1. Capitalization

Section 5 reads:

> **5. Technical Specifications**
> In all cases, the UITS payload has two primary components: metadata and the associated signature. **The syntax used for organization is XML, and there is no special requirement that element or attribute names be capitalized.** At a high level, the payload structure is
> <UITS>
> <metadata></metadata>
> <signature></signature>
> </UITS>

This section is revised as follows:

> **5. Technical Specifications**
> In all cases, the UITS payload has two primary components: metadata and the associated signature. **Element names ARE case sensitive.** At a high level, the payload structure is
> <UITS>
> <metadata></metadata>
> <signature></signature>
> </UITS>

To summarize:

- XML element tags (tag names) ARE case sensitive
- XML element contents ARE NOT case sensitive (includes key ID, media hash, and signature)

2. ID3 PRIV frame

The UITS MUST be properly formatted into the PRIV frame. All frames within the ID3 tag SHOULD be properly formatted.

Common Issues

- Actual length of tag and tag length stated in header are not the same
- Extra nulls before owner put owner in position normally occupied by the UITS

3. UITS Element

Section 5.2 specifies:

5.2 UITS
REQUIRED. The UITS element contains a <metadata> element and a <signature> element. The version of the UITS specification is denoted by the namespace. The current version is 1.1. Within the UITS element, namespace notations **MAY** be included.

```
<UITS xmlns:uits="http://www.udirector.net/schemas/2009/uits/1.1">
```

This section is revised as follows:

5.2 UITS
REQUIRED. The UITS element contains a <metadata> element and a <signature> element. The version of the UITS specification is denoted by the namespace. The current version is 1.1. Within the UITS element, namespace notations **MUST** be included. The namespace must be specified, but the prefix does not necessarily need to be "uits".

```
<uits:UITS xmlns:uits="http://www.udirector.net/schemas/2009/uits/1.1">
```

4. Recommended Owner (PRIV frame)

Recommended value is:

```
mailto:uits-info@umusic.com
```

5. Nonce

The nonce does NOT have to be static across multiple downloads of the same track and transaction. All other information related to the track and transaction must be the same as in the original download.

6. User ID (UID) and Transaction ID (TID)

Please note that User ID (UID) ***OR*** Transaction ID (TID) is required. It is not an error to include both, but only one is required.

7. Media Hash

Section 6.3.8 reads:

6.3.8 Media Identifier
REQUIRED. A hash of the media (e.g. the audio, video, etc.) portion of the file MUST be included, such that the UITS payload can be directly tied to the file associated with it. The hash type is passed as an attribute, and the currently valid type is "SHA256". Metadata fields MUST NOT be included in the hash computation, because it must be possible for users to update standard ID3 or similar tags without affecting the media hash. Note that the hash can be pre-computed so that it is not necessary to compute it at the time of sale.

Instructions for MP3s: The hash value for the audio part of the MP3 is calculated using the hash algorithm over all of the audio frames within the file in the order in which they appear in the file. The audio frames all start with a 12-bit syncword, their frame size is calculated in the standard manner from the bitrate, sampling and padding. Frames that are not audio (such as frames containing ID3 tags) are excluded. In addition, one type of audio-related frame must be identified and not included in the cryptographic hash, specifically, older Xing variable byte rate data frames. They are documented at: http://gabriel.mp3-tech.org/mp3infotag.html and http://www.codeproject.com/KB/audio-video/mpegaudioinfo.aspx.

Note that future versions of UITS may support fingerprints, embedded watermarks, or other techniques for identifying audio.

```
<Media algorithm="SHA256">A675BD878C8658C99A…</Media>
```

This section is revised as follows:

6.3.8 Media Identifier
REQUIRED. A hash of the media (e.g. the audio, video, etc.) portion of the file MUST be included, such that the UITS payload can be directly tied to the file associated with it. The hash type is passed as an attribute, and the currently valid type is "SHA256". Metadata fields MUST NOT be included in the hash computation, because it must be possible for users to update standard ID3 or similar tags without affecting the media hash. Note that the hash can be pre-computed so that it is not necessary to compute it at the time of sale.

