Eric Fleischman
Anders Klemets
Microsoft Corporation

November 14, 1997
Expires: May 14, 1998

Recording MBone Sessions to ASF Files

Status of This Memo

Abstract

This document specifies two approaches by which multimedia data (e.g., MBone conferences), transmitted using the Real-Time Protocol (RTP), may be recorded to Advanced Streaming Format (ASF) files. The first method requires a minimum amount of buffering at the recording station but results in recordings which identically preserve the received content including out of order packets, network ''jitter'', etc. The second approach requires buffering at the recording station but results in enhanced recordings (i.e., higher percentage of correctly ordered packets, elimination of a percentage of received jitter, potential recovery of a percentage of lost packets). Both approaches record all received RTP content and the relevant subset of RTCP information. This recording occurs transparently to the MBone conference or RTP session, and does not involve any alterations to normal RTP, RTCP, or ASF use.

1. Introduction

The MBone is the part of the Internet that supports IP multicast, and thus permits efficient many-to-many communication. It is used extensively for multimedia conferencing. Such conferences usually have the property that tight coordination of conference membership is not necessary; to receive a conference, a user at an MBone site only has to know the conference's multicast group address and the UDP ports for the conference data streams. The specific MBone conferences addressed by this document are those which use the Real-time Transport Protocol (RTP, see [1]). In addition, the mechanisms described within this document also support unicast RTP uses.

This document describes two methods for recording multimedia data that is transmitted using the Real-Time Transport Protocol (RTP, see [1]) into Advanced Streaming Format (ASF; see [2]) files. The approach is independent of the network protocol used to transmit RTP packets and supports the recording of both unicasted and multicasted sessions. Data thus recorded may subsequently be played back by recreating the original RTP packets and transmitting them using either unicast or multicast techniques. A recording can also be played back locally, using a suitable playback tool. Playback can be controlled using RTSP [4] or other comparable stream control mechanisms.

RTP is a protocol for carrying arbitrary real-time data. Each RTP packet contains a sequence number and timestamp, which can be used by a receiver to detect losses and present the data at the right time. RTP uses a control protocol, RTCP, which can be used to synchronize different real-time streams. For synchronization to be possible, the streams must be transmitted such that each stream has a distinct RTP synchronization source (SSRC) identifier. RTP is most commonly used over UDP. However, it may be used with any transport protocol that detects bit errors, and that conveys the length of an RTP packet. RTP does not specify a mechanism for the reliable transfer of data. The protocol also does not address the encapsulation of specific media types, but instead defers it to various profile specifications.

ASF is an extensible file format for recording optionally synchronized multimedia streams. The format is not tied to any particular media type or compression scheme. Similarly, the file format was designed to be operating system and data communications protocol independent.

2. ASF Overview

The Advanced Streaming Format is defined in [2]. An ASF file consists of three top-level objects: The Header Object, the Data Object and, optionally, the Index Object.

The Header Object provides global information about the file as a whole as well as specific information about the multimedia data stored within the Data Object. This latter content provides the information necessary to correctly interpret each of the media streams found within the Data Object. The Header Object is a container for other objects that provide the following specific functions:

* File Properties Object -- describes the global file attributes.
* Stream Properties Object -- defines a media stream, its characteristics, and the information needed to decode that stream.
* Content Description Object -- contains all bibliographic information, which may be either general for the file as a whole or stream specific.
* Component Download Object -- provides information on playback components.
* Stream Group Object -- logically groups media streams together into specific rendering contexts.
* Scaleable Object -- defines scalability relationships among (scaleable) media streams containing bands.
* Prioritization Object -- defines the relative prioritization between media streams.
* Mutual Exclusion Object -- defines exclusion relationships between media streams (e.g., language selection)
* Inter-Media Dependency Object -- defines dependency relationships among mixed media streams.
* Rating Object -- provides the W3C PICS ([5], [6]) rating of the file.
* Index Parameters Object -- supplies the information necessary to regenerate the index of an ASF file.
* Language List Object -- supplies Language Identifier information that is used by several other ASF objects.

