug-in. Support can be implemented on additional platforms by using the Smooth Streaming Client Porting Kit.

Smooth Streaming is based on the ISO base media file brand's Protected Interoperable File Format (PIFF). Standardized by Microsoft, PIFF defines a standard format for audio-video containers and ABR metadata. It also defines an optional encryption scheme that is designed to interoperate with multiple content protection systems. The only differences are how content keys are acquired and how policies are associated with a media file. For detailed technical information about PIFF, see the Protected Interoperable File Format specification on the Microsoft website.

PlayReady technologies provide full support for Smooth Streaming content. PlayReady client support includes clients for all supported versions of Windows operating systems, Windows Phone, Xbox One, Xbox 360, and web browsers and Silverlight applications on Windows and Mac devices that have the Silverlight plug-in. Support can be extended to other platforms by using the appropriate PlayReady client SDK or the PlayReady Device Porting Kit.

HLS

HTTP Live Streaming (HLS) is developed by Apple for the iOS and Mac OS X (Snow Leopard and later) operating systems and Apple's QuickTime and Safari software. It can be used for live or on-demand ABR streaming. As its name implies, streams can be served by a standard HTTP web server; a dedicated media server isn't required.

HLS is the only streaming format that is natively supported by Apple devices, including iPhones, iPads, iPods, and Apple TV. It's also supported by Android devices and some STBs and third-party players. HLS is based on the MPEG-2 file format and it provides a standard AES-128 encryption mechanism. By default, HLS exchanges content keys by using a combination of HTTPS and either device-specific credentials or an HTTP cookie. The result is a simple DRM system.

For more robust protection, PlayReady technologies can be used to protect HLS content either in addition to or in lieu of default HLS encryption mechanisms. PlayReady client support can be implemented on iOS and Android devices by using the appropriate PlayReady client SDK and on other HLS-supported platforms by using the PlayReady Device Porting Kit.

Backward Compatibility with Windows Media DRM

PlayReady license services and some of clients support backward compatibility with content that is protected by Windows Media® DRM (WMDRM) technology. However, there are some constraints:

- License acquisition – A PlayReady license service can issue a license for WMDRM content only if a PlayReady client requests the license by using the PlayReady message protocol. PlayReady license services cannot issue a license to a WMDRM client (such as Windows Media Player) by using the legacy WMDRM message protocol.
- Clients – Although PlayReady clients offer full support for WMDRM content, WRDRM clients cannot decrypt and use content that is protected by PlayReady.

If those constraints are addressed, PlayReady license services and clients can issue and acquire licenses for WMDRM and PlayReady content concurrently. This means that a content provider can deploy and migrate to PlayReady in manageable stages. Silverlight and PlayReady Client SDK for Windows Store app can support this backward compatibility.

Development Tools and Options

Your choice of development tool for building a PlayReady client depends primarily on the target platform, the application model that you want to use, and the media formats and distribution methods that your client needs to support.

For Microsoft Windows, Windows Phone, Android, and iOS platforms, you can use PlayReady Client SDKs that are designed specifically for those platforms. For web applications that work on multiple platforms, your choice of development tool depends largely on the technology that you want to use. For other platforms and types of devices, you can use the PlayReady Device Porting Kit.In the following sections we'll walk through the various development tools and options for building PlayReady clients, starting with the native application and web application on PlayReady supported platforms, and other type of platforms and devices with PlayReady Device Porting Kit.

Native Applications

With the PlayReady Client SDKs, you can develop native applications for Windows, Windows Phone, Android, and iOS devices to provide more rich experiences to end-users with calling platform's native APIs. But this type of application is dependent on the platform functionalities and interfaces, such as platform media player function. You need to consider application development efforts (costs, resources, and so on) for each platform which you want to deliver the contents.

