PlayReady Header Object

Microsoft Corporation

October 2014

Applies to

Microsoft® Silverlight™ Product and any end product created with the PlayReady Server Software Development Kit, PlayReady Porting Kit or PlayReady PC Software Development Kit.

Legal Notice

This document is provided "as-is." The Information contained in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may not remove any notices from this document.

PlayReady Header Objects

The PlayReady header object contains the following fields.

Field name	Field type	Size (bits)	Description
Length	DWORD	32	Holds the size of the PlayReady header object in bytes. The length of a PlayReady header object should not exceed 15 kilobytes (KB).
PlayReady Record Count	WORD	16	Specifies the number of PlayReady records in the PlayReady object.
PlayReady Records	BYTE array	Varies	Contains a variable number of records that contain information related to licenses and license acquisition.

PlayReady Records

The PlayReady header object consists of additional sub-objects called *PlayReady records*. PlayReady records contain the following fields.

Field name	Field type	Size (bits)	Description
Record Type	WORD	16	Specifies the type of data stored in the record value field.
Record Length	WORD	16	Specifies the size in bytes of the record value field.
Record Value	BYTE array	Varies	The content of the object depends on the value of record type.

The **Record Type** field has one of the following values.

Value type	Description
0x0001	Indicates that the record contains a rights management header.
0x0002	Reserved.
0x0003	Indicates an embedded license store.

Rights Management Header

v4.1.0.0

The rights management header is used for a client to locate or acquire a license for the piece of content it is stored in. It is encoded using UTF-16. Content packaged using the PlayReady Server SDK uses this header.

Differences Between Versions

PlayReady 2.0 SDKs and later clients are able to process both the v4.0 and v4.1 WRMHeader versions. Prior PlayReady SDKs return an "unsupported version" error when provided 4.1 headers.

The WRMHeader format v.4.1.0.0 has the following changes compared to v4.0.0.0:

- The **WRMHEADER** element's version attribute is set to the string "4.1.0.0".
- The **DATA** element contains an optional **DECRYPTORSETUP** element.
- The **KID** element is located inside the **PROTECTINFO** element and is optional rather than required.
- The **KID** element contains the attributes ALGID (required), CHECKSUM (optional), and VALUE (required).
- The **KEYLEN** element has been removed. The KEYLEN attribute was previously used to disambiguate cocktail licenses with different length keys. The v4.1 header will break the ability to support anything but 8-byte cocktail keys. If you use cocktail keys that aren't 8-byte, you must use v4.0 headers.
- The **ALGID** and **CHECKSUM** elements have been removed since their data is contained within attributes of the **KID** element.

Format

The v4.1 header has the following syntax.

```
<WRMHEADER version="4.1.0.0"
xmlns="http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader">
  <DATA>
      <PROTECTINFO>
        <KID value=”base64-encoded guid” ALGID=”AESCTR” CHECKSUM=”base64-
encoded value” />
      </PROTECTINFO>
      <LA_URL> URL for license acquisition WS </LA_URL>
      <LUI_URL>
        URL for Non-silent license acquisition web page
      </LUI_URL>
    <DS_ID> base64-encoded guid </DS_ID>
    <CUSTOMATTRIBUTES xmlns="">
      <mm:Publisher xmlns:mm="urn:schema-musicmogul-com"
           <mm:Author>
                Elvis Presley
           </mm:Author>
           <mm:CreationDate>
                2007/08/21:12:00:00
           </mm:CreationDate>
        </mm:Publisher>
    <CUSTOMATTRIBUTES>
    <DECRYPTORSETUP>ONDEMAND</DECRYPTORSETUP>
  </DATA>
</WRMHEADER>
```

The v4.1 tags are described below.