The Data Object contains all the data for each of the recorded media streams. This data is stored in the form of ASF Data Units. In the general case, ASF Data Units are designed to be directly insertable into the payloads of data communications transport protocols in order to be streamed across the network. Each ASF Data Unit is of variable length, and contains data for only one media stream. Data units are sorted within the Data Object based on the time at which they should be delivered (send time). Due to the way Data Units are sorted, consecutive Data Units may contain data from different media streams.

ASF media streams logically (in the general case) consist of sub-elements that are referred to as objects. What an object happens to be in a given media stream is entirely media stream dependent (e.g., it is a specific image within an image media stream, a frame within a (non-scalable) video stream, etc).

The Index Object contains a time-based index into the multimedia data of an ASF file. The time interval that each index entry represents is set at authoring time and stored in the Index Object. Since it is not required to index into every media stream in a file, a list of the media streams that are indexed follows the time interval value. Each index entry consists of one data unit offset per media stream being indexed. This information allows stream-specific index operations to occur.

A minimal ASF implementation consists of a Header Object containing solely a File Properties Object, one Stream Properties object, and one Language List Object as well as a Data Object containing only a single ASF data unit.

3. Recording MBone Sessions

The process of recording MBONE sessions may be viewed as optionally consisting of four steps:

Step 1 -- Create the ASF Header Object, which will provide the context for correctly interpreting the data that may subsequently be recorded.

Step 2 -- Record one or more RTP streams into the ASF Data Object.

Step 3 -- Optionally post-process the ASF Header Object to ensure that it is as complete and as efficiently stored as possible

Step 4 -- Optionally create an ASF Index Object.

3.1. Preparing ASF Header Information

The ASF Header Object contains various other objects that contain information about the media streams in the Data Object. It is often desirable to create an ASF Header Object before the transmission that is to be recorded has begun. This would be appropriate if information is already available that describes the RTP sources that are to be recorded. Such information might be obtained through SDP [7], RTSP [4], or some other non-RTP means. It is also possible to add information to the ASF Header Object as new information is learned during the recording of the RTP traffic.

ASF requires that an instance of the Stream Properties Object (SPO) must be defined to describe each media stream recorded within the Data Object. A media stream generally corresponds to an RTP source in an RTP session. RTP sources, in turn, are identified by the value of the SSRC field in the RTP header. The IP address and port number to which the data is sent identifies RTP sessions. On the MBone, most applications send audio and video on separate RTP sessions, and thus audio and video would be recorded as two separate media streams. However, all RTP packets that belong to a media stream are expected to have identical RTP Payload Type fields. If an RTP source changes the value it is using for the RTP Payload Type field :mid-session", then RTP packets with the new (i.e., different) Payload Type fields should be stored as a different media stream within ASF with its own unique SPO. It is recommended that the relationship between streams that compose the traffic from a single RTP source be associated by grouping them via the ASF Header Object's Stream Group Object.

While the session announcement will generally provide enough information to construct an initial File Properties Object (FPO) and some of the necessary SPOs before the session begins, loosely controlled (MBone) conferences can permit additional participants to join the conference. Therefore, provision should be made to anticipate the possibility of additional speakers joining the session. A recommended way to satisfy this provision is to reserve space within the ASF Header Object via the ASF Placeholder Object (See Appendix A) where additional ASF objects may be written (e.g., additional SPOs) as the MBone session dynamically progresses.