PlayReady Client SDK for Windows and Windows Phone

For Microsoft platforms, you can use standard platform SDKs in combination with a PlayReady client SDK that's designed specifically for the platform. For the latest Microsoft platforms, those SDKs are:

- For Windows 8.1, the PlayReady Client SDK for Windows 8.1 Store Apps
- For Windows Phone 8.1, the PlayReady Client SDK for Windows 8.1 Phone Apps

Both SDKs are available from the Visual Studio Gallery. To learn more, see Developing PlayReady Windows Store and Web Apps in the Windows Developer Center on MSDN®. Microsoft also published a sample universal app that demonstrates how to play back PlayReady-protected content on Windows 8.1 and Windows Phone 8.1 devices. You can learn more about that app on the sample's download page in the Visual Studio Gallery.

Smooth Streaming Client SDK for Windows Store app/Windows Phone can support clear and PlayReady-protected Smooth Streaming content. With Player Framework for Windows and Windows Phone, clear and PlayReady-protected MPEG-DASH content can be supported.

Xbox ADK

Xbox 360 and Xbox One natively support PlayReady. To develop an application for either console, use the Xbox Application Development Kit (ADK). To learn more about developing Xbox applications, see the Xbox Developers Program website.

PlayReady API included in the Xbox One ADK is very similar with PlayReady Client SDK for Windows and Windows Phone. An almost part of JavaScript application code for Windows and Windows Phone can be re-used for Xbox One application.PlayReady Client SDK for Android

The PlayReady Client SDK for Android provides PlayReady libraries, tools, and reference information for building PlayReady-enabled apps that work on Android devices running Android version 4.0 (API level 14, Ice Cream Sandwich) or later.

The SDK is designed to be used with the Android SDK and Eclipse IDE, which you can download as part of the Android Developer Tools (ADT) bundle from the Android Developers website. After you install the ADT bundle and configure it by using the Android SDK manager, simply install the PlayReady Client SDK for Android to begin developing your app. To install and use the PlayReady SDK with ADT, your computer must be running Windows Vista or later or meet the base ADT requirements for Linux.

By using the SDK, you can easily integrate PlayReady functionality with native Android APIs and implement PlayReady protection for media formats that Android natively supports, such as HLS. For example, the Android **MediaPlayer** class controls playback of media content on an Android device. By using the PlayReady SDK, you extend that class with PlayReady APIs to enable playback of media content that is protected by PlayReady technologies.

You can also use the SDK to implement PlayReady protection for MPEG-DASH and Smooth Streaming content that's delivered live or on demand, as well as offline playback of files in the Common File Format (CFF) or Protected Interoperable File Format (PIFF) formats. For video content, PlayReady support includes PD, SD, and HD profiles, as well as a Timed Text Markup Language (TTML) parser for captions.

The SDK also helps you implement support for a wide range of PlayReady protection mechanisms and features, including device activation and revocation, network device (ND) receiver functionality, proactive and reactive license acquisition, enforcement of output-protection levels, embedded licenses, chained licenses, domain management, and metering. For Live TV scenarios, you can also implement support for key rotation, scalable chained licenses, blackout services, and ad insertion.

To help you develop a PlayReady-enabled app, the SDK provides the following resources, all of which are customized specifically for Android development:

- A complete set of PlayReady shared libraries.
- A sample application that you can use as a reference or starting point for your own app.
- A programming guide that explains how to implement specific PlayReady features and provides details about platform-specific considerations and limitations.

- Detailed reference information for relevant interfaces, classes, enumeration types, and error codes.
- Instructions for preparing to release your app.

In addition, the SDK includes specifications for PlayReady extensions, protocols, and formats.

As an SDK user, you also have access to a PlayReady test server that you can use to test and debug your app for tasks such as acquiring content and licenses, joining and leaving domains, and processing metering data.

To obtain the PlayReady Client SDK for Android, you can purchase a PlayReady license or request a PlayReady evaluation agreement. To learn more, see Licensing Options.

PlayReady Client SDK for iOS

The PlayReady Client SDK for iOS provides PlayReady libraries, tools, and reference information for developing PlayReady-enabled apps that work on devices running iOS version 6.0 or later. Applicable devices include iPad, iPhone, and iPod touch.