Tag name	Required	Description
WRMHEADER	Yes	Outermost element of the header object. It can contain one **DATA** element and must contain one version attribute. The version for the header is “4.1.0.0”. Every time Microsoft defines new mandatory tags or attributes, a new version number is associated with those tags or attributes. If the version is greater than that for which the client code was written, then the client code must fail, because it implies that the header contains mandatory tags that the client does not understand. If the version is less than or equal to that for which the client code was written, than the client code can safely skip any tags or attributes that it does not understand.
DATA	No	Container element for header data, including third-party tags. Only up to one **DATA** element may be included in the **WRMHEADER** element.
PROTECTINFO	No	Specifies zero or one KID elements that may be used for creating decryptor objects for the associated content. Only up to one **PROTECTINFO** element may be included in the **DATA** element.
KID	No	Contains all key data for a given license. Either one or zero **KID** elements may exist under the PROTECTINFO node. KID supports the following attributes VALUE: Required. Contains a base64-encoded key ID GUID value. Note that this GUID (DWORD, WORD, WORD, 8-BYTE array) value must be little endian byte order. ALGID: Required. Specifies the encryption algorithm. Must be set to either: AESCTR-128, or COCKTAIL CHECKSUM: Optional. Contains a checksum calculated using the KeyId and content key. Refer to the PlayReady AES Key Checksum Algorithm section of this document for details. If this node exists in the WRMHeader XML then its data value must be empty.
LA_URL	No	Contains the URL for the license acquisition Web service. Only absolute URLs are allowed. Only up to

Tag name	Required	Description
		one **LA_URL** element may be included in the **DATA** element. If this node exists in the WRMHeader XML then its data value must not be empty.
LUI_URL	No	Contains the URL for a non-silent license acquisition Web page. Only absolute URLs are allowed. Only up to one **LUI_URL** element may be included in the **DATA** element. If this node exists in the WRMHeader XML then its data value must not be empty.
DS_ID	No	Service ID for the domain service. Only up to one **DS_ID** element may be included in the **DATA** element. If this node exists in the WRMHeader XML then its data value must not be empty.
CUSTOMATTRIBUTES	No	The content author can add arbitrary XML inside this element. Microsoft code does not act on any data contained inside this element. Only up to one **CUSTOMATTRIBUTES** element may be included in the **DATA** element. If this node exists in the WRMHeader XML then its data value must not be empty.
DECRYPTORSETUP	No	This tag may only contain the value of ONDEMAND. When this tag present in the DATA node and its value is set to ONDEMAND then it indicates to an application that it should not expect the full license chain for the content to be available for acquisition, or already present on the client machine, prior to setting up the media graph. If this tag is not set then it indicates that an application can enforce the license to be acquired, or already present on the client machine, prior to setting up the media graph. Only up to one **DECRYPTORSETUP** element may be included in the **DATA** element.

Notes for v4.1:

- All XML tags and attributes in the rights management header are defined by Microsoft. The only exception is the content of the **CUSTOMATTRIBUTES** element. PlayReady PC application developers must not add any custom tags outside of the **CUSTOMATTRIBUTES** element.
- The WRMHeader should abide by the W3C Canonical XML v1.1 specifications (http://www.w3.org/TR/xml-c14n11/).
- The rights management header does not contain a top-level ?XML tag that is required in well-formed XML.

- It is recommended that the size of this field should not exceed 1 KB.

v4.0.0.0

The rights management header is used for a client to locate or acquire a license for the piece of content in which it is stored. It is encoded using UTF-16. Content packaged using the PlayReady Server SDK and encrypted with Advanced Encryption Standard (AES) in counter mode uses this header.

The v4.0 header is stored as a record of type 0x0001 in the PlayReady object and has the following syntax:

```
<WRMHEADER xmlns="http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader"
version="4.0.0.0" >
<DATA>
      <PROTECTINFO>
        <KEYLEN>16</KEYLEN>
        <ALGID>AESCTR</ALGID>
      </PROTECTINFO>
      <LA_URL> URL for license acquisition WS </LA_URL>
      <LUI_URL>
        URL for Non-silent license acquisition web page
      </LUI_URL>
    <DS_ID> base64-encoded guid </DS_ID>
    <KID>  base64-encoded kid  </KID>
    <CUSTOMATTRIBUTES xmlns="">
      <mm:Publisher xmlns:mm="urn:schema-musicmogul-com"
           <mm:Author>
                Elvis Presley
           </mm:Author>
           <mm:CreationDate>
                2007/08/21:12:00:00
           </mm:CreationDate>
        </mm:Publisher>
    <CUSTOMATTRIBUTES>
    <CHECKSUM>
      checksum of the content key for verification
    </CHECKSUM>
  </DATA>
</WRMHEADER>
```

The following table describes the different tags.