Static RTP Payload Types may be handled in one of two ways:

1. **Static RTP Payload Types should be translated into the equivalent ASF standard media type (see Section 8 of [2]) using the equivalent** ASF codec (e.g., see Reference [10]), if known.
2. **Alternatively, they can be recorded as RTP Media Types as defined in Appendix B.**

Dynamic RTP Payload Types may be handled in one of three ways:

1. **The dynamic RTP payload type should be translated into the equivalent ASF standard media type (see Section 8 of [2]) using the equivalent** ASF codec, if known. This means that the recorder will need to identify the actual codec used by that dynamic RTP Payload Type instance based upon the available information. The identity of this

codec will then need to be expressed as a specific ASF UUID identifier (e.g., see Reference [10]) within the SPO's Codec ID field.

2. **Alternatively, the recorder can translate the dynamic RTP payload type to the appropriate static RTP payload type, if any and record it** as an RTP Media Type as defined in Appendix B.
3. **Alternatively, the recorder can record it as a dynamic RTP payload type as defined in Appendix B.**

RTP payload types, which can not be deciphered by any of the above approaches, should be ignored (i.e., that media stream can not be recorded).

Note that if the RTP payload is translated into the equivalent ASF standard media type, an inverse transformation will need to be applied by a playback device, if the recording is retransmitted as RTP packets.

3.2. Two Recording Approaches

The capabilities of local systems vary. For this reason, the document suggests that limited capability systems seek to record data via the Packet Capture Mode, which is described in section 3.2.1. More capable systems are recommended to use the Record Structure Mode, described in section 3.2.2.

3.2.1. Packet Capture Mode (Limited Buffering)

The Packet Capture Mode recording alternative seeks to write RTP data as it is received to the ASF Data Object on the disk. The clock of the recording computer is used to determine the ASF Data Unit's Send Time value. The Send Time value is calculated by subtracting the multimedia session's start time (as recorded by the recording computer) from the recording computer's current time and converting the result into millisecond units.

The RTP timestamp is directly written as the ASF Data Unit's Presentation Time value, again making the necessary conversions to account for the fact that the initial RTP timestamp value is random while the initial ASF Send Time and Presentation Time values are zero. The granularity of the Presentation Time units (i.e., the Presentation Time Numerator and Presentation Time Denominator fields within the SPO) should be set to the clock granularity for that RTP source. ASF's default presentation time granularity (i.e., a millisecond) should initially be used for those cases in which the actual clock granularity is not known.

The value of the Presentation Time Flags within the SPO for this media stream shall thus be configured to be "11" (i.e., Full Data Unit Presentation Time).

RTCP Sender Reports (SR) for the RTP source being recorded can be used to calculate the clock granularity of the source. This is useful if the clock granularity is otherwise unknown. It is also possible to use Sender Reports to detect skews between the clock granularity used by the source, and the granularity that is given by the RTP Payload Type specification or profile. If such a skew is detected, the Rational Time Values (i.e., Presentation Time Numerator and Presentation Time Denominator fields) of the SPO should be altered accordingly.

This approach has the advantage of being simple and direct to implement. It has the following disadvantages:

* Jitter is preserved - and repeated re-recordings of the same contents by this manner may exacerbate the jitter on each subsequent recording.
* Out-of-order packets remain out of order.

3.2.2. Record Structure Mode (Buffering)

The Record Structure Mode requires that packets be buffered a finite amount of time (e.g., 5 seconds) before being written to disk. Packets within the buffer should be correctly ordered. Packet holes occurring within the buffer interval should be filled by retransmitted packets (if any).

Within this approach, the value of the RTP Timestamp field is used to compute the send time. Since the RTP timestamp starts at a random value, while the ASF Send Time and Presentation Time start at zero, a conversion into appropriate ASF Send Time values must be made. The send time is stored with a 1-millisecond granularity. The appropriate RTP Payload Type specification or profile gives the granularity of the RTP Timestamp. RTCP Sender Reports (SR) may be used calculate the granularity of the RTP Timestamp if it is otherwise unknown. Sender Reports can also be used to detect skews between the RTP Timestamp granularity and the granularity specified in the RTP Payload Type specification or profile. If such a skew is detected, the send time values for currently buffered packets of that media type have to be altered (retaining their millisecond granularities) to correctly reflect the skew.