The SDK is designed to be used with Xcode 4.5.2 or later running on Mac OS X Lion (10.7) or later. You can download Xcode from the Apple Developer website. After you install and configure Xcode, set up your provisioning profile with the iOS Developer Program, and install the PlayReady Client SDK for iOS to begin developing your app.

Much like the PlayReady SDK for Android, the PlayReady SDK for iOS helps you integrate PlayReady functionality with native iOS APIs and implement PlayReady protection for media formats that iOS natively supports, such as HLS. For example, the iOS Media Player framework can be used to play back video on an iOS device. By using the PlayReady SDK, you extend that framework with PlayReady APIs that enable playback of media content that's protected by PlayReady technologies.

You can also use the SDK to implement PlayReady protection for live and on-demand playback of MPEG-DASH and Smooth Streaming content, as well as offline playback of CFF and PIFF files. For video content, PlayReady supports PD, SD, and HD profiles, and a TTML parser for captions.

The PlayReady SDK also helps you implement support for PlayReady protection mechanisms and features such as device activation and revocation, network device (ND) receiver functionality, proactive and reactive license acquisition, enforcement of output-protection levels, embedded licenses, chained licenses, domain management, and metering. For Live TV scenarios, you can also implement support for key rotation, scalable chained licenses, blackout services, and ad insertion.

To help you develop a PlayReady-enabled app, the SDK provides the following resources, all of which are customized specifically for iOS development:

- A complete set of PlayReady shared libraries.

- A sample application that you can use as a reference or starting point for your own app.
- A programming guide that explains how to implement specific PlayReady features and provides details about platform-specific considerations and limitations.
- Detailed reference information for relevant interfaces and classes, protocols, enumeration types, notifications, and error codes.
- Instructions for preparing to release your app.

In addition, the SDK includes specifications for PlayReady extensions, protocols, and formats.

As an SDK user, you also have access to a PlayReady test server that you can use to test and debug your app for tasks such as acquiring content and licenses, joining and leaving domains, and processing metering data.

To obtain the PlayReady Client SDK for iOS, you can purchase a PlayReady license or request a PlayReady evaluation agreement. To learn more, see Licensing Options.

Web Applications

While the SDKs are ideal development tools for Windows, Android, and iOS devices and the PlayReady Device Porting Kit is an ideal solution for other platforms, you have additional options for developing PlayReady clients. If you prefer to develop a PlayReady client that works on multiple platforms, a web or browser-based application may be the best solution. Currently, optimal solutions for these types of applications derive from using Silverlight or a combination of MPEG-DASH, HTML5 Media Source Extensions (MSE), and HTML5 Encrypted Media Extensions (EME).

Silverlight

Silverlight is a cross-platform technology for developing web, desktop, and mobile applications that work on Windows, Mac OS X, and Windows Phone devices. It uses various web standards — primarily eXtensible Application Markup Language (XAML) and the HTML Document Object Model (DOM) — and a lightweight version of the .NET Framework. It also provides built-in support for PlayReady technologies.

On Windows and Mac OS X platforms, Silverlight can be used for both in- and out-of-browser scenarios if a free plug-in is installed. The plug-in works on computers running Windows Vista or later and Mac OS X Lion (10.5.7) or later. For in-browser scenarios, the plug-in is available for current versions of Internet Explorer and versions of Chrome, Safari, and Firefox which support Netscape Plugin Application Programming Interface (NPAPI). Silverlight is primarily designed for hosting in a Web page. However, there are several alternative hosting options that enable Silverlight to run outside the browser or within another host environment. See more details at "Alternative Hosting" on MSDN.

To develop a Silverlight application for Windows or Mac OS X, you can use Visual Studio 2010 or later and the Silverlight Tools for Visual Studio. To develop a Silverlight application for Windows Phone, use the Silverlight for Windows Phone Toolkit. You can download the tool kits and learn more about developing Silverlight applications at the Silverlight Developer Center on MSDN. For a current list of supported browsers and operating systems, visit the Silverlight website.