Instructions for MP3s: *The hash value for the audio part of the MP3 is calculated using the hash algorithm over all of the audio frames within the file in the order in which they appear in the file. Non-audio data such as ID3 tags of any version (ID3.1.x through ID3v2.4.x) are excluded. Beware that ID3V1 tags appear following the end of audio, not the beginning. The audio frames all start with an 11-bit syncword, their frame size is calculated in the standard manner from the bitrate, sample rate, and padding. In addition, variable bit-rate ("VBR") files usually contain a "Xing", "Info", or "VBRI" frame, which must be excluded. Such frames appear as the first audio frame, though they do not produce any sound when decoded. Therefore, when they are present, the hash*

should start at the first byte of the second audio frame. These special VBR frames are documented at: HYPERLINK "http://gabriel.mp3-tech.org/mp3infotag.html" http://gabriel.mp3-tech.org/mp3infotag.html and http://www.codeproject.com/KB/audio-video/mpegaudioinfo.aspx.

Note that future versions of UITS may support fingerprints, embedded watermarks, or other techniques for identifying audio.

```
<Media algorithm="SHA256">A675BD878C8658C99A…</Media>
```

Please note that the media hash should be expressed in hex. The UITS gem outputs the media hash in hex. The XML Schema Definition (XSD) indicates that the media hash can be expressed in base64. Base64 is presently not accepted in UITS Version 1.1 verification. Base64 may be accepted in a later version.

8. URLs

Section 5.3.9 states:

5.3.9 URL
OPTIONAL. The URL field is used for links that are worth transporting in a cryptographically signed manner. The type field specifies the URL type and currently uses a subset of ID3v2.3 tag names. Options for the type field include WCOM, WCOP, WOAF, WOAR, WOAS, WORS, WPAY, and WPUB, which have the meaning described in the ID3 specification. In addition, a legal URL type is KeyURI, which identifies the public key needed to validate the signature (see the Implementation Details section below). More than one URL element MAY be included.

This section is revised as follows:

5.3.9 URL
OPTIONAL. The URL field is used for links that are worth transporting in a cryptographically signed manner. The type field specifies the URL type and currently uses a subset of ID3v2.3 tag names. Options for the type field include WCOM, WCOP, WOAF, WOAR, WOAS, WORS, WPAY, and WPUB, which have the meaning described in the ID3 specification.

In addition, some non-ID3v2.3-derived URL types may be used: A legal URL type is KeyURI, which identifies the public key needed to validate the signature (see the Implementation Details section below).

More than one URL element MAY be included.

9. Key ID

The specification reads:

> keyID – an identifier associated with the public/private key pair used by the distributor or CDN for computing the signature value. The keyID is the SHA1 hash of the public key needed to validate the signature. For RSA, the KeyID is the hash of the DER-encoded RSAPublicKey (as defined in RFC 3447). For DSA, the KeyID is the hash of the DER-encoded DSAPublicKey (as defined in RFC 3279).

The keyID specification is revised as follows:

> keyID – an identifier associated with the public/private key pair used by the distributor or CDN for computing the signature value.
>
> The keyID SHOULD be generated as follows:
>
> The keyID is the SHA1 hash of the PEM-encoded public key file sent to the content owner for validation purposes. The hash MUST be of the entire file needed to validate the signature, inclusive of the standard header and footer.
>
> Example: openssl dgst -sha1 rsa-public-key.pem
>
> The keyID MAY be generated as follows:
>
> For RSA, the KeyID is the hash of the DER-encoded RSAPublicKey (as defined in RFC 3447). For DSA, the KeyID is the hash of the DER-encoded DSAPublicKey (as defined in RFC 3279).
>
> Example: openssl dgst -sha1 rsa-public-key.der

There is an open issue regarding potential platform dependency issues with the preferred method of keyID generation. This will be addressed in a later version of the specification.

10. XML Schema Document

The XML Scheme Document SHOULD be used as a reference for element names and document structure.

The XML Schema Document should NOT be used as a reference for element data types. Invalid values MAY be allowed in the XSD so that they can be caught later in the process, where more informative feedback can be generated.