The following values should be recorded within the Stream Properties Object for the media streams recorded by this approach: The clock frequency of the RTP payload type should be appropriately recorded into the Presentation Time Numerator and Presentation Time Denominator fields. The Presentation Time Flag value should have the value of "01" and the Presentation Time Delta field should have a value of zero. This means that both the ASF send time and presentation time have the same value and that subsequent RTP retransmissions of this data will contain only one timestamp (i.e., RTP's timestamp).

This approach has the advantage of correcting some of the received jitter, correctly sorting some of the out-of-order packets, and potentially filling in some lost packets (assuming a retransmission scheme is used). The disadvantage of this approach is that it is more complex to implement. This is particularly the case if the RTP payload type's clock frequency is not known ahead of time and has to be subsequently learned via RTCP transmissions. In addition, it requires additional buffering on the recording computer.

3.3. Recording MBONE Sessions

The following translations from RTP packet fields to ASF data fields are identical for both recording approaches.

3.3.1. RTP Mixers and Translators

The combined streams resulting from Mixers and Translators need to be demultiplexed back into their original component streams when being recorded into ASF, if possible. If this is not possible, then copies of the RTP packet containing data that is attributed to multiple sources need to be stored into each of these sources' media streams (i.e., ASF Data Units). In either case, these streams may be optionally re-mixed when they are subsequently replayed from the ASF files depending upon local implementation considerations.

3.3.2. RTP Packet Information

The RTP Header's Payload Type field combined with the SSRC is used to determine the ASF Stream Number value for that media stream. This Stream Number value identifies which SPO instance should be used to define this media stream. This value is recorded into the Stream Number field of the ASF Data Unit.

The Version field in the RTP header is not recorded into the ASF file unless it is a version other than 2. If the Version field in the RTP header is other than 2, the RTP version number should be recorded into the ASF Header Object's Content Description Object (CDO; see Section 5.4 of [2]) using a value of 73 for the Field Type field.

The Padding bit, and the Padding field that is present if the bit is set, is not recorded. If an RTP packet where the Padding bit was set is received, the padding field should be removed from the RTP payload. Padding may be regenerated when retransmitting the recording, if necessary.

SSRC information should be written into the CDO as an aid for remembering the association between an SSRC and a Media Stream. This will also permit the original sequence number to be optionally recreated once the recorded data is retransmitted. The 32-bit SSRC value will need to be converted into a string when it is stored into the Value field of the CDO. When storing the SSRC as a Unicode string, the SSRC is treated as an unsigned 32-bit integer, and it must be converted to the local byte order (i.e., host byte order). The value of the Field Type field is 70.

Because the initial RTP timestamp value is a random value, the initial RTP timestamp value should also be recorded into the CDO. This will permit the original timestamp sequence to be optionally recreated once the recorded data is retransmitted. The 32-bit timestamp value will need to be converted into a Unicode string when it is recorded into the Value field of the CDO. The value of the CDO's Field Type field is 71.

The initial RTP Sequence Number value should be recorded into the CDO. This will permit the original number to be optionally recreated once the recorded data is retransmitted. The 16-bit Sequence Number value will need to be converted into a Unicode string when it is stored within the Value field of the CDO. When storing the Sequence Number as a string, the Sequence Number is treated as an unsigned 16-bit integer, and it must be converted to the local byte order (i.e., host byte order). The value of the Field Type field is 72.