SmoothStreamingMediaElement provided by Smooth Streaming Client SDK (for Silverlight) can support clear and PlayReady-protected Smooth Streaming and MPEG-DASH.

HTML5 Video/Audio with Media Extensions (EME/MSE)

When used with HTML5 Media Source Extensions (MSE) and Encrypted Media Extensions (EME), MPEG-DASH also provides a cross-platform solution for delivering protected content. Unlike Silverlight however, it doesn’t require any specialized plug-ins. PlayReady provides full support for MPEG-DASH, MSE, and EME. To learn more about the standards and PlayReady support for them, see Supported Media Formats.

MPEG-DASH and MSE are currently supported by the latest versions of Internet Explorer, Chrome, Safari, and Firefox. In addition, Windows 8.1 introduced built-in support for EME APIs and maintained native support for PlayReady.

To develop a web application that delivers PlayReady-protected content by using MPEG-DASH, MSE, and EME, you can use your preferred IDE. For detailed information about implementing PlayReady protection for MPEG-DASH content, see DASH Content Protection using Microsoft PlayReady on the PlayReady website.

PlayReady Device Porting Kit

The PlayReady Device Porting Kit provides a primitive set of PlayReady APIs, portable source code, tools, test resources, and documentation to help you port PlayReady technologies to a wide variety of system architectures, operating system environments, and types of devices — game consoles, STBs, Smart TVs, embedded displays, gateway devices, and more. You can use the porting kit, along with your preferred development environment, to implement PlayReady technologies on virtually any platform.

A definitive component of the porting kit is a comprehensive API for calling PlayReady functions from an application. The functions span all PlayReady features and are designed to help you leverage and accommodate differing device and platform capabilities and configurations. Some functions are required because they enable fundamental tasks such as initializing, decrypting, and rendering protected media files. Other functions are optional, depending on the features that you want to implement.

For example, by using the kit you can build a client that performs basic protection functions and also does any or all of the following:

- Uses secure and anti-rollback clocks to enforce time-based policies.

- Acquires licenses directly “over the air” (DLA-OTA) through a wireless network or indirectly via a proxy application.
- Uses a hardware-based cryptographic core that you provision by using the kit.
- Acts as an ND transmitter or receiver that sends or receives and plays streaming content, enabling users to share media content between devices that are connected to the same IP network.
- Supports key rotation, scalable chained licenses, blackout services, and ad insertion for Live TV content.
- Meters content usage.
- Provides backward compatibility with content that is protected by Windows Media DRM (WMDRM) technology.

The porting kit clearly indicates whether a specific function is required or optional. It also explains which functions to use for specific PlayReady features and how to implement them.

Another key component of the porting kit is platform-independent source code for a PlayReady client application. The source code is written to conform to the ANSI C standards and it supports both big- and little-endian formats, which means it’s truly designed to work with multiple compilers and platforms. You can use the source code as a starting point for your own client by modifying it to conform to your target architecture. The kit makes it easy to set system and environment variables such as processor type and target platform, choose compile-time options that optimize memory usage and threading, and set link-time options that specify which features and functionality to enable. From there, you’re ready to start building, testing, and optimizing your client.

Overall, the PlayReady Device Porting Kit offers the following components:

- A complete set of PlayReady APIs that conform to the ANSI C standards and support both big- and little-endian formats.
- Platform-independent source code that you can port to any device or platform and use as a starting or reference point for your own client.
- A hardware abstraction layer that enables critical cryptography tasks to be executed on a secure processor or for processor-intensive tasks to be run on dedicated hardware.
- Code samples that demonstrate how various functions work and can serve as a starting point for your own code. You can customize each sample to target a specific platform.
- A comprehensive programming guide that explains how to implement specific PlayReady features and support various distribution models.