Tag name	Required	Description
WRMHEADER	Yes	Outermost element of the header object. It can contain one **DATA** element and one version attribute. The current version for the header is 4.0.0.0. Semantics for packager:

Tag name	Required	Description
		Every time Microsoft defines new mandatory tags or attributes, a new version number is associated with those tags or attributes. The version of the RMHEADER must be set to the highest of the versions of the mandatory tags and attributes present in the header. Semantics for client: If the version is greater than that for which the client code was written, then the client code must fail because it implies that the header contains mandatory tags that the client does not understand. If the version is less than or equal to that for which the client code was written, then the client code can safely skip any tags or attributes it does not understand.
DATA	Yes	Container element for header data, including third-party tags.
PROTECTINFO	Yes	Specifies the type of encryption using the KEYLEN and ALGID child elements.
KEYLEN	Yes	Specifies the size of the content key. Must be set to 16 if ALGID is set to AESCTR and 7 if ALGID is set to COCKTAIL.
ALGID	Yes	Specifies the encryption algorithm. Must be set to the following value: AESCTR: Corresponds to the AES algorithm in counter mode. COCKTAIL: Corresponds to the Cocktail algorithm.
KID	Yes	Contains a base64-encoded key ID GUID value. Note that this GUID (DWORD, WORD, WORD, 8-BYTE array) value must be little endian byte order.
CHECKSUM	No	Contains checksum calculated using KeyId and content key. See Checksum Algorithm section for details. Previous versions of PlayReady treated this field as required, so it should be included in any header that is going to be consumed by previous versions of PlayReady.
LA_URL	No	Contains the URL for the license acquisition Web service. Only absolute URLs are allowed.
LUI_URL	No	Contains the URL for a non-silent license acquisition Web page. Only absolute URLs are allowed.

Tag name	Required	Description
DS_ID	No	Service ID for the domain service.
CUSTOMATTRIBUTES	No	The content author can add arbitrary XML inside this element. Microsoft code does not act on any data contained inside this element.

Notes for v4.0:

- All XML tags and attributes in the rights management header are defined by Microsoft. The only exception is the content of the **CUSTOMATTRIBUTES** element. PlayReady PC application developers must not add any custom tags outside of the **CUSTOMATTRIBUTES** element as doing so may clash with future tags that Microsoft defines.
- The order of child elements within a container element does not matter.
- Note that the rights management header does not contain a top-level `?XML` tag that is required in well-formed XML.
- It is recommended that the size of this field should not exceed 1 KB.
- CHECKSUM is required by PlayReady Server SDK up to version 1.2.
  Since version 1.5, PlayReady Server SDK treats the CHECKSUM as optional.
  PlayReady Porting Kit 1.2 out of the box requires the CHECKSUM.
  PlayReady Porting Kit 2.0 treats the CHECKSUM as optional.

Embedded License Store

It is good practice to add an empty embedded license store to the PlayReady header object under the following conditions:

- The PlayReady Header Object is to be inserted in a content file.
- The content may be used in a context of PlayReady domains with embedded licenses.

This allows a PlayReady client to further embed a domain-bound license in the PlayReady Header Object by simply populating the existing embedded license store, and saves the effort of having to re-header the entire file with a new PlayReady Header Object of a larger size than that of the initial one.

Note It is recommended that you do not include an empty embedded license store in a PlayReady Header Object aimed at being inserted as a base-64 string in a Smooth Streaming Client Manifest.

Note The recommended size is 10KB.