It should be noted that ASF's concept of Object Number differs from RTP's concept of Sequence Number although they are both used to identify out-of-order and missing information. [Note: earlier versions of the ASF spec used the term "ObjectID" instead of "Object Number".] The former identifies specific media stream "objects" as a part of a fragmentation and grouping schema. What an object happens to be in a given media stream is entirely media stream dependent (e.g., it is a specific image within an image media stream, a frame within a (non-scalable) video stream, etc). Since object fragmentation occurs within a specific RTP Payload Type instance and RTP headers do not indicate this type of information, an identical translation of the original Object Number semantics would require a decoding of the media stream. The value of pursuing this type of overhead is highly questionable, especially when the ultimate goal of identifying missing or out-of-order information is common between the two approaches. Therefore, the RTP sequence number should be directly mapped into the ASF's Object Number field of the ASF Data Unit. Since the 16-bit Sequence Number starts at a random interval while the 8-bit Object Number starts at zero, the mapping between the Sequence Number and Object ID needs to reflect this difference (e.g., Current-Sequence-Number value minus Original Sequence-Number value = Object Number) and account for the fact that Object Numbers "wrap around" to zero every 2^8th packet and Sequence Numbers "wrap around" when their value hits 2^16.

If the CSRC fields within the RTP header are demultiplexed into their original component streams when being recorded, then the CSRC fields are not recorded. If, however, this is not possible, then the CSRC information should be written into the ASF Data Unit's extension field as described below.

If the RTP payload has been converted into an "equivalent ASF standard media type" (see Section 3.1), then the RTP Extension Object described by the next paragraph is optional. However, if the RTP Media Type described in Appendix B has been used to record the data, then the RTP Extension Object is required to be used if either the RTP Header's M-bit or the RTP Header's eXtension (X) bit are ever set within that stream, or if CSRC information is ever needed to be recorded within that media stream. The RTP Extension Object permits exact copies of the original RTP packets to be regenerated, if desired.

The RTP Extension Object is an instance of the Extension Object that is described within Section 5.3.1 of [2]. Extension Objects are associated with a specific media stream's SPO and indicate the semantics and format of specific data (i.e., in this case RTP Packet Header data) that is stored on a per packet basis within the Extension Data field of the ASF Data Unit (see Section 6.1 of [2]). The RTP Extension Object is defined as follows:

* The value of the Extension Data Size field is 0xFFFF
* The UUID value of the Extension System field is {96800c63-4c94-11d1-837b-0080c7a37f95}.

These definitions indicate that this recording shall follow the "variable length" extension data encoding format (i.e., one bit length field followed by the extension data) within the Extension Data field of the ASF Data Unit.

In the case of the RTP Extension Object, the Extension Data field of the ASF Data Unit has the following syntax:

Field Name:	Size:
Extension Length	8 bits - Size in bytes of the Extension Data and Flag fields (i.e. sizeof (Extension Data) + 1)
Flag	8 bits
X-bit	1 bit (LSB) -- contains the RTP Header's X-bit value
CSRC Count	4 bits -- contains the RTP Header's CC value
M-bit	1 bit -- contains the RTP Header's M-bit value
Reserved	2 bits (MSB)
Extension Data	RTP Header CSRC list, if any, followed by Extension Data, if any

The "variable length" encoding means that if either the X bit is set or the CSRC Count has a non-zero value, then the Extension length, flag, and RTP header extension data are written into the Extension Data field of the ASF Data Unit. If both the X bit is cleared and the CSRC Count has a zero value, then only the extension length and flag fields are written to the Extension Data field of the ASF Data Unit. If both the X-bit is set and the CSRC Count field has a non-zero value, then the CSRC list of the RTP Header appears first immediately followed by the RTP Header Extension data within the Extension Data field. These fields are arranged in big-endian order (also known as network byte order).

3.3.3. RTCP Packet Information

RR and BYE packets are not recorded into ASF files. Clock skew information obtained from SR packets is used for the timestamp calculations described in Sections 3.3.1 and 3.3.2. Other information contained in SR packets, except for APP and SDES information, is not recorded.

SDES information is stored in the ASF Header Object's Content Description Object (CDO). Appropriate SDES items (i.e., "CNAME", "NAME", "EMAIL", "PHONE", "LOC", "TOOL", "NOTE", and "PRIV") shall be written into the CDO as described by Appendix C. Synchronization relationships between media streams containing the same CNAME value should be retained via associating them by ASF's Inter-Media Dependency Object (Section

5.12 of [2]).