- Detailed reference information for functions, structures, enumeration types, data types, error codes, and messages.
- Technical specifications for PlayReady extensions, protocols, and formats.
- A test kit that contains a collection of scripts that you can compile and run individually or as a batch to test your code. Each test area includes its own executable, script file, and output examples. You can also build custom test areas.
- Several sets of tools and utilities, including tools for testing cryptography code, utilities for inspecting files and stores such as licenses in a license store, and a tool for verifying that media files are transferred correctly to a device.
- Test certificates and keys, and a certificate request kit for generating key pairs and requesting the certificates needed to release a client to a production environment.

As a porting kit user, you also gain access to a PlayReady test server. The test server provides a series of test scenarios and protected media files that you can use to test whether your client can acquire content and process licenses correctly.

There are no specific hardware or system requirements for using the porting kit, although the kit has been tested specifically on computers running current versions of the Windows operating system and Ubuntu-based Linux distributions. In addition, the porting kit works with any compiler that supports ANSI C, such as Microsoft Visual Studio®. To help you get started, the kit includes compiler definitions for Microsoft C and GNU C as well as empty compiler definitions that you can use to customize the kit for a specific compiler.

Tools and Options by Platform

Considering the variety of options, the following table can help you identify the client development options for each major platform. The table is sorted alphabetically by platform.

Platform	Client Development Options
Android	To develop apps for Android version 4.0 (API level 14) and later, a PlayReady Client SDK is available for Android. A variety of SDKs have also been developed by PlayReady licensees that support additional functionalities including support for earlier versions of Android.
iOS	To develop apps for iOS 6.0 and later, a PlayReady Client SDK is available for iOS. A variety of SDKs for iOS have also been developed by PlayReady licensees that support additional functionalities.

Platform	Client Development Options
Linux	To develop applications on a Linux based device platform, use the PlayReady Device Porting Kit. The porting kit has been tested specifically with Ubuntu-based distributions but can be used for any Linux distribution.
Max OS X	Use the Silverlight browser plug-in which provides built-in support for PlayReady technologies. For information about the Silverlight plug-in, visit the Silverlight website.
Windows 7	Use the Silverlight browser plug-in which provides built-in support for PlayReady technologies. For information about the Silverlight plug-in, visit the Silverlight website.
Windows 8	Windows 8 and later natively support PlayReady. Use any of the following: • For apps, use the PlayReady Client SDK for Windows 8.1 Store Apps and the Smooth Streaming Client SDK. • For web- or browser-based applications that support HTML5 media extensions, use EME and MSE for ABR streaming delivery. • For web- or browser-based applications that don't support PlayReady via EME, use the Silverlight browser plug-in. • For desktop applications use Silverlight to develop an Out-of-Browser application. All of those SDKs are available from the Visual Studio Gallery.
Windows Phone	Windows Phone 8 and later natively support PlayReady. To develop apps for Windows Phone, use the Smooth Streaming Client SDK for Windows Phone 8.1 and the PlayReady Client SDK for Windows 8.1 Phone Apps. Both SDKs are available from the Visual Studio Gallery.
Xbox	Xbox 360 and Xbox One natively support PlayReady. To develop an application for either console, use the Xbox SDK (XDK) or Application Development Kit (ADK), depending on the type of application.

Testing Resources

To help you test a PlayReady client, Microsoft provides a comprehensive set of test tools for licensed users of a PlayReady client SDK or the PlayReady Device Porting Kit.

Of the test tools, a key resource is the PlayReady test server. The test server is a fully functioning PlayReady server that helps you test and debug a PlayReady client for a multitude of tasks such as acquiring licenses for different types of media content, enforcing various types of policies, joining and leaving domains, and processing metering data. The server hosts a PlayReady license service, provides a series of test cases, and stores sample media files in various formats and with various policies. To find and learn about each test case and related sample files, you can use the server's web interface and in-depth documentation that's included in the SDKs and porting kit.