Content Key Algorithm

```
byte[] GeneratePlayReadyContentKey(byte[] keySeed, Guid keyId)
{
    const int DRM_AES_KEYSIZE_128 = 16;
    byte[] contentKey = new byte[DRM_AES_KEYSIZE_128];

    //
    //  Truncate the key seed to 30 bytes, key seed must be at least 30 bytes
long.
    //
    byte[] truncatedKeySeed = new byte[30];
    Array.Copy(keySeed, truncatedKeySeed, truncatedKeySeed.Length);

    //
    //  Get the keyId as a byte array
    //
    byte[] keyIdAsBytes = keyId.ToByteArray();
    //
    //  Create sha_A_Output buffer.  It is the SHA of the truncatedKeySeed
and the keyIdAsBytes
    //
    SHA256Managed sha_A = new SHA256Managed();
    sha_A.TransformBlock(truncatedKeySeed, 0, truncatedKeySeed.Length,
truncatedKeySeed, 0);
    sha_A.TransformFinalBlock(keyIdAsBytes, 0, keyIdAsBytes.Length);
    byte[] sha_A_Output = sha_A.Hash;
    //
    //  Create sha_B_Output buffer.  It is the SHA of the truncatedKeySeed,
the keyIdAsBytes, and
    //  the truncatedKeySeed again.
    //
    SHA256Managed sha_B = new SHA256Managed();
    sha_B.TransformBlock(truncatedKeySeed, 0, truncatedKeySeed.Length,
truncatedKeySeed, 0);
    sha_B.TransformBlock(keyIdAsBytes, 0, keyIdAsBytes.Length, keyIdAsBytes,
0);
    sha_B.TransformFinalBlock(truncatedKeySeed, 0, truncatedKeySeed.Length);
    byte[] sha_B_Output = sha_B.Hash;

    //
    //  Create sha_C_Output buffer.  It is the SHA of the truncatedKeySeed,
the keyIdAsBytes,
    //  the truncatedKeySeed again, and the keyIdAsBytes again.
    //
    SHA256Managed sha_C = new SHA256Managed();
    sha_C.TransformBlock(truncatedKeySeed, 0, truncatedKeySeed.Length,
truncatedKeySeed, 0);
    sha_C.TransformBlock(keyIdAsBytes, 0, keyIdAsBytes.Length, keyIdAsBytes,
0);
    sha_C.TransformBlock(truncatedKeySeed, 0, truncatedKeySeed.Length,
truncatedKeySeed, 0);
    sha_C.TransformFinalBlock(keyIdAsBytes, 0, keyIdAsBytes.Length);
    byte[] sha_C_Output = sha_C.Hash;
    for (int i = 0; i < DRM_AES_KEYSIZE_128; i++)
    {
        contentKey[i] = Convert.ToByte(sha_A_Output[i] ^ sha_A_Output[i + 
DRM_AES_KEYSIZE_128]
                                        ^ sha_B_Output[i] ^ sha_B_Output[i + 
DRM_AES_KEYSIZE_128]
                                        ^ sha_C_Output[i] ^ sha_C_Output[i + 
DRM_AES_KEYSIZE_128]);
    }

    return contentKey;
}
```

PlayReady AES Key Checksum Algorithm

The checksum in the RM headers are intended to protect against mismatched keys. In the early days of DRM, songs were encrypted with incorrectly labeled keys and when the songs were decrypted, white noise was played back and if played loud, destroyed playback equipment. With the checksum, it can be verified that the content key was the key that was used to encrypt the file. The algorithm works as follows:

For **ALGID** value set to AESCTR, 16-byte KeyId is encrypted with 16-byte AES content key using ECB mode. The first 8 bytes of the buffer is extracted and base64 encoded.

For **ALGID** value of COCKTAIL, perform the following steps:

1. A 21-byte buffer is created.
2. The content key is put in the buffer and the rest of the buffer is filled with zeros.
3. For five iterations:
    a. buffer = SHA-1 (buffer).
4. The first 7 bytes of the buffer are extracted and base64 encoded.
5. After these steps are performed, the base64-encoded bytes are used as the checksum.

CUSTOMATTRIBUTES

A service provider can add proprietary XML inside the **CUSTOMATTRIBUTES** element of the RM header. Any tags used inside the **CUSTOMATTRIBUTES** element are guaranteed to not clash with future tags defined by Microsoft.

Microsoft code does not act on any XML inside this element. The service provider's backend or their client side code are the only ones who typically interpret the value of this element. For example, let's say a white label service represents front-end services AAA, BBB, CCC. Such a service can encrypt its content library only once (since that is an expensive operation), but when it serves out content to an end-user, it can set the **CUSTOMATTRIBUTES** to the name of the specific front-end service that the end-user subscribes to. When the end user requests a license for that content, this enables the white label service to determine which front-end service the end-user subscribes to, so that it can issue a different license.

It is recommended that the size of this field should not exceed 1kilobyte (KB).