APP information should be handled in one of two ways.

1. **If the recorder understands (through out-of-band mechanisms outside of the scope of both ASF and RTP) that the APP information contains** script commands or invocations, which correspond to either the ASF Header Object's Script Command Object (see section 5.5 of [2]) or to a Command Media stream type (see section 8.7 of [2]), then the recorder can convert the APP information into the appropriate ASF constructs.
2. **If the recorder does not understand the APP information then that information should be appropriately recorded "as is" into the ASF** Header Object's Script Command Object.

If the values of the SDES fields from a particular RTP source change during the recording, it is recommended that the CDO contain the initial value for the SDES field. Subsequent values of the SDES fields should then be recorded as a separate media stream, via the mechanisms described in Appendix D.

3.4. Optional Post-Processing of the ASF Header

Whenever live recordings are made, the Live Bit must be set in ASF's File Properties Object. This signifies that certain fields in the ASF File Properties Object and the Stream Properties Object(s) are invalid and should be ignored. In addition, these same files are likely to also contain the ASF Placeholder Object (see Appendix A). It is highly recommended, but not required, that post-processing be done to ASF files to clear the Live Bit, remove the ASF Placeholder Object, and to write valid data into the fields which are invalid when the Live Bit is set.

3.5. Optional Creation of the ASF Index Object

ASF uses the Index Parameters Object in the ASF Header to identify the parameters and media streams whose data will be indexed. This object is described in Section 5.14 of [2]. If the Index Parameters Object does not yet exist for this file, then it needs to be constructed before the Index Object is built. Using the information contained within the Index Parameters Object, the Index Object is constructed as defined in Section **7 of [2].**

3.6. Playback of the Recorded RTP Data

Recorded media streams are stored into the ASF Data Object as ASF Data Units (see Section 6.1 of [2]). Each ASF Data Unit contains a "header field" together with the media data which is being stored. The payload of each RTP packet comprises the media data stored within the ASF Data Unit. The RTP header itself is not stored but its content is mapped into the SPO, CDO, and the header field of the ASF Data Unit.

The ASF file contains sufficient information to play back the recorded data, either locally or via a remote playback device. When RTP packets are recorded into the ASF file using the RTP Media Type (see Appendix B), sufficient information exists to regenerate RTP packets with the same SSRC and sequence numbers as the original packets, if desired. Additionally, it is possible to regenerate RTCP SDES and APP packets with the same content as those sent by the original RTP source. This permits recorded data to be retransmitted into an existing MBone conference, for example, in such a manner that it may appear that the data originates from the original RTP source.

This specification does not define a required feature set for playback devices. For example, even though it is possible to retransmit the recorded data using RTP, playback devices are not required to do so.

Appendix A. ASF Placeholder Object Definition

"Loosely controlled" sessions permit participants to enter and leave without membership control or parameter negotiation. Since one can not always predict how many participants will speak, nor what media types they will use, a mechanism is needed to reserve space within the Header Object so that new Header Objects (e.g., Stream Properties Objects) may be readily added to the header when needed without requiring the header to be re-written.

The purpose of the ASF Placeholder Object is to fulfill this "place holder" function. New header objects are added into the space reserved by the ASF Placeholder Object. The ASF Placeholder Object will then reduce the amount of space it is reserving by the amount taken by the new object(s).

ASF Placeholder Objects are ignored (skipped over) when ASF Header Information is conveyed to remote nodes. Even so, it is recommended that they be removed by post processing (see section 3.4) to make more compact files.