In addition to test server access, the PlayReady Device Porting Kit provides a suite of tools and utilities for testing ported code. This includes executable files and source code for learning and testing specific PlayReady functions, tools for testing cryptography code, and utilities for inspecting files and stores. The porting kit also provides a test framework with predefined tests, organized by functional area, and a modular structure and source code that you can use to build and integrate custom tests into the framework.

All of the PlayReady test resources can be used in addition to the tools and test resources provided by other SDKs and the IDE that you use to develop a PlayReady client.

Release Requirements

Before you can release a PlayReady client to a production environment, you need to perform various tasks in addition to typical release tasks for any client. For PlayReady clients, additional pre-release tasks include verifying that the client meets PlayReady compliance and robustness requirements and obtaining the appropriate certificates from Microsoft. Pre-release tasks vary based on several factors, such as which type of client you want to release and how you want to distribute it. The PlayReady client SDKs and PlayReady Device Porting Kit provide instructions for preparing to release a PlayReady client and Microsoft provides additional resources to guide you through the release process.

Compliance and Robustness

PlayReady clients, like all PlayReady technologies and implementations, are governed by a content protection regime. The regime defines and enforces compliance and robustness rules, administers a chain of trust, and establishes remedies for security breaches. A security breach is a circumvention or non-compliant implementation of PlayReady technology that puts content at risk by allowing users to bypass or remove restrictions on content usage.

Compliance rules specify required behaviors of PlayReady implementations and the software that accesses those implementations. They describe how media content may be accessed and shared according to specific rights and restrictions. Robustness rules specify protection requirements for each type of media asset in a PlayReady implementation. They ensure that implementations are designed to protect content in a robust manner.

The PlayReady compliance and robustness rules are specified in PlayReady license agreements and all PlayReady implementations and products must satisfy the requirements defined in those rules. Microsoft defines and enforces the rules, acting as the compliance and robustness administrator for PlayReady technologies. Microsoft also maintains processes and resources for discovering and remedying security breaches, including a dedicated breach response team. To read the compliance and robustness rules, see the Compliance & Robustness Rules area of the PlayReady website or your license agreement

Licensing Options

To develop and distribute a PlayReady client for Microsoft endpoints, such as Windows 8, Windows Phone, Xbox, or Silverlight, you do not need a PlayReady license. You do however need a PlayReady Master Agreement and a PlayReady Service Deployment License to distribute licenses to Microsoft endpoints.

To develop and distribute a PlayReady client for all other platforms, you'll need to sign the PlayReady Master Agreement and purchase the PlayReady license that reflects your business needs, as indicated in the following table. After you obtain a license, you can download the SDK and other packages associated with the license.

License	Scenarios	Includes
Microsoft PlayReady Intermediate Product License	For developing a PlayReady client for iOS or Android devices or developing a client device such as an STB, Smart TV, or media player.	PlayReady Device Porting Kit, PlayReady Client SDKs for iOS and Android, PlayReady Document Pack, PlayReady Windows 8.1 Sample Application with ND, CDMi Example Code for PlayReady, Client SDK SL2000 Test Library, Company Device Test Certificate
Microsoft PlayReady Final Product License	For distributing a PlayReady client or client device to users or using PlayReady clients in a commercial deployment.	PlayReady Certificate Generation Kit, PlayReady Client SDKs for iOS and Android, PlayReady Document Pack, PlayReady Windows 8.1 sample application with ND, client SDK SL2000 library, Company Device Certificate

If you want to try PlayReady before purchasing a license to develop an Android or iOS client, you can request a PlayReady evaluation agreement by contacting wmla@microsoft.com. The evaluation agreement includes the appropriate PlayReady client SDK and a non-production SL150 certificate for testing a client with PlayReady.

Instead of licensing PlayReady technologies directly, you can contract with a Microsoft PlayReady ASP licensee. For more information, see Approved Microsoft PlayReady Licensees.

For additional information about PlayReady licensing, see the Licensing page on the PlayReady website. If you have questions about PlayReady licenses or the licensing process, please contact Microsoft at wmla@microsoft.com.