The ASF Placeholder Object is defined as follows:

Field Name:	Size:	Value:
Object ID	128 bits	This field contains the following UUID value: {D6E22A0F-35DA-11d1-9034-00A0C90349BE}
Object Size	64 bits	The size of this object in bytes (i.e., Reserved field value + 24)
Reserved	(Object Size - 24) * 8 bits	Reserved space

Appendix B. RTP Media Type

ASF has defined standard media types for Audio, Video, Image, Timecode, Text, MIDI, Command, and Media-Objects (Hotspots) in Section 8 of [2]. Implementations, which support these types of media streams, are expected to implement them in the manner defined within the ASF standard. MBone content, which is stored within ASF, is therefore expected to be mapped into the standard ASF media streams format whenever possible.

However, occasions will exist when it will not be possible to conform to this requirement. Possible reasons include the following:

* The recorder may not be aware of which media type is associated with an RTP Payload Type (i.e., whether the RTP Payload Type is referring to Audio, Video, or some other media type).
* The recorder may not know which ASF-defined codec corresponds to the codec assumed by the RTP Payload Type and therefore it would be unable to complete the mapping into a standard ASF media type.
* The RTP Payload Type may indicate an interleaved data stream (e.g., video and audio combined into a single stream). No standard ASF media type has yet been defined for such interleaved data.
* The RTP Payload Type may indicate a media type which is not among the standard ASF Media Types.

For these reasons and others, a provision must exist to record MBone data as a distinct RTP Media Type. This appendix defines the format of RTP Media Type.

The RTP Media Type is defined within the Stream Properties Object (SPO) by placing the UUID value {96800c65-4c94-11d1-837b-0080c7a37f95} into the Stream Type field. The following information is then stored as Type-

Specific Data field within the SPO:

Field Name:	Field Type:	Size (bits):	Description:
Payload Type	UINT	8	The Payload Type value indicated by the RTP header.
Profile Size	UINT	16	Size in bytes of the Profile field.
Profile	UINT8	?	ASCII string identifying the Profile which has defined the Payload Type. (E.g., "AVP" for the profile defined by [3] and [9].) An empty string is used if the profile is not known.
Announcement ID Size	UINT	16	Size in bytes of the Announcement ID field.
Announcement ID	UINT8	?	MIME Type of the session announcement mechanism used. (E.g., "application/x-sdp" for SDP [7] announcements.)
Announcement Size	UINT	16	Size in bytes of the Announcement field.
Announcement	UINT8	?	ASCII string containing the definition for this media stream. (E.g., for SDP [7] announcements, this would contain the entire rtpmap entry for this media stream.)

All ASCII strings in the RTP Media Type are terminated by a NULL character. These fields should be stored in little-endian byte order (i.e., the orientation used in the ASF Header Object).

The final four fields (i.e., Announcement ID Size, Announcement ID, Announcement Size, and Announcement) are used to convey information about the dynamic RTP payload type. This information might have been available to the recording device through non-RTP means. Examples of possible sources of such information include session descriptions, such as SDP [7], and presentation descriptions [4]. However, if a static RTP Payload Type is being specified, both the Announcement ID Size and the Announcement Size fields may have a value of zero indicating that the Announcement ID and Announcement fields have not been specified.

The rest of the SPO should be specified as indicated in Section 3.2

above.

The received RTP data of this media stream is stored into the ASF Data Object as described in Section 3.2 and Section 3.3 above.

Appendix C. Recording SDES Information

Section 5.4 of [2] describes the syntax and semantics of the Content Description Object (CDO) within the ASF Header Object. This object consists of an array of Description Records containing four logical entries:

1. **A Field Type value which identifies the semantics of the entry. Each SDES packet may be recorded to the CDO using the following pre-**defined Field Type (unsigned integer) values:

SDES entry:	Field Type Value:
CNAME	61
NAME	62
EMAIL	63
PHONE	64
LOC	65
TOOL	66
NOTE	67
PRIV	68

2. **Stream Number to identify to which media stream this CDO entry refers.**
3. **Name - Name of the entry. This field is redundant to the Field Type value and therefore the field is frequently not used. However,** applications may optionally use this field for language "localization" reasons (e.g., to translate the entry into a specific target language).
4. **Value - the information conveyed by the specific SDES message (e.g., User and domain name in a CNAME packet).**

Appendix D. SDES Media Streams

Section 3.3.3 stated that the first instance of a specific SDES RTCP instance (i.e., a specific SDES item associated with a specific RTP source identifier; e.g., a CNAME value for a specific SSRC) should be recorded into the Content Description Object (CDO). The Stream Number field within the CDO should refer to the media stream associated with the RTP source identifier (i.e., SSRC/CSRC field of section 6.4 of [1]) of that SDES packet chunk. The CDO has provisions for storing only one SDES type instance (e.g., only one instance of a CNAME) for any given media stream. Therefore, subsequent instances of the same SDES type for that media stream will need to be recorded as a distinct "media stream" if that information is to be preserved. This appendix defines how to create such an SDES media stream.

An SDES media stream consists of SDES information written into the ASF Data Object via the mechanisms described in section 3.2. Each SDES media stream records SDES information from only one RTP source identifier. A Stream Properties Object (SPO) is constructed for each SDES media stream. That SDES media stream should also be associated with (i.e., synchronized with) the media stream containing the RTP data of that same RTP source identifier via the ASF Header Object's Inter-Media Dependency Object.

The SPO for a SDES media stream should be constructed as follows:

* The UUID of the SDES Media Stream is {96800c62-4c94-11d1-837b-0080c7a37f95}. This value should be written into the Stream Type field of ASF's Stream Properties Object (SPO) to identify SDES Media Streams.
* The value of the Type-Specific Data Length field within the SPO is zero (i.e., no Type-Specific Data).

The format of an SDES Media Stream consists of one or more instances (per ASF Data Unit) of the following structure:

Field Name:	Field Type:	Size (bits):	Description:
Type Array Size	UINT	16	Size in bytes of the Type Array
Value Array Size	UINT	16	Size in bytes of the Value Array
Type Array	UINT8	?	UTF-2 string [8] identifying the specific SDES type instance (e.g., "CNAME")
Value Array	UINT8	?	UTF-2 string [8] containing the SDES value (e.g., "user and domain name" for a CNAME)

Authors Address
Eric Fleischman
E-mail: ericfl@microsoft.com
and
Anders Klemets
E-mail: anderskl@microsoft.com
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052-8300
USA

References:

1 H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson., "RTP : A Transport Protocol for Real-Time Applications", IETF RFC 1889, January 1996.
2 Microsoft Corporation, "Advanced Streaming Format (ASF) Specification", http://www.microsoft.com/asf/specs.htm, September 1997.
3 H. Schulzrinne, "RTP Profile for Audio and Video Conference with Minimal Control", IETF RFC 1890, January 1996.
4 H. Schulzrinne, A. Rao, and R. Lanphier "Real Time Streaming Protocol (RTSP)", work in progress.
5 J. Miller, P. Resnick, and D. Singer, "Rating Services and Rating Systems (and Their Machine Readable Descriptions)," World Wide Web Consortium http://www.w3.org/PICS/services.html, May 5 1996.
6 T. Krauskopf, J. Miller, P. Resnick, and G. W. Treese, "Label Syntax and Communication Protocols," World Wide Web Consortium http://www.w3.org/PICS/labels.html, May 5 1996.
7 M. Handley, V. Jacobson, "SDP: Session Description Protocol", work in progress.
8 International Standards Organization, "ISO/IEC DIS 10646-1:1993 information technology - universal multiple-octet coded character set (UCS) - part I: Architecture and basic multilingual plane," 1993.
9 "RTP Payload types (PT) for standard audio and video encodings", ftp://ftp.isi.edu/in-notes/iana/assignments/rtp-av-payload-types
10 "ASF Codec GUIDs", http://www.microsoft.com/asf/guids